Received: 27 March 2020 | Revised: 31 October 2020 | Accepted: 30 November 2020

DOI: 10.1002/nme.6608

RESEARCH ARTICLE

WILEY

Tuned hybrid nonuniform subdivision surfaces with optimal convergence rates

Xiaodong Wei[1] | **Xin Li**[2] | **Yongjie J. Zhang**[3] | **Thomas J. R. Hughes**[4]

[1]Institute of Mathematics, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

[2]School of Mathematical Sciences, University of Science and Technology of China, Hefei, China

[3]Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA

[4]Oden Institute, The University of Texas at Austin, Austin, Texas, USA

Correspondence
Xin Li, School of Mathematical Sciences, University of Science and Technology of China, Hefei, China.
Email: lixustc@ustc.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 61872328; NSF grant CBET, Grant/Award Number: 1804929; Office of Naval Research Global, Grant/Award Number: N00014-17-1-2119,N00014-13-1-0500

Abstract

This article presents an enhanced version of our previous work, hybrid nonuniform subdivision (HNUS) surfaces, to achieve optimal convergence rates in isogeometric analysis (IGA). We introduce a parameter λ ($\frac{1}{4} < \lambda < 1$) to control the rate of shrinkage of irregular regions, so the method is called tuned hybrid nonuniform subdivision (tHNUS). Thus, HUNS is a special case of tHNUS when $\lambda = \frac{1}{2}$. While introducing λ in hybrid subdivision significantly complicates the theoretical proof of G^1 continuity around extraordinary vertices, reducing λ can recover optimal convergence rates when tHNUS functions are used as a basis in IGA. From the geometric point of view, tHNUS retains comparable shape quality as HNUS under nonuniform parameterization. Its basis functions are refinable and the geometric mapping stays invariant during refinement. Moreover, we prove that a tHNUS surface is globally G^1-continuous. From the analysis point of view, tHNUS basis functions form a nonnegative partition of unity, are globally linearly independent, and their spline spaces are nested. In the end, we numerically demonstrate that tHNUS basis functions can achieve optimal convergence rates for the Poisson's problem with nonuniform parameterization around extraordinary vertices.

KEYWORDS

extraordinary vertex, isogeometric analysis, nonuniform subdivision, optimal convergence rates

1 | INTRODUCTION

Isogeometric analysis (IGA) has emerged as a powerful technology to unify geometric modeling and numerical simulation,[1,2] which employs the basis functions used in computer-aided design (CAD) also for simulations. IGA has grown into a large family of numerical methods incorporating various spline techniques, such as nonuniform rational B-splines (NURBS),[1] hierarchical B-splines,[3] T-splines,[4-11] polynomial splines over T-meshes,[12] locally refinable B-splines,[13] and subdivision methods.[14-20]

The study of extraordinary vertices has been one of the most active research directions in IGA because they are inevitable in complex watertight geometric representations. An extraordinary vertex in a quadrilateral mesh is an interior vertex shared by other than four faces. Along this direction, simultaneously fulfilling the requirements from both design and analysis is a significant challenge. Numerous methods have been developed over the past few years, but among them, only a few constructions can achieve optimal convergence rates in IGA, such as geometrically smooth multipatch

construction,[21,22] degenerated Bézier construction,[23-25] manifold-based construction,[26] and blended C^0 construction for unstructured hexahedral meshes.[27] A common simplification in all these constructions is to adopt uniform parameterization around extraordinary vertices, that is, the surrounding knot intervals are assumed to be the same. While the support of nonuniform parameterization is a necessary step forward to be compatible with the current industry standard in CAD, that is, NURBS, the related study on the above-mentioned constructions has not been reported in the literature.

On the other hand, subdivision methods, as a generalization of splines, provide a flexible means to deal with extraordinary vertices, where an infinite series of spline patches are smoothly joined around extraordinary vertices. The combination of flexibility and global smoothness makes them not only the standard in the computer animation industry but also a promising candidate for IGA. Indeed, some of the subdivision methods have been studied in the context of IGA, such as the use of Loop subdivision in thin-shell analysis[15] and the development of Catmull–Clark solids.[16] However, several challenging problems need to be carefully investigated before we can fully leverage the power of subdivision methods, such as developing efficient quadrature rules to integrate infinite piecewise polynomials around extraordinary vertices,[28,29] supporting nonuniform parameterizations to be compatible with NURBS,[30-34] and recovering optimal convergence rates.[20] This article intends to address both nonuniform parameterization and the optimal convergence behavior at the same time.

To achieve optimal convergence, the present work is motivated by the idea of *tuned* Catmull–Clark subdivision.[20] Tuned subdivision is a well-studied subject aiming to optimized subdivision stencils (i.e., coefficients in the subdivision matrix) to improve certain properties of a subdivision scheme, for example, to minimize curvature variations to achieve a better surface fairness.[35-37] Recently, it has been explored in the context of IGA to improve accuracy[38] as well as convergence.[20] In particular, the tuned Catmull–Clark subdivision is the first work in IGA that is able to use a subdivision scheme to achieve optimal convergence rates (in the L^2-norm error by solving the Poisson's equation). However, the optimization framework proposed there only works for uniform parameterization and cannot be extended to nonuniform subdivision schemes because the subdivision stencils in a uniform subdivision scheme like Catmull–Clark only depend on the valence of a given extraordinary vertex, and the optimization can be focused on a finite number of stencils of interest. Thus, optimization only needs to be done once and the optimized stencils can be stored for future use. On the other hand, the subdivision stencils in a nonuniform subdivision depend on not only the valence of an extraordinary vertex, but also the surrounding knot intervals, leading to infinite possible cases of stencils. Therefore, it is not feasible to apply optimization to nonuniform subdivision because otherwise it would be very time-consuming and also problem specific.

To deal with nonuniform parameterization, we follow our preceding work on hybrid nonuniform subdivision (HNUS),[14] which generalizes bicubic NURBS to arbitrary topology with proved G^1 continuity. HNUS features high quality in geometric modeling under nonuniform parameterization. When applied to IGA, HNUS basis functions lead to improved yet suboptimal convergence rates compared with Catmull–Clark subdivision.

In this work, we introduce a tuning parameter $\lambda \in \left(\frac{1}{4}, 1\right)$ in HNUS to control the shrinkage rate in irregular regions such that we can recover optimal convergence under nonuniform parameterization. The enhanced version of HNUS is therefore called *tuned hybrid nonuniform subdivision* (tHNUS). In fact, the parameter λ is the subdominant eigenvalue (the second and third eigenvalues which are equal) of the tHNUS subdivision matrix, that plays a crucial role in surface continuity[39] as well as the convergence performance.[20] Note that tHNUS coincides with the original HNUS when $\lambda = \frac{1}{2}$. From the geometric point of view, tHNUS retains comparable shape quality as HNUS. Its basis functions are refinable and the geometric mapping stays invariant during refinement. Moreover, we prove that the tHNUS surface is globally G^1-continuous. From the analysis point of view, tHNUS basis functions form a nonnegative partition of unity, are globally linearly independent, and their spline spaces are nested. Moreover, we numerically demonstrate that tHNUS can achieve optimal convergence rates in the Poisson's problem by reducing λ, regardless of whether parameterization around extraordinary vertices is uniform or not. As an interesting side product, we also show that simply applying the standard Gauss quadrature rule to every element (close to or far away from extraordinary vertices) in tHNUS does not influence simulation accuracy or convergence.

The reminder of the article is organized as follows. Section 2 presents the subdivision rules of tHNUS. The proof of G^1 continuity for tHNUS surfaces is given in Section 3. The tHNUS basis functions their properties are discussed in Section 4. In Section 5, we present numerical tests of both geometric modeling and IGA. Section 6 concludes the article and discusses the future work.

2 | THNUS SURFACES

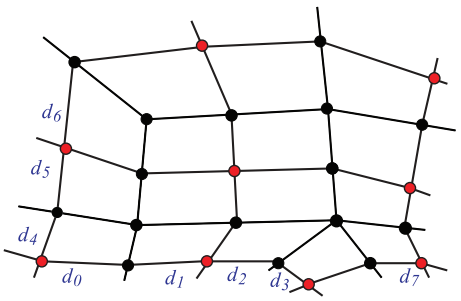
Our discussion assumes that the input control mesh is a regular manifold mesh where all the faces are quadrilaterals (or quads). If initially a mesh has nonquad faces, we apply a single NURSS (Nonuniform Recursive Subdivision Surface) refinement[30,34] to obtain an all-quad mesh. A nonnegative scalar, called the *knot interval*, is assigned to each edge of the control mesh. We further assume that in each face, the knot intervals on the opposite edges coincide. A nonuniform parameterization is obtained by assigning different knot intervals to edges as long as the assumption of knot intervals holds.

The tHNUS consists of two steps of rules: *the topological step* to manipulate mesh connectivity, and *the geometric step* to update the coordinates of involved control points. Each step of rules can be further divided into that for the first level and those for the subsequent levels.

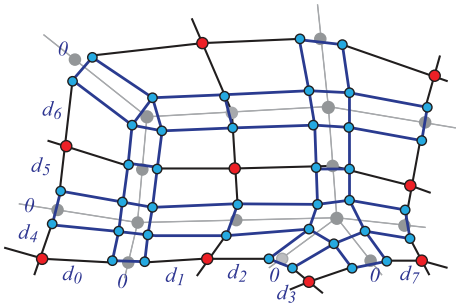
2.1 | Topological step

As shown in Figure 1, we start with the topological step. The rule corresponding to the first level converts the input quad mesh (Figure 1(A)) to its hybrid counterpart (Figure 1(B)). Each extraordinary vertex is replaced by a nonquad face, whereas each spoke edge is replaced by a quad face. An edge is a spoke edge if it touches a certain extraordinary vertex. To make the resulting mesh conforming, additional vertices and edges are further replaced by certain quad faces; see Figure 1(B). All the edges of each nonquad face are assigned with a zero knot interval. Under the assumption of knot intervals, this means that all the newly added faces have a zero (parametric) measure. In regular regions, introducing zero-knot-interval edges leads to a reduction in continuity of basis functions from C^2 to C^1.

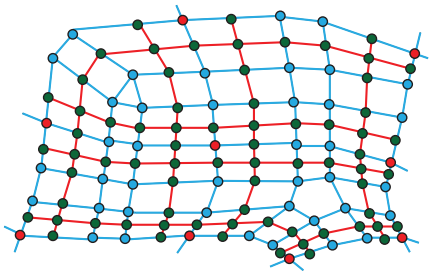
The topological rule for the subsequent levels defines how to split different types of faces in a given hybrid mesh. There are three types of faces depending on the knot intervals of their edges. First, for a zero-measure face whose edges all have a zero knot interval, no split is needed. Thus, all the nonquad faces fall in this case and stay unchanged. Next, if a zero-measure face has a pair of opposite edges with nonzero knot intervals, then the face is split into two. Finally, every nonzero-measure face is split into four subfaces. According to this rule, Figure 1(C) shows the (topological) split of the hybrid mesh in Figure 1(B).



(A) Input quad mesh



(B) The first level



(C) The subsequent levels

FIGURE 1 The topological steps of tuned hybrid nonuniform subdivision. (A) The input quad mesh, (B) converting the input quad mesh (light gray dots and lines) to the first-level hybrid mesh (black lines, and blue and red dots), and (C) refinement of (B) to obtain a subsequent-level hybrid mesh

2.2 | Geometric step

We next introduce the geometric step of tHNUS to compute the vertex coordinates in the hybrid mesh. At the first level, the rule to update regular vertices is the same as NURBS refinement, whereas the rule to compute vertices of a nonquad face plays a crucial role in shape quality and thus needs careful treatment. Two options are available:[14] a sophisticated but rather complicated treatment, and a simple explicit solution. The sophisticated solution is derived such that the limit surface of tHNUS has the same limit point and tangent plane as that of the nonuniform subdivision via eigen-polyhedron,[34] which is taken as the reference because it shows demonstrated shape quality under nonuniform parameterization. However, the computation is rather complicated and there is no explicit formula available.

Alternatively, the simple explicit solution defines each nonquad vertex as a convex combination of certain points in the input quad mesh. We primarily adopt this treatment in the article because the first-level geometric rule has nothing to do with the proof of surface G^1 continuity or the convergence performance in IGA. Interested readers may refer to Reference 14 for the eigen-polyhedron-based treatment. With reference to Figure 2(A), the explicit first-level geometric rule is given as

$$\begin{cases} P_i^{1,1} = F_i, \\ P_i^{1,0} = p_i F_i + (1-p_i) E_i, \\ P_i^{0,1} = q_i F_i + (1-q_i) E_{i+1}, \\ P_i^{0,0} = p_i q_i F_i + (1-p_i) q_i E_i + p_i (1-q_i) E_{i+1} + (1-p_i)(1-q_i) V, \end{cases} \qquad p_i = \frac{d_{i-1}}{d_{i-1}+d_{i+1}+a_{i+1}}, \quad q_i = \frac{d_{i+2}}{d_{i+2}+d_i+a_i},$$

where E_i, E_{i+1}, F_i, and V are the vertices of the ith face (indexing local to each extraordinary vertex) in the input quad mesh, $P_i^{k,l}$ $(k,l \in \{0,1\})$ are the vertices of the ith face in the first-level hybrid mesh, and p_i and q_i are coefficients computed from knot intervals a_i, d_i, and so forth.

Next, we provide the geometric rule of tHNUS for the subsequent levels. Referring to Figure 2(B,C) and given knot intervals a_i, d_i, the points $\overline{P}_i^{0,0}$, $\overline{P}_i^{1,0}$, $\overline{P}_i^{1,1}$ and $\overline{P}_i^{0,1}$ in the refined hybrid mesh are defined as

$$\begin{cases} \overline{P}_i^{0,0} = (1-\lambda) C + \lambda P_i^{0,0} + 2\lambda\alpha_i \left(-n P_i^{0,0} + \sum_{j=0}^{n-1} \left(1 + 2\cos\left(\frac{2(j-i)\pi}{n}\right)\right) P_j^{0,0} \right), \\ \overline{P}_i^{1,1} = \dfrac{d_i d_{i+1} P_i^{1,1} + d_i(d_{i+1}+2a_{i+1}) P_i^{1,0} + d_{i+1}(d_i+2a_i) P_i^{0,1} + (d_i+2a_i)(d_{i+1}+2a_{i+1}) P_i^{0,0}}{4(d_i+a_i)(d_{i+1}+a_{i+1})}, \\ \overline{P}_i^{1,0} = \dfrac{d_i d_{i+1} P_{i-1}^{0,1} + d_i(d_{i+1}+2d_{i-1}) P_i^{1,0} + d_{i+1}(d_i+2a_i) P_{i-1}^{0,0} + (d_i+2a_i)(d_{i+1}+2d_{i-1}) P_i^{0,0}}{4(d_{i-1}+d_{i+1})(d_i+a_i)}, \\ \overline{P}_{i-1}^{0,1} = \dfrac{d_i d_{i-1} P_i^{1,0} + d_i(2d_{i+1}+d_{i-1}) P_{i-1}^{0,1} + d_{i-1}(d_i+2a_i) P_i^{0,0} + (d_i+2a_i)(2d_{i+1}+d_{i-1}) P_{i-1}^{0,0}}{4(d_{i-1}+d_{i+1})(d_i+a_i)}, \end{cases} \tag{1}$$

where $\lambda \in \left(\frac{1}{4}, 1\right)$, $\alpha_j = \frac{1}{n} \frac{d_{j-1} d_{j+2}}{(d_{j-1}+d_{j+1})(d_j+d_{j+2})}$, and

$$C = \frac{\sum_{i=0}^{n-1} \left(d_i P_{i+1}^{0,0} + d_{i+2} P_i^{0,0}\right)(d_{i-1}+d_{i+3})}{\sum_{j=0}^{n-1} \left(d_j + d_{j+2}\right)\left(d_{j-1}+d_{j+3}\right)} \doteq \sum_{i=0}^{n-1} \beta_i P_i^{0,0}. \tag{2}$$

Points $P_i^{k,l}$ denote those in the given hybrid mesh, and the range of λ will be explained in the following section. The remaining points are computed by the NURBS mid-knot insertion. For example,

$$\overline{P}_i^{2,0} = \frac{a_i \overline{P}_i^{1,0}}{2(d_i+a_i)} + \frac{1}{4} \frac{(d_{i+1}+2d_{i-1}) P_i^{1,0} + d_{i+1} P_{i-1}^{0,1}}{d_{i+1}+d_{i-1}} + \frac{d_i \overline{P}_i^{3,0}}{2(d_i+a_i)},$$

$$\overline{P}_i^{2,1} = \frac{a_i \overline{P}_i^{1,1}}{2(d_i+a_i)} + \frac{1}{4} \frac{d_{i+1} P_i^{1,1} + (d_{i+1}+2a_{i+1}) P_i^{1,0}}{d_{i+1}+a_{i+1}} + \frac{d_i \overline{P}_i^{3,1}}{2(d_i+a_i)},$$

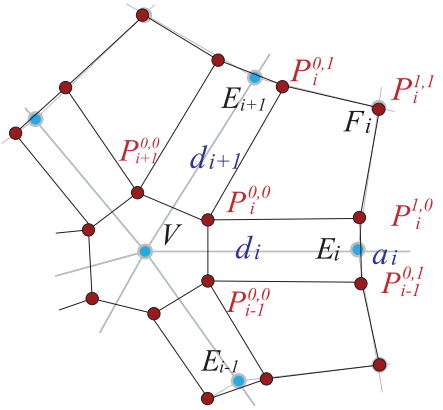
$$\overline{P}_i^{2,2} = \frac{P_i^{1,1}}{4} + \frac{a_i a_{i+1}\overline{P}_i^{1,1} + d_i a_{i+1}\overline{P}_i^{3,1} + a_i d_{i+1}\overline{P}_i^{1,3} + d_i d_{i+1}\overline{P}_i^{3,3}}{4(d_i + a_i)(d_{i+1} + a_{i+1})} + \frac{a_{i+1}}{4(d_{i+1} + a_{i+1})}M_1 + \frac{d_{i+1}}{4(d_{i+1} + a_{i+1})}M_3 + \frac{d_i}{4(d_i + a_i)}M_2 + \frac{a_i}{4(d_i + a_i)}M_4, \tag{3}$$

where

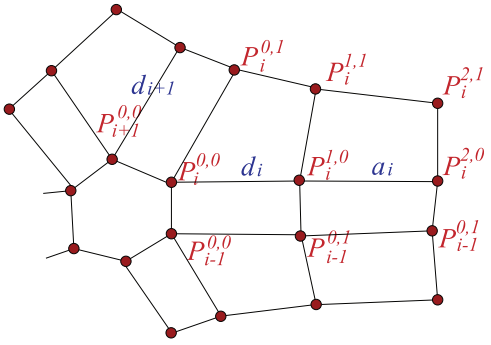
$$M_1 = \frac{d_{i+1}P_i^{1,1} + (d_{i+1} + 2a_{i+1})P_i^{1,0}}{2(d_{i+1} + a_{i+1})}, \qquad M_2 = \frac{P_i^{1,1} + P_i^{2,1}}{2},$$
$$M_4 = \frac{P_i^{1,1} + (d_i + 2a_i)P_i^{0,1}}{2(d_i + a_i)}, \qquad M_3 = \frac{P_i^{1,1} + P_i^{1,2}}{2}. \tag{4}$$

Remark 1. All the computations of these points are the same as those in HNUS except for $\overline{P}_i^{0,0}$, the vertices of nonquad faces, where the tuning parameter λ is introduced to control the size of shrinkage in an updated polygon: the smaller λ is, the more the polygon shrinks. As a result, isoparametric lines become more concentrated around extraordinary vertices. We will see such examples in Section 5. When $\lambda = \frac{1}{2}$, the computation of $\overline{P}_i^{0,0}$ coincides with that in HNUS, and thus tHNUS is equivalent to HNUS in this particular case. However, the generalization via introducing λ is not as straightforward as it appears. The key insight is that λ turns out to be the subdominant eigenvalues (i.e., the second and third eigenvalues) of the subdivision matrix in tHNUS. As has been reported,[20] convergence behavior in a subdivision scheme is mostly influenced by the subdominant eigenvalues. Therefore, tuning λ is equivalent to "controlling" convergence. We will have more detailed discussion about how λ improves convergence with specific examples in Section 5.

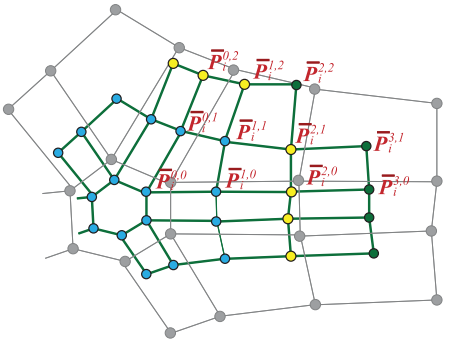
Remark 2. We introduce λ *explicitly* to the formula of $\overline{P}_i^{0,0}$. Note that λ is a single parameter for all situations. It is independent of the valence of extraordinary vertices and the choice of knot intervals. However, this is only possible when



(A) Computation of points in the first-level hybrid mesh



(B) A given hybrid mesh



(C) The refined hybrid-mesh

FIGURE 2 The geometric steps of tuned hybrid nonuniform subdivision. (A) Computation of points ($P_i^{0,0}$, $P_i^{1,0}$, $P_i^{0,1}$, and $P_i^{1,1}$) in the first-level hybrid mesh from those (V, E_i, E_{i+1}, and F) in the input quad mesh; (B) a given hybrid mesh with knot intervals d_i, a_i and d_{i+1}; and (C) the refined hybrid mesh of (B)

bounded curvature is not of primary interest. tHNUS generally does not have bounded curvature. Nonetheless, bounded curvature under nonuniform parameterization remains an open problem and may not be available at all.

3 | PROOF OF CONTINUITY

In order to prove tHNUS surfaces to be G^1 continuous, we need to prove that the spectrum of the subdivision matrix satisfies certain constraints and the associated characteristic map is regular and injective. Note that introducing λ to HNUS indeed significantly complicates the proof of G^1 continuity. Referring to Figure 3 for the notations, the subdivision rule can be written into the following equations since the neighbor knot intervals a_i equals to d_i with enough subdivision levels.

$$\begin{cases} \overline{P}_j^{0,0} = (1-\lambda)C + \lambda P_j^{0,0} + 2\lambda\alpha_j \left[-nP_j^{0,0} + \sum_{i=0}^{n-1} \left(1 + 2\cos\left(\frac{2(j-i)\pi}{n}\right)\right) P_i^{0,0} \right], \\ \overline{P}_j^{1,0} = \frac{3\left(2d_{j-1}+d_{j+1}\right)}{8\left(d_{j-1}+d_{j+1}\right)} P_j^{0,0} + \frac{3d_{j+1}}{8\left(d_{j-1}+d_{j+1}\right)} P_{j-1}^{0,0} + \frac{\left(2d_{j-1}+d_{j+1}\right)}{8\left(d_{j-1}+d_{j+1}\right)} P_j^{1,0} + \frac{d_{j+1}}{8\left(d_{j-1}+d_{j+1}\right)} P_{j-1}^{0,1}, \\ \overline{P}_{j-1}^{0,1} = \frac{3d_{j-1}}{8\left(d_{j-1}+d_{j+1}\right)} P_j^{0,0} + \frac{3\left(d_{j-1}+2d_{j+1}\right)}{8\left(d_{j-1}+d_{j+1}\right)} P_{j-1}^{0,0} + \frac{d_{j-1}}{8\left(d_{j-1}+d_{j+1}\right)} P_j^{1,0} + \frac{\left(d_{j-1}+2d_{j+1}\right)}{8\left(d_{j-1}+d_{j+1}\right)} P_{j-1}^{0,1}, \\ \overline{P}_j^{1,1} = \frac{9}{16} P_j^{0,0} + \frac{3}{16} P_j^{0,0} + \frac{3}{16} P_j^{0,0} + \frac{1}{16} P_j^{1,1}. \end{cases} \tag{5}$$

We arrange them in a matrix form $\overline{M} = S_n M$, that is,

$$\begin{pmatrix} \overline{P}_0^{0,0} \\ \vdots \\ \overline{P}_{n-1}^{0,0} \\ \overline{P}_0^{1,0} \\ \vdots \\ \overline{P}_{n-1}^{0,1} \\ \overline{P}_0^{1,1} \\ \vdots \\ \overline{P}_{n-1}^{1,1} \end{pmatrix} = \begin{pmatrix} & & & & & & & & \\ Q_n & & 0 & & & 0 & & & \\ & & & & & & & & \\ & E_0 & \cdots & 0 & & & & & \\ \divideontimes & \vdots & \ddots & 0 & & 0 & & & \\ & 0 & 0 & E_{n-1} & & & & & \\ & & & & \frac{1}{16} & 0 & 0 & & \\ \divideontimes & & \divideontimes & & \vdots & \ddots & & & \\ & & & & 0 & 0 & \frac{1}{16} & & \end{pmatrix} \begin{pmatrix} P_0^{0,0} \\ \vdots \\ P_{n-1}^{0,0} \\ P_0^{1,0} \\ \vdots \\ P_{n-1}^{0,1} \\ P_0^{1,1} \\ \vdots \\ P_{n-1}^{1,1} \end{pmatrix}. \tag{6}$$

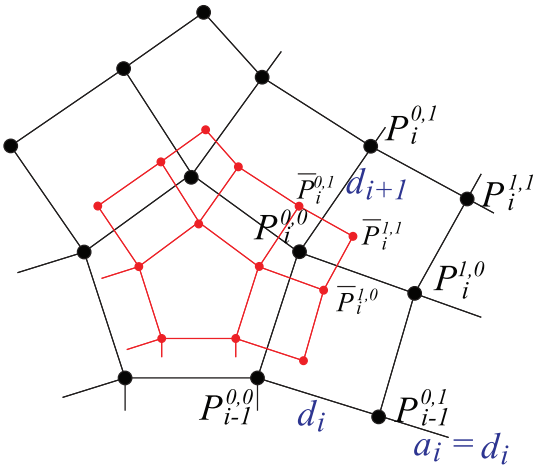


FIGURE 3 The notations to define the subdivision matrix around a nonquad face

Denote $Q_n = (Q_{i,j})$, where $i, j \in \{0, 1, \dots, n-1\}$, and then we have

$$Q_{i,j} = \begin{cases} (1-\lambda)\beta_j + 2\left(1 + 2\cos\left(\frac{2(j-i)\pi}{n}\right)\right)\lambda\alpha_i, & j \neq i \\ \lambda + (1-\lambda)\beta_i - 2(n-3)\,\lambda\alpha_i, & j = i \end{cases}$$

and

$$E_j = \begin{pmatrix} \frac{2d_{j-1}+d_{j+1}}{8(d_{j-1}+d_{j+1})} & \frac{d_{j+1}}{8(d_{j-1}+d_{j+1})} \\ \frac{d_{j-1}}{8(d_{j-1}+d_{j+1})} & \frac{d_{j-1}+2d_{j+1}}{8(d_{j-1}+d_{j+1})} \end{pmatrix}.$$

Lemma 1. *Given an extraordinary vertex of any valence and an arbitrary choice of positive knot intervals, $1 > \lambda > \frac{1}{4}$, the eigenvalues of Q_n satisfy*

$$\lambda_1 = 1 > \lambda_2 = \lambda_3 = \lambda > |\lambda_k|, k = 4, 5, \dots, n. \tag{7}$$

Proof. We use the discrete Fourier transform to compute the eigenvalues of Q_n. Let p_k and $\overline{p}_k$ ($k = 0, \dots, n-1$) be the Fourier vectors corresponding to P_j and $\overline{P}_j$, respectively, that is,

$$p_k = \frac{1}{n}\sum_{j=0}^{n-1} P_j^{0,0}\overline{\omega}^{jk}, \overline{p}_k = \frac{1}{n}\sum_{j=0}^{n-1} \overline{P}_j^{0,0}\overline{\omega}^{jk}, \tag{8}$$

$$P_k^{0,0} = \sum_{j=0}^{n-1} p_j\omega^{jk}, \overline{P}_k^{0,0} = \sum_{j=0}^{n-1} \overline{p}_j\omega^{jk}, \tag{9}$$

where $\omega = e^{\frac{2\pi}{n}\iota}$ and $\overline{\omega} = e^{-\frac{2\pi}{n}\iota}$ (ι is the imaginary unit). Now the subdivision rule can be formulated in terms of the Fourier vectors,

$$\sum_{k=0}^{n-1}\overline{p}_k\omega^{jk} = \sum_{k=0}^{n-1}\left(\sum_{j=0}^{n-1}(1-\lambda)\beta_j\omega^{jk}\right)p_k + p_0 + \lambda\omega^j p_1 + \lambda\omega^{j(n-1)}p_{n-1} + 2\lambda\left(\frac{1}{2} - n\alpha_j\right)\sum_{k=2}^{n-2} p_k\omega^{jk}. \tag{10}$$

Using the inverse discrete Fourier transform, we obtain

$$\begin{pmatrix} \overline{p}_0 \\ \overline{p}_1 \\ \vdots \\ \overline{p}_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & (1-\lambda)\beta_1 & \dots & (1-\lambda)\beta_{n-1} \\ 0 & \lambda & \divideontimes & 0 \\ \vdots & \vdots & B_{n-3} & \vdots \\ 0 & 0 & \divideontimes & \lambda \end{pmatrix}\begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \end{pmatrix}, \tag{11}$$

where

$$B_{n-3} = \lambda I - 2\lambda\begin{pmatrix} \sum_{j=0}^{n-1}\alpha_j & \sum_{j=0}^{n-1}\alpha_j\omega^j & \dots & \sum_{j=0}^{n-1}\alpha_j\omega^{(n-4)j} \\ \sum_{j=0}^{n-1}\alpha_j\omega^{(n-1)j} & \sum_{j=0}^{n-1}\alpha_j & \dots & \sum_{j=0}^{n-1}\alpha_j\omega^{(n-5)j} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=0}^{n-1}\alpha_j\omega^{4j} & \sum_{j=0}^{n-1}\alpha_j\omega^{5j} & \dots & \sum_{j=0}^{n-1}\alpha_j \end{pmatrix} =: \lambda I - 2\lambda G_{n-3}. \tag{12}$$

We can find that the matrix Q_n has its first three eigenvalues 1, λ, and λ. Although the eigenvalues μ_k ($k = 1, \dots, n-3$) of G_{n-3} cannot be computed explicitly, we can conclude that $0 < \mu_k < 1$ according to Reference 14. Denote $\lambda_{B,i}$ to be

the eigenvalues of B_{n-3}. As G_{n-3} is positive definite, $\frac{1}{2}I - \frac{1}{2\lambda}B_{n-3}$ $(=G_{n-3})$ is also a positive definite matrix, which means that $\lambda_{B,i} < \lambda$. On the other hand, $I - G_{n-3}$ is a positive definite matrix as well because $\mu_k < 1$, and equivalently, $I - \left(\frac{1}{2}I - \frac{1}{2\lambda}B_{n-3}\right)$ is positive definite, which means that $\lambda_{B,i} > -\lambda$. Therefore, we complete the proof. ∎

Lemma 2. *Given an extraordinary vertex of any valence and an arbitrary choice of positive knot intervals, if* $1 > \lambda > \frac{1}{4}$, *then the eigenvalues of* S_n *satisfy*

$$\lambda_1 = 1 > \lambda_2 = \lambda_3 = \lambda > |\lambda_k|, \text{where } k = 4, 5, \ldots, 4n. \tag{13}$$

Proof. The eigenvalues of S_n consist of those of Q_n, E_i, and $\frac{1}{16}I_n$, where I_n is an $n \times n$ identity matrix. As proved in Lemma 1, the first three eigenvalues of Q_n are 1, λ, λ, and the remaining ones are less than λ. $\frac{1}{16}I_n$ has n equal eigenvalues $\frac{1}{16}$ $(< \lambda)$. It is also straightforward to verify that the eigenvalues of the 2×2 matrix E_i are $\frac{1}{4}$ $(< \lambda)$ and $\frac{1}{8}$ $(< \lambda)$. Therefore, we conclude that the eigenvalues of S_n are

$$\lambda_1 = 1 > \lambda_2 = \lambda_3 = \lambda > |\lambda_k|, \text{where } k = 4, 5, \ldots, 4n. \tag{14}$$

∎

The next step is to compute the characteristic map and prove that it is regular and injective. We first prove the following lemma.

Lemma 3. *Let* $P_i = \left(\cos\left(\frac{2i\pi}{n}\right), \sin\left(\frac{2i\pi}{n}\right)\right) \in \mathbb{R}^2$ $(i = 0, \ldots, n-1)$, $C_P = \sum_{i=0}^{n-1} \beta_i P_i$, *and* P *be an* $n \times 2$ *vector containing all* P_i, *that is,* $P = [P_0, P_1, \ldots, P_{n-1}]^T$. *Then we have*

$$S_n(P - C_P) = \lambda(P - C_P). \tag{15}$$

Proof. Denote $\overline{P} = S_n P$, and we can obtain

$$\begin{aligned}
\overline{P}_j - C_P &= \lambda\left(P_j - C_P\right) + 2\lambda\alpha_j\left[-n\left(\cos\left(\frac{2j\pi}{n}\right), \sin\left(\frac{2j\pi}{n}\right)\right) + \sum_{i=0}^{n-1}\left(1 + 2\cos\left(\frac{2(j-i)\pi}{n}\right)\right)\left(\cos\left(\frac{2i\pi}{n}\right), \sin\left(\frac{2i\pi}{n}\right)\right)\right] \\
&= \lambda\left(P_j - C_P\right) + 2\lambda\alpha_j\left[-n\left(\cos\left(\frac{2j\pi}{n}\right), \sin\left(\frac{2j\pi}{n}\right)\right) + \sum_{i=0}^{n-1} 2\cos\left(\frac{2(j-i)\pi}{n}\right)\left(\cos\left(\frac{2i\pi}{n}\right), \sin\left(\frac{2i\pi}{n}\right)\right)\right] \\
&= \lambda\left(P_j - C_P\right) + 2\lambda\alpha_j\Bigg[-n\left(\cos\left(\frac{2j\pi}{n}\right), \sin\left(\frac{2j\pi}{n}\right)\right) \\
&\quad + \sum_{i=0}^{n-1}\left(\cos\left(\frac{2j\pi}{n}\right) + \cos\left(\frac{2(j-2i)\pi}{n}\right), \sin\left(\frac{2j\pi}{n}\right) - \sin\left(\frac{2(j-2i)\pi}{n}\right)\right)\Bigg] \\
&= \lambda\left(P_j - C_P\right).
\end{aligned}$$

Since the above equation holds for any $0 \le j \le n-1$, we conclude

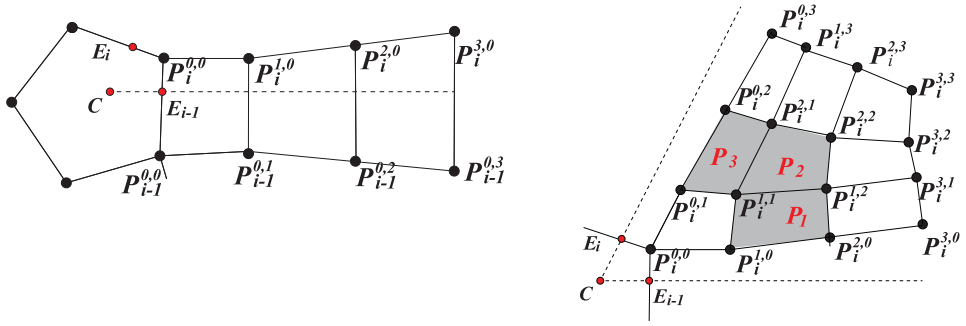
$$S_n(P - C_P) = \lambda(P - C_P). \tag{16}$$

∎

Lemma 4. *The characteristic map of tHNUS is regular and injective for any valence extraordinary vertices and any positive knot intervals if* $\lambda \in \left(\frac{1}{4}, 1\right)$.

Proof. To prove that the characteristic map is regular and injective, we need a 4×4 grid of control points. We first compute the coordinates of this control grid that is used to define the characteristic map. The key idea is based on the fact that applying subdivision to the control grid of a characteristic map is equivalent to scaling the control grid by λ.

Referring to Figure 4, we have control points $P_i^{j,k}$, where $0 \le j, k \le 3$ $(0 \le i \le n-1)$. According to Lemma 3, if we let $P_i^{0,0} = \left(\cos\left(\frac{2i\pi}{n}\right), \sin\left(\frac{2i\pi}{n}\right)\right) \in \mathbb{R}^2$, $C = \sum_{i=0}^{n-1} \beta_i P_i^{0,0}$, then we have $S_n[P_0^{0,0} - C, \ldots, P_{n-1}^{0,0} - C]^T = \lambda[P_0^{0,0} - C, \ldots, P_{n-1}^{0,0} - C]^T$.



(A) Control points $P_i^{j,k}$, $j = 0$ or $k = 0$ (B) The other control points

FIGURE 4 The control points of the characteristic map of tuned hybrid nonuniform subdivision. (A) Shows the control points P_i^{0j} and P_i^{j0} while (B) shows the rest of the control points of the characteristic map

Further let $E_i = \frac{d_i}{d_i+d_{i+2}} P_{i+1}^{0,0} + \frac{d_{i+2}}{d_i+d_{i+2}} P_i^{0,0}$, $p = P_i^{0,0} - C$, $v = E_{i-1} - C$ and $w = E_i - C$. By definition, we have

$$\frac{1}{4}\left(\frac{d_{i+1}+2d_{i-1}}{2d_{i+1}+2d_{i-1}}\left(P_i^{1,0}-P_i^{0,0}\right)+\frac{d_{i+1}}{2d_{i+1}+2d_{i-1}}\left(P_{i-1}^{0,1}-P_{i-1}^{0,0}\right)\right)+\frac{1}{2}(E_{i-1}-C)=\lambda\left(P_i^{1,0}-P_i^{0,0}\right),$$
$$\frac{1}{4}\left(\frac{d_{i-1}}{2d_{i+1}+2d_{i-1}}\left(P_i^{1,0}-P_i^{0,0}\right)+\frac{d_{i-1}+2d_{i+1}}{2d_{i+1}+2d_{i-1}}\left(P_{i-1}^{0,1}-P_{i-1}^{0,0}\right)\right)+\frac{1}{2}(E_{i-1}-C)=\lambda\left(P_{i-1}^{0,1}-P_{i-1}^{0,0}\right).$$

Solving the linear systems, we obtain

$$P_i^{1,0}-P_i^{0,0}=P_{i-1}^{0,1}-P_{i-1}^{0,0}=\frac{4(1-\lambda)}{4\lambda-1}v+\frac{4(1-2\lambda)}{8\lambda-1}(p-v). \tag{17}$$

Similarly, we compute $P_i^{2,0}$, $P_i^{3,0}$, $P_{i-1}^{0,3}$, $P_{i-1}^{0,3}$ as follows,

$$P_i^{2,0}-P_i^{1,0}=\frac{18(1-\lambda)}{(8\lambda-1)(4\lambda-1)}v+\frac{18(1-2\lambda)}{(16\lambda-1)(8\lambda-1)}(p-v),$$
$$P_i^{3,0}-P_i^{2,0}=\frac{6(1-\lambda)(1+\lambda)}{(8\lambda-1)\lambda(4\lambda-1)}v+\frac{3\left(1-4\lambda^2\right)}{(16\lambda-1)\lambda(8\lambda-1)}(p-v).$$

We can also compute the remaining control points $P_i^{j,k}$ $(1 \le j,k \le 3)$, whose coefficients are complex expressions in λ. The detailed expressions are given in the Appendix.

With all these control points, we can now extract the Bézier control points for patches P_1, P_2, and P_3; see Figure 4(B). For example, in the patch P_2, let $B_2^{j,k}$ $(j,k=0,\ \dots\ ,3)$ be the 4×4 Bézier control points. We denote $S_2^{j,k} = B_2^{j+1,k} - B_2^{j,k}$ and $T_2^{j,k} = B_2^{j,k+1} - B_2^{j,k}$. All $S_2^{j,k}$ and $T_2^{j,k}$ can be written as linear combinations of p, v and w, where the coefficients are again complex expressions in λ; see Appendix. We further plot some of these coefficients as functions of $\lambda \in \left(\frac{1}{4}, 1\right)$; see Figures 5 and 6. We observe that $S_2^{j,k}$ are convex combinations of vectors p, v and $-w$, while $T_2^{j,k}$ are convex combinations of p, $-v$ and w. Moreover, C is a convex combination of the points E_i from Equation (2), so the patch P_2 is regular and injective. As a result, all the control points $P_i^{j,k}$ $(0 \le j,k \le 3)$ lie in the region bounded by two rays CE_{i-1} and CE_i, which means that any two different patches must not intersect with one another. Similar results can also be achieved for patches P_1 and P_3. Therefore, the characteristic map of tHNUS is regular and injective for any $\lambda \in \left(\frac{1}{4}, 1\right)$, any valence extraordinary vertices and any positive knot intervals. ■

Theorem 1. *Given an arbitrary 2-manifold control mesh with any choice of positive knot intervals and any $\lambda \in \left(\frac{1}{4}, 1\right)$, the corresponding tHNUS limit surface is globally G^1-continuous.*

Proof. The theorem is a direct result of Lemmas 1,2, and 4. ■

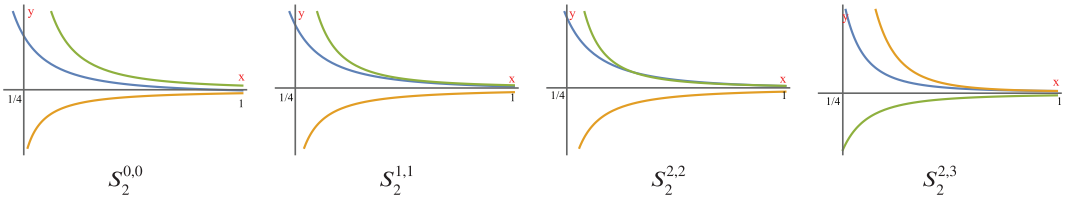


FIGURE 5 The plots of the coefficients of $S_2^{j,k}$ in terms of $\lambda \in \left(\frac{1}{4}, 1\right)$, where the x-axis represents λ and y-axis represents the value of the coefficients. The green, blue and orange lines represent coefficients corresponding to v, p, and w, respectively. Each $S_2^{j,k}$ is a convex combination of p, v, and $-w$

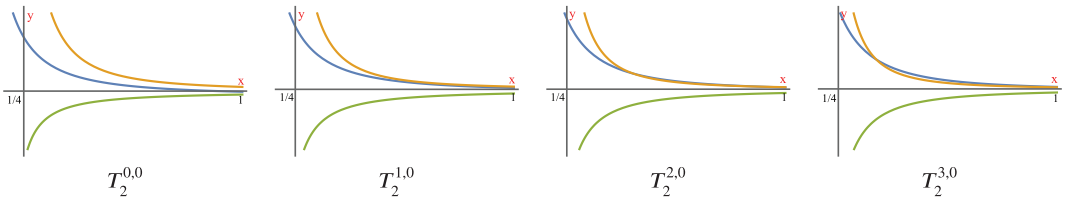


FIGURE 6 The plots of the coefficients of $T_2^{j,k}$ in terms of $\lambda \in \left(\frac{1}{4}, 1\right)$, where the x-axis represents λ and y-axis represents the value of the coefficients. The green, blue and orange lines represent coefficients corresponding to v, p, and w, respectively. Each $T_2^{j,k}$ is a convex combination of p, $-v$, and w

4 | HYBRID SUBDIVISION BASIS FUNCTIONS

In this section, we introduce basis functions of HNUS. The derivation of such subdivision functions essentially follows Stam's method for Catmull–Clark subdivision.[40] However, there are two major differences. First, Catmull–Clark basis functions are associated with the input quad control mesh, whereas tHNUS basis functions are associated with the hybrid control mesh; see Figure 1(A). Second, Catmull–Clark subdivision features uniform knot intervals everywhere, leading to a subdivision matrix that only depends on the valence of a particular extraordinary vertex. By contrast, tHNUS (or HNUS) supports general nonuniform knot intervals, so the subdivision matrix depends not only on the valence of the extraordinary vertex, but also on the surrounding knot intervals.

4.1 | Definition of basis functions

We now introduce how tHNUS basis functions are defined on a hybrid control mesh. We start with distinguishing different types of faces. Recall that there exists both quad and nonquad faces in the hybrid mesh, and each edge in a nonquad face is assigned with a zero knot interval by construction. The knot intervals of other edges inherit from the input quad mesh and are constrained by the assumption that opposite edges in a quad face have the same knot interval. Moreover, note that edges perpendicular to the boundary also have zero knot intervals to make use of open knot vectors. An example of the knot interval configuration is shown in Figure 7(B), where the hybrid mesh is obtained from the input mesh in Figure 7(A).

We identify faces of zero-measure and nonzero-measure according to their parametric areas, which are computed using knot intervals. Zero-measure faces are not used in geometric representation and have no contribution to analysis. Note that all the nonquad faces and boundary faces have a zero-measure. The nonzero-measure faces, on the other hand, are divided into *regular* and *irregular* faces. An irregular face is a nonzero-measure face that shares a vertex with a certain nonquad face; all the other nonzero-measure faces are regular; see Figure 7(C). The tHNUS basis functions defined on a regular element[1] are simply B-spline basis functions. In what follows, we restrict our attention to those defined on an irregular element. For simplicity of explanation, we assume that there is only one nonquad face in the 1-ring neighborhood

[1]We use *face* and *element* interchangeably, but "face" emphasizes mesh topology whereas "element" is IGA-oriented.

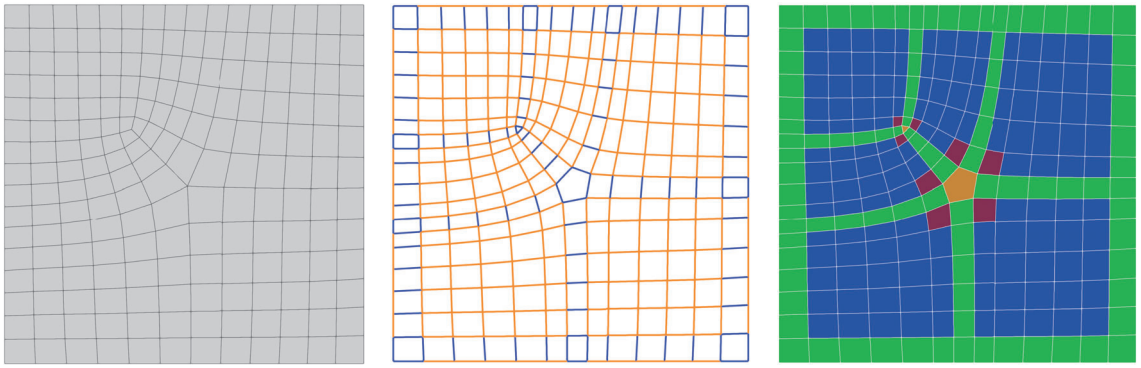


FIGURE 7 Knot intervals and mesh terminologies. (A) The input quad mesh, (B) edges with zero knot intervals (blue) and nonzero intervals (orange), and (C) different types of faces: quad faces with zero-measure (green), nonquad faces (orange), regular faces (blue), and irregular faces (red)

of an irregular element. The 1-ring neighborhood of a face is a collection of faces that share vertices with this face, and recursively, the n-ring ($n \geq 2$) neighborhood consists of faces in the $(n-1)$-ring neighborhood as well as the faces sharing vertices with the $(n-1)$-ring neighborhood.

Remark 3. In a hybrid control mesh, each interior vertex is shared by four faces (or edges) and thus it has a regular valence of four. However, it does not mean that mesh irregularities are removed by converting the input quad mesh to its hybrid counterpart. In fact, irregularities are now manifested in the nonquad faces, which will be detailed in the following.

Given an irregular element Ω, let N denote the number of vertices in its adjacent nonquad face which is equivalent to the valence of the corresponding extraordinary vertex in the input quad mesh. There are $K := N + 12$ basis functions defined on Ω, associated with a local mesh around the nonquad face; see Figure 8(A). We denote

$$\mathbf{B}_0(u,v) = [B_{0,1}(u,v), B_{0,2}(u,v), \ \dots \ , B_{0,K}(u,v)]^T$$

and

$$\mathbf{P}_0 = [P_{0,1}, P_{0,2}, \ \dots \ , P_{0,K}]^T$$

the basis functions and the corresponding control vertices, respectively. Their indices are ordered according to Figure 8(A). The surface patch, that is, the geometric mapping restricted to Ω is then represented by

$$s(u,v) = \mathbf{P}_0^T \mathbf{B}_0(u,v), \quad (u,v) \in \Omega. \tag{18}$$

Note that Ω naturally has a parametric domain $[0, d_1] \times [0, d_2]$ that is determined by the corresponding knot intervals. We rescale it to $\Omega = [0,1]^2$ to unify the treatment of irregular elements. The influence of the rescaling will be discussed in Remark 4.

Our focus is to derive $\mathbf{B}_0$, which relies on subdivision of the corresponding control mesh $\mathbf{P}_0$. Applying subdivision once yields Level-1 control vertices, denoted by

$$\begin{aligned} \mathbf{P}_1 &= [P_{1,1}, P_{1,2}, \ \dots \ , P_{1,K}]^T = \mathbf{S}_1 \mathbf{P}_0, \\ \overline{\mathbf{P}}_1 &= [P_{1,1}, P_{1,2}, \ \dots \ , P_{1,K}, P_{1,K+1}, \ \dots \ , P_{1,M}]^T = \overline{\mathbf{S}}_1 \mathbf{P}_0, \end{aligned} \tag{19}$$

where $M := K + 9 = N + 21$. The subdivision matrices $\mathbf{S}_1$ and $\overline{\mathbf{S}}_1$ have the dimension of $K \times K$ and $M \times K$, respectively. Clearly, $\overline{\mathbf{S}}_1$ yields additional nine vertices compared with $\mathbf{S}_1$. The entries of $\mathbf{S}_1$ and $\overline{\mathbf{S}}_1$ come from the tHNUS geometric rules as well as the mid-knot insertion of B-splines; see Equations (1,2) and (3,4), respectively. Among the four subelements at Level 1, three of them Ω_k^1 ($k = 1, 2, 3$) are regular and correspond to regular C^1 B-spline patches. In other words, the

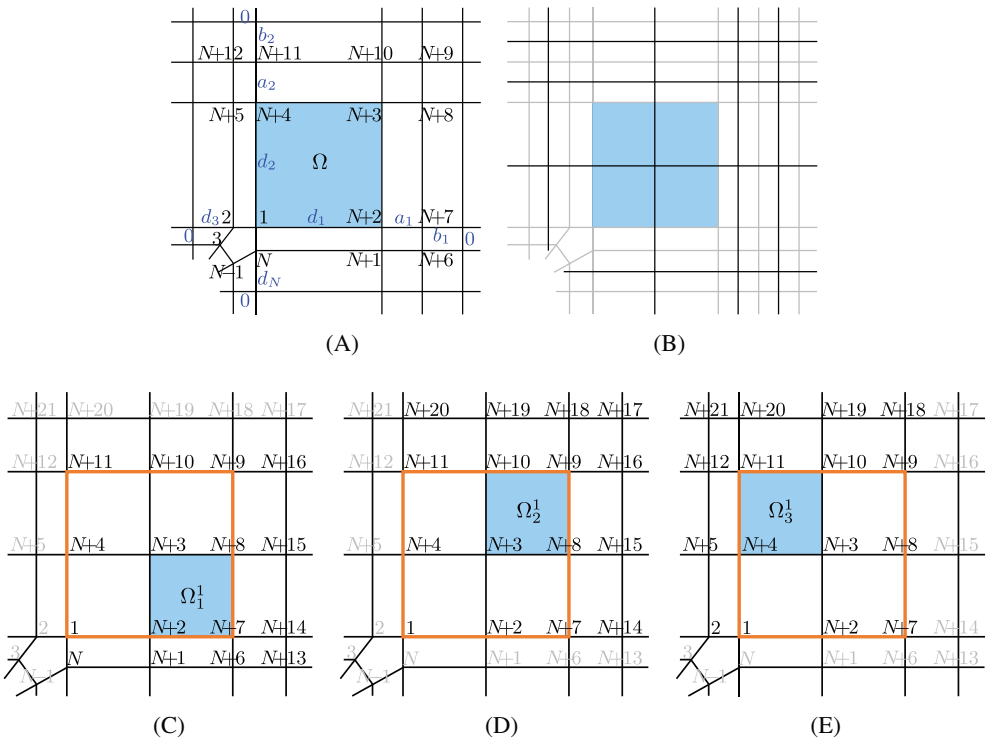


FIGURE 8 Local meshes of an irregular element Ω and its refined subelements Ω_k^1 ($k = 1, 2, 3$). (A) The local mesh and surrounding knot intervals of Ω, (B) the globally refined mesh, and (C–E) the local meshes of Ω_k^1, where the orange lines indicate the boundary of Ω, and indices in light gray imply that the corresponding basis functions have no support on the highlighted subelement

surface patch restricted to Ω_k^1 is given by

$$s(u,v) = \mathbf{P}_{1,k}^T \mathbf{N}_{1,k}(u,v), \quad (u,v) \in \Omega_k^1 \subset \Omega, \tag{20}$$

where $\mathbf{P}_{1,k}$ is a subvector of $\overline{\mathbf{P}}_1$ and $\mathbf{N}_{1,k}(u,v)$ is the vector of B-splines defined on Ω_k^1 ($k = 1, 2, 3$); see Figure 8(C–E). Both $\mathbf{P}_{1,k}$ and $\mathbf{N}_{1,k}$ have a dimension of 16 due to the bicubic degree setting. $\mathbf{P}_{1,k}$ can be obtained with the help of a permutation matrix $\mathbf{T}_k$, that is, $\mathbf{P}_{1,k} = \mathbf{T}_k \overline{\mathbf{P}}_1 = \mathbf{T}_k \overline{\mathbf{S}}_1 \mathbf{P}_0$. Therefore, Equation (20) becomes

$$s(u,v) = \left(\mathbf{T}_k \overline{\mathbf{S}}_1 \mathbf{P}_0\right)^T \mathbf{N}_{1,k}(u,v) = \mathbf{P}_0^T \left(\mathbf{T}_k \overline{\mathbf{S}}_1\right)^T \mathbf{N}_{1,k}(u,v), \quad (u,v) \in \Omega_k^1. \tag{21}$$

For Equations (18,21) to be equivalent under arbitrary choice of $\mathbf{P}^0$, we need

$$\mathbf{B}_0(u,v) = \left(\mathbf{T}_k \overline{\mathbf{S}}_1\right)^T \mathbf{N}_{1,k}(u,v), \quad (u,v) \in \Omega_k^1. \tag{22}$$

In other words, we have found the definition of $\mathbf{B}_0$ on three quarters (Ω_1^1, Ω_2^1, and Ω_3^1) of Ω.

Now we are left to find the definition of $\mathbf{B}_0$ on the remaining quarter $[0, \frac{1}{2}]^2$, and we proceed with the same idea explained above. As a result, the domain Ω is partitioned into an infinite series of tiles,

$$\Omega = \bigcup_{n=1}^{\infty} \bigcup_{k=1}^{3} \Omega_k^n,$$

where

$$\Omega_1^n = \left[\frac{1}{2^n}, \frac{1}{2^{n-1}}\right] \times \left[0, \frac{1}{2^n}\right],$$
$$\Omega_2^n = \left[\frac{1}{2^n}, \frac{1}{2^{n-1}}\right] \times \left[\frac{1}{2^n}, \frac{1}{2^{n-1}}\right],$$
$$\Omega_3^n = \left[0, \frac{1}{2^n}\right] \times \left[\frac{1}{2^n}, \frac{1}{2^{n-1}}\right].$$

Analogous to Equation (19), we have

$$\mathbf{P}_n = \mathbf{S}_n\mathbf{P}_{n-1} = \mathbf{S}_n\mathbf{S}_{n-1}\ \dots\ \mathbf{S}_1\mathbf{P}_0,$$
$$\overline{\mathbf{P}}_n = \overline{\mathbf{S}}_n\mathbf{P}_{n-1} = \overline{\mathbf{S}}_n\mathbf{S}_{n-1}\ \dots\ \mathbf{S}_1\mathbf{P}_0.$$

Again, $\mathbf{S}_n$ and $\overline{\mathbf{S}}_n$ $(n \geq 1)$ have the dimension of $K \times K$ and $M \times K$, respectively. Note that $\mathbf{S}_2 = \mathbf{S}_3 = \dots = \mathbf{S}_n$ $(n \geq 2)$ and $\overline{\mathbf{S}}_3 = \overline{\mathbf{S}}_4 = \dots = \overline{\mathbf{S}}_n$ $(n \geq 3)$ because the ratios of knot intervals around an irregular subelement become fixed as the subdivision level increases. Therefore, following the same argument in deriving Equation (22), we have the general expressions for $\mathbf{B}_0$,

$$\mathbf{B}_0(u,v) = \begin{cases} \left(\mathbf{T}_k\overline{\mathbf{S}}_1\right)^T\mathbf{N}_{1,k}(u,v) & n = 1 \\ (\mathbf{T}_k\overline{\mathbf{S}}_2\mathbf{S}_1)^T\mathbf{N}_{2,k}(u,v) & n = 2\ , \\ (\mathbf{T}_k\overline{\mathbf{S}}_3(\mathbf{S}_2)^{n-2}\mathbf{S}_1)^T\mathbf{N}_{n,k}(u,v) & n \geq 3 \end{cases}$$

where $(u,v) \in \Omega_k^n$, and $\mathbf{N}_{n,k}(u,v)$ $(n \geq 1)$ is the vector of B-splines defined on Ω_k^n.

Remark 4. Rescaling Ω to $[0,1]^2$ only affects the knot vectors (or equivalently, the vectors of knot intervals) of B-splines $\mathbf{N}_{n,k}(u,v)$ $(n \geq 1)$. For example, when $n = 1$ without scaling, the vector of knot intervals in the u direction is

$$\left\{\frac{d_3}{2}, \frac{d_3}{2}, 0, \frac{d_1}{2}, \frac{d_1}{2}, \frac{a_1}{2}, \frac{a_1}{2}, \frac{b_1}{2}\right\},$$

whereas after scaling with respect to d_1, it becomes

$$U_1 = \left\{\frac{d_3}{2d_1}, \frac{d_3}{2d_1}, 0, \frac{1}{2}, \frac{1}{2}, \frac{a_1}{2d_1}, \frac{a_1}{2d_1}, \frac{b_1}{2d_1}\right\},$$

and similarly, we have the vector of knot intervals in the v direction,

$$V_1 = \left\{\frac{d_N}{2d_2}, \frac{d_N}{2d_2}, 0, \frac{1}{2}, \frac{1}{2}, \frac{a_2}{2d_2}, \frac{a_2}{2d_2}, \frac{b_2}{2d_2}\right\}.$$

Moreover, when $n = 2$, we have

$$U_2 = \left\{\frac{d_3}{2^2d_1}, \frac{d_3}{2^2d_1}, 0, \frac{1}{2^2}, \frac{1}{2^2}, \frac{1}{2^2}, \frac{1}{2^2}, \frac{a_1}{2^2d_1}\right\},\ V_2 = \left\{\frac{d_N}{2^2d_2}, \frac{d_N}{2^2d_2}, 0, \frac{1}{2^2}, \frac{1}{2^2}, \frac{1}{2^2}, \frac{1}{2^2}, \frac{a_2}{2^2d_2}\right\},$$

and when $n \geq 3$,

$$U_n = \left\{\frac{d_3}{2^nd_1}, \frac{d_3}{2^nd_1}, 0, \frac{1}{2^n}, \frac{1}{2^n}, \frac{1}{2^n}, \frac{1}{2^n}, \frac{1}{2^n}\right\},\ V_n = \left\{\frac{d_N}{2^nd_2}, \frac{d_N}{2^nd_2}, 0, \frac{1}{2^n}, \frac{1}{2^n}, \frac{1}{2^n}, \frac{1}{2^n}, \frac{1}{2^n}\right\}.$$

$\mathbf{N}_{n,k}(u,v)$ are defined using U_n and V_n.

In fact, it is practically useful to rescale each tile Ω_k^n to $[0,1]^2$ by

$$\begin{aligned} k = 1 \quad & \xi = 2^nu - 1, \quad \eta = 2^nv, \\ k = 2 \quad & \xi = 2^nu - 1, \quad \eta = 2^nv - 1, \\ k = 3 \quad & \xi = 2^nu, \qquad \eta = 2^nv - 1. \end{aligned}$$

Correspondingly, the vectors of knot intervals are rescaled to

$$
\begin{aligned}
n=1 \quad & \Xi_1=\left\{\tfrac{d_3}{d_1},\tfrac{d_3}{d_1},0,1,1,\tfrac{a_1}{d_1},\tfrac{a_1}{d_1},\tfrac{b_1}{d_1}\right\}, \quad \Theta_1=\left\{\tfrac{d_N}{d_2},\tfrac{d_N}{d_2},0,1,1,\tfrac{a_2}{d_2},\tfrac{a_2}{d_2},\tfrac{b_2}{d_2}\right\},\\
n=2 \quad & \Xi_2=\left\{\tfrac{d_3}{d_1},\tfrac{d_3}{d_1},0,1,1,1,1,\tfrac{a_1}{d_1}\right\}, \qquad \Theta_2=\left\{\tfrac{d_N}{d_2},\tfrac{d_N}{d_2},0,1,1,1,1,\tfrac{a_2}{d_2}\right\},\\
n\geq 3 \quad & \Xi_3=\left\{\tfrac{d_3}{d_1},\tfrac{d_3}{d_1},0,1,1,1,1,1\right\}, \qquad \Theta_3=\left\{\tfrac{d_N}{d_2},\tfrac{d_N}{d_2},0,1,1,1,1,1\right\},
\end{aligned}
$$

where the rescaled knot intervals are now independent of the subdivision level n when $n\geq 3$. In summary, the basis functions of interest are defined as

$$
\mathbf{B}_0(u,v)=\begin{cases}
\left(\mathbf{T}_k\bar{\mathbf{S}}_1\right)^T\mathbf{b}_{1,k}(\xi(u),\eta(v)) & n=1\\
(\mathbf{T}_k\bar{\mathbf{S}}_2\mathbf{S}_1)^T\mathbf{b}_{2,k}(\xi(u),\eta(v)) & n=2\\
(\mathbf{T}_k\bar{\mathbf{S}}_3(\mathbf{S}_2)^{n-2}\mathbf{S}_1)^T\mathbf{b}_{3,k}(\xi(u),\eta(v)) & n\geq 3
\end{cases}, \tag{23}
$$

where $(\xi,\eta)\in[0,1]^2$ and $\mathbf{b}_{l,k}(\xi,\eta)$ are B-splines defined using Ξ_l and Θ_l $(l=1,2,3)$. Note that when $n=1,2$, we have $\mathbf{N}_{n,k}(u,v)=\mathbf{b}_{n,k}(\xi(u),\eta(v))$, and when $n\geq 3$, $\mathbf{N}_{n,k}(u,v)=\mathbf{b}_{3,k}(\xi(u),\eta(v))$.

Remark 5. Evaluation of $\mathbf{B}_0(u,v)$ at $(0,0)$ in an irregular element needs the computation of $\lim_{n\to\infty}(\mathbf{S}_2)^{n-2}$. Following Reference 40, we need to eigen-decompose $\mathbf{S}_2$ such that $\mathbf{S}_2=\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, where $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\mathbf{S}_2$ and $\mathbf{V}$ is an invertible matrix with columns being the corresponding eigenvectors. Accordingly, we have $\lim_{n\to\infty}(\mathbf{S}_2)^{n-2}=\lim_{n\to\infty}\mathbf{V}\mathbf{\Lambda}^{n-2}\mathbf{V}^{-1}$. Recall that all the eigenvalues are smaller than 1 (and greater than 0) except the first one $\lambda_1=1$, so $\lim_{n\to\infty}\mathbf{\Lambda}^{n-2}$ is a matrix whose entries are all zero except the first-row-first-column entry, which is one. On the other hand, the derivatives of $\mathbf{B}_0(u,v)$ are not bounded around $(0,0)$. We can see this by applying the chain rule, for example,

$$
\frac{\partial\mathbf{B}_0(u,v)}{\partial u}=2^n\left(\mathbf{T}_k\bar{\mathbf{S}}_3(\mathbf{S}_2)^{n-2}\mathbf{S}_1\right)^T\frac{\partial\mathbf{b}_{3,k}(\xi,\eta)}{\partial\xi},
$$

where the factor comes from $d\xi/du=2^n$. A differentiable version of $\mathbf{B}_0$ (with respect to certain parameters) can be obtained via characteristic-map-based reparameterization.[41] However, in this article, we are interested in applying tHNUS basis functions in the context of IGA, so we only need derivatives at quadrature points that are away from $(0,0)$. Moreover, what we eventually need is derivatives with respect to the physical coordinates, for example,

$$
\frac{\partial\mathbf{B}_0\circ s^{-1}(x,y)}{\partial x}=\frac{\partial\mathbf{B}_0}{\partial u}\frac{\partial u}{\partial x}+\frac{\partial\mathbf{B}_0}{\partial v}\frac{\partial v}{\partial x},
$$

where s^{-1} is the inverse mapping of $s(u,v)$. The troublesome factor 2^n, which may cause overflow when n becomes too large, is canceled out with that from $\partial u/\partial x$ (which is 2^{-n}) and does not cause any numerical issues. The same argument applies to higher order derivatives.

Remark 6. In Reference 40, the eigen structure $(\mathbf{\Lambda},\mathbf{V})$ is precomputed for different valence numbers and stored in a file for repeated use. However, the same scheme cannot be applied to tHNUS because the subdivision matrix $\mathbf{S}_2$ depends on not only the valence number but also the surrounding knot intervals, leading to infinite possible cases of $\mathbf{S}_2$. Therefore, the eigen structure of $\mathbf{S}_2$ needs to be found in real time for every irregular element. Alternatively, we can directly perform matrix multiplications to compute $(\mathbf{S}_2)^{n-2}$, especially when the valence number is small and basis functions need to be computed at points other than $(0,0)$. This is indeed the case in IGA where evaluation is needed at quadrature points. In practice, we adopt a near-machine-precision tolerance (e.g., 10^{-13}) to prevent a potential overflow issue.

Remark 7. In the previous discussion, $\mathbf{B}_0$ is derived under the assumption that there is only one nonquad face next to an irregular element, which, however, is not a necessary condition. When an irregular element has multiple adjacent nonquad faces, we treat it as a macro element and pseudo-subdivide it once. Each of the resulting four subelements only has one nonquad face, where basis functions are defined according to our previous discussion. In other words, basis

functions are well defined on each quarter of the original macro element. This extension follows the same idea proposed in Reference 18, which extends Stam's derivation[40] to arbitrary unstructured quad meshes.

4.2 | Quadrature

To apply the standard Gauss quadrature rule, we need to guarantee that the involved basis functions are polynomials (rather than piecewise polynomials) on each integration cell. However, the functions in $\mathbf{B}_0$ are piecewise smooth polynomials defined on an infinite series of subdomains, that is, $\{\Omega_k^n\}_{n=1}^{\infty}$ $(k = 1, 2, 3)$. The straightforward way is to apply the Gauss quadrature rule on each cell Ω_k^n up to a certain fine level, which was adopted in several subdivision-based isogeometric methods.[18,42] We call such a quadrature *the full quadrature scheme*. In our patch test, we observe that the solution achieves machine precision ($\sim 10^{-16}$) when the 4-point rule is used and the level n is set to be 10. As a result, a total number of 496 quadrature points are needed for a single irregular element. Note that only 16 Gauss quadrature points are used for a regular element.

Alternatively, we can "brutally" apply the Gauss quadrature rule to the entire irregular element without respecting the fact that subdivision functions are piecewise polynomials. In other words, only 16 (rather than 496) Gauss quadrature points are placed on an irregular element. We call it the *reduced quadrature scheme* due to this significant reduction in the number of required quadrature points. In Section 5, we will observe that the reduced quadrature does not influence convergence. In fact, it does not introduce noticeable numerical error in terms of the L^2- or H^1-norm error compared with the full quadrature.

4.3 | Properties

Now we briefly discuss several properties of tHNUS basis functions, including nonnegative partition of unity, refinability (equivalent to nested spline spaces), and global linear independence. The nonnegative partition of unity of tHNUS basis functions follows from the fact that all the entries in the subdivision matrix are nonnegative and each row sum of the subdivision matrix is one. Refinability states that each basis function of a given mesh can be represented as a linear combination of those defined on a refined mesh. In fact, we can see this property in the derivation of $\mathbf{B}_0$, where basis functions are always expressed as linear combinations of functions in the refined meshes.

Finally, the global linear independence implies linear independence on the entire domain, and it can be easily shown under the mild assumption that each irregular element has at least one regular element as its direct neighbor. Under this assumption, every basis function has support on a certain regular element, where it is simply a B-spline. As B-splines are linearly independent on such an element, we can conclude that all the basis functions are linearly independent on the entire domain by going through all the regular elements. The proof on general meshes becomes more involving because we need to resolve different configurations of nonquad faces, or equivalently, configurations of extraordinary vertices in the input mesh. A complex configuration usually occurs when the mesh is very coarse such that many extraordinary vertices may be next to one another. When this is the case, we can perform global refinement to guarantee linear independence.

5 | NUMERICAL EXAMPLES

In this section, we present several numerical examples using tHNUS surfaces in both geometric modeling and IGA.

5.1 | Geometric modeling with tHNUS surfaces

We show some tHNUS limit surface examples and compare them with the existing nonuniform subdivision schemes. We first show the graphs of blending functions for the extraordinary points (EPs) with different valences, such as valence-5 EP in Figure 9, valence-6 EP in Figure 10 and valence-7 EP in Figure 11. As stated in Reference 14, the approaches in References 30,32 and 43 produce limit surfaces with very similar quality in all the examples. Therefore, we only show the limit surface comparisons in one example as shown in Figure 9. All the rest of the examples only show the limit surface of

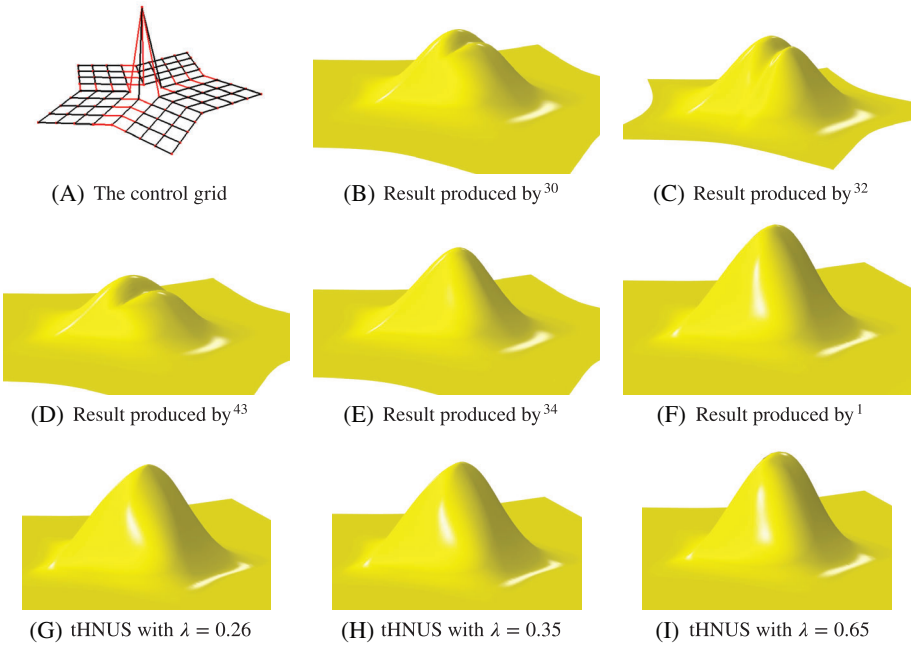


FIGURE 9 The blending function for a valence-5 nonuniform extraordinary point using different approaches, where the knot intervals of the red edges are 10 and those of the other edges are 1

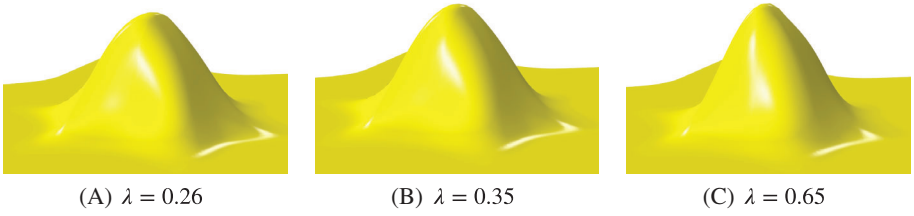


FIGURE 10 The blending function for a valence-6 nonuniform extraordinary point using different λ

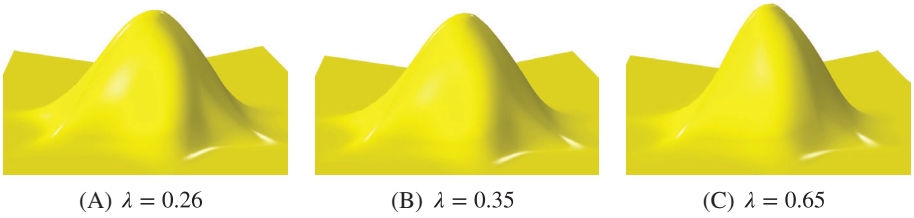


FIGURE 11 The blending function for a valence-7 nonuniform extraordinary point using different λ

the new tHNUS with different λ. According to Lemma 2, λ is allowed to take any values in the range $\left(\frac{1}{4}, 1\right)$. In particular, we choose λ as 0.26, 0.35, and 0.65 to show noticeable differences in the resulting shapes. We can observe that all different λ can produce better shape quality than those approaches in References 30,32 and 43, but the small λ produces worse shape quality surround the EPs, see Figures 12 and 13 for the details.

5.2 | IGA applications using tHNUS basis functions

In this section, we test the performance of tHNUS basis functions in the context of IGA. We solve the Poisson's equation with several unstructured quad meshes as the input. We start with convergence tests on a unit square, whose input control mesh has two EPs, one of valence 3 and the other of valence 5; see Figure 7(A). These tests are aimed at studying: (1) the role of λ in convergence, (2) the feasibility of using reduced quadrature, (3) the influence of nonuniform parameterizations on convergence, and (4) how the matrix conditioning varies with λ.

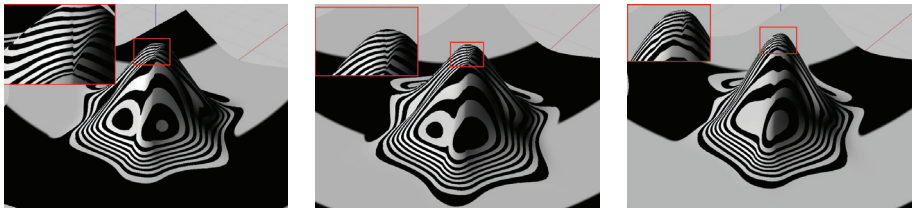


FIGURE 12 The blending function for a valence-6 nonuniform extraordinary point using different λ, where larger λ produces more satisfactory reflection lines

(A) $\lambda = 0.26$ (B) $\lambda = 0.35$ (C) $\lambda = 0.65$

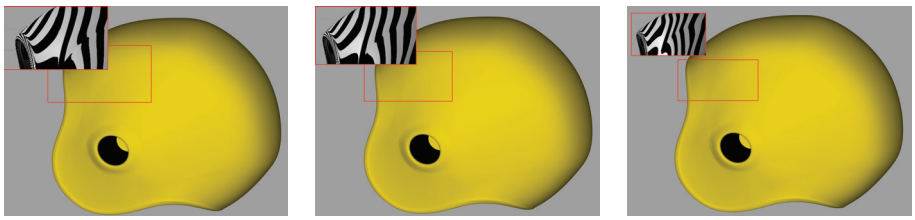


FIGURE 13 A comparison of different λ applied to the helmet model. The artifact for the reflection lines exists for $\lambda = 0.26$

(A) $\lambda = 0.26$ (B) $\lambda = 0.35$ (C) $\lambda = 0.65$

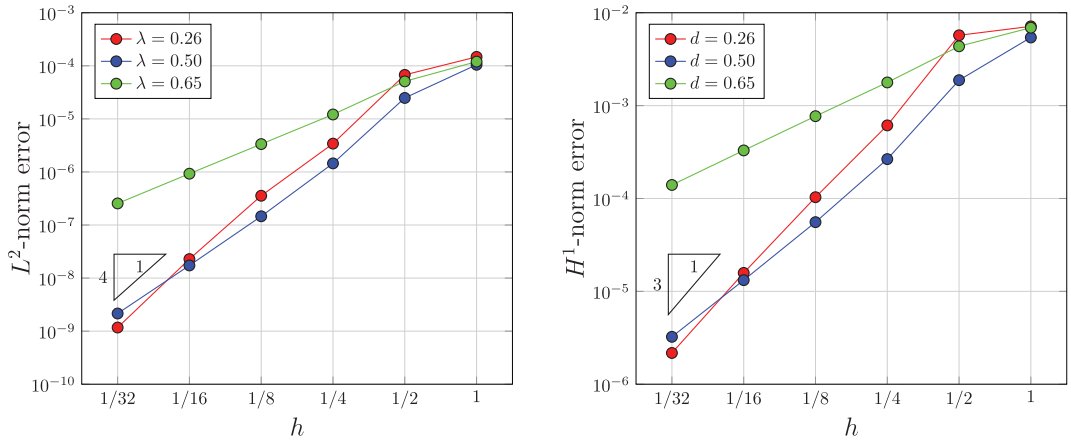


FIGURE 14 Convergence plots using $\lambda \in \{0.65, 0.5, 0.26\}$. Particularly, $\lambda = 0.5$ corresponds to the original hybrid nonuniform subdivision whereas $\lambda = 0.26$ recovers optimal convergence rates

First, we study the influence of the tuning parameter λ on the convergence behavior. We are primarily interested in the case when $\lambda \in \left(\frac{1}{4}, \frac{1}{2}\right]$, so we sample within this range and check the corresponding convergence. We intend to find the largest possible value that recovers optimal convergence, which is called the critical parameter and is denoted as λ_c. We will observe that λ_c is independent of geometries, knot interval configurations, and the solution field to be approximated. To better highlight different convergence behaviors, here we only pick three values for λ: 0.26, 0.5, and 0.65. Recall that tHNUS is equivalent to the original HNUS when $\lambda = 0.5$.

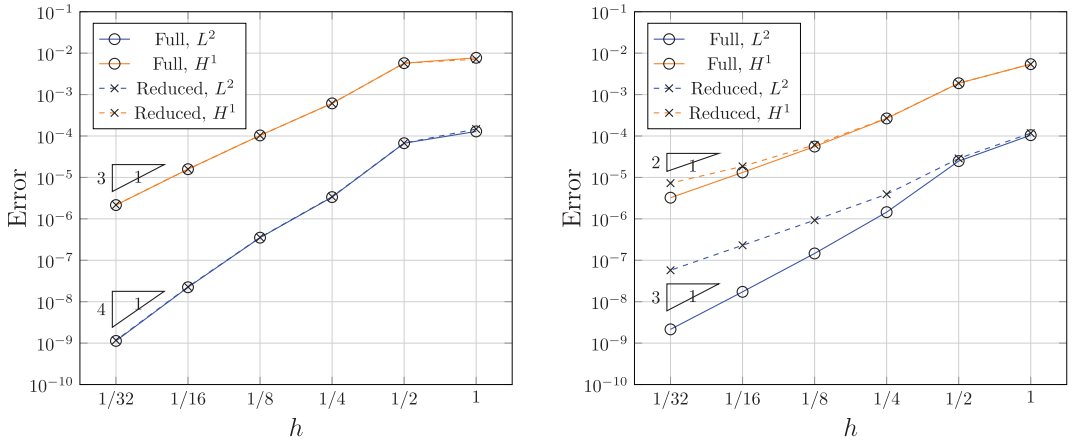
We adopt uniform parameterization (i.e., same knot intervals) around EPs as well as full quadrature in this study. With the manufactured solution $u(x, y) = \sin(\pi x)\sin(\pi y)$, we summarize the convergence plots in Figure 14. We observe that a smaller λ delivers a better convergence behavior, and particularly, optimal convergence rates are achieved when $\lambda = 0.26$ (i.e., $\lambda_c = 0.26$). The tuned Catmull–Clark subdivision (with uniform parameterization) was studied in Reference 20, where optimal convergence rates in the L^2-norm were observed in the Poisson's problem when $\lambda = 0.39$. It indicates that the tuning parameter in tHNUS plays a less sensitive role than that in Reference 20 because tHNUS requires a smaller λ to recover optimal convergence. The reason may be that λ brings more vertices to move further towards each EP than in tHNUS. As a result, the tuned Catmull–Clark subdivision has a faster shrinkage in irregular regions. We will provide insights about why reducing λ recovers optimal convergence later when we study the meshes with high-valence EPs.

Second, we compare the two quadrature schemes for irregular elements as discussed in Section 4.2, full quadrature versus reduced quadrature, under uniform parameterization and with $\lambda = 0.26$ and $\lambda = 0.5$. We observe in Figure 15(A) that there is no noticeable difference in terms of both L^2- and H^1-norm errors. In other words, both quadrature schemes deliver the same level of accuracy when $\lambda = 0.26$. By contrast, quadrature plays an important role when $\lambda = 0.5$, where the full quadrature yields nearly one-order higher convergence rates than the reduced quadrature. This indicates that when $\lambda = 0.26$, the corresponding basis functions (piecewise polynomials) in irregular elements can be better approximated by polynomials than those using $\lambda = 0.5$, and 16 quadrature points seem to suffice to retain accuracy. We further conjecture that when a subdivision basis has the optimal approximation property, quadrature may only play a secondary role, which is opposed to the subdivision schemes that lead to suboptimal convergence. However, further study is needed to fully understand the mechanism behind.

Third, we study several different nonuniform parameterizations for convergence test, which can be obtained by assigning different knot intervals to the edges in the input control mesh. Semiuniform knot intervals are usually adopted in the literature, where all the edges are assigned a unit knot interval except for those perpendicular to the boundary, which are assigned a zero knot interval. To have nonuniform parameterization around EPs, we modify the semiuniform setting in two ways: (1) the knot interval (denoted by d) of highlighted spoke edges takes values $d \in \{1, 2, 5, 10\}$; and (2) every spoke edge is assigned a different knot interval; see Figure 16(A,B). In both cases, we observe in Figure 16(C,D) that tHNUS basis functions can achieve optimal convergence rates with $\lambda = 0.26$. We also observe that the convergence plots corresponding to a larger d slightly shift up, meaning that larger difference in knot intervals yields larger approximation error. In other words, the "distortion" in parameterization influences accuracy rather than convergence.

Fourth, we check how λ influences the conditioning of stiffness matrices. The condition number is defined as the ratio of the maximum eigenvalues versus the minimum, and it is computed on a series of refined meshes with $\lambda = 0.26$, 0.5, and 0.65. The corresponding results are summarized in Figure 17 for $\lambda = 0.26$ and 0.5. We observe that (1) the condition number increases as expected with h^2; and (2) the condition numbers are identical for different λ's. Indeed, the condition number is also the same when $\lambda = 0.65$. The second discovery indicates that changing λ does not affect conditioning.

Now, we consider meshes with high-valence EPs (valence 6, 7, and 8), where each spoke edge is assigned a different knot interval. Two different manufactured solutions are adopted: $u(x, y) = \sin(\pi x)\sin(\pi y)$ and $u(x, y) = \exp((x + y)/2)$. The Dirichlet boundary condition is strongly imposed on the entire boundary, where a least-squares fitting is performed to project the Dirichlet data on the spline space. We again observe optimal convergence rates with $\lambda = 0.26$ in all the three meshes; see Figure 18. Moreover, let us have a close look at how λ influences parameterization around EPs. In particular, we compare isoparametric lines using two different λ's (0.5 versus 0.26) around a valence-6 EP, where the same input control mesh in Figure 18(A) is used in both cases. We find in Figure 19 that isoparametric lines are



(A) $\lambda = 0.26$ (B) $\lambda = 0.5$

FIGURE 15 Convergence plots using full and reduced quadrature

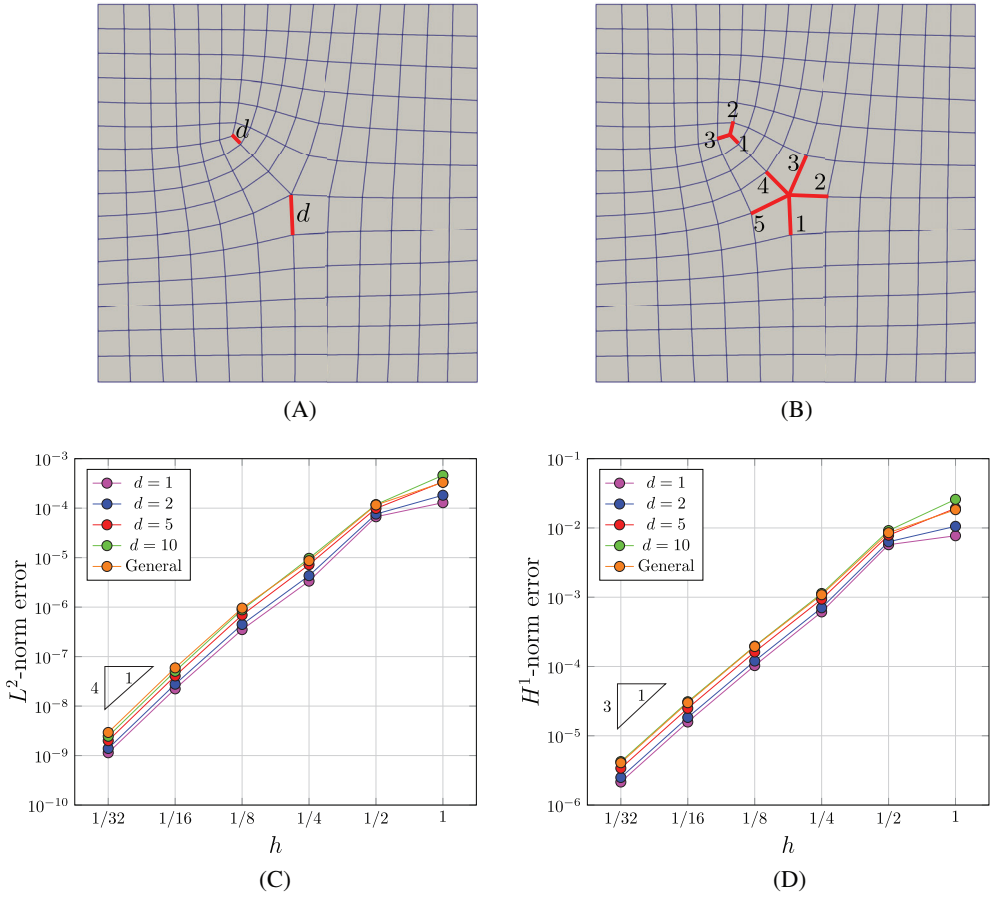


FIGURE 16 Convergence plots under different nonuniform parameterizations. (A, B) The configurations of knot intervals around extraordinary points, and (C, D) convergence plots in L^2- and H^1-norm errors

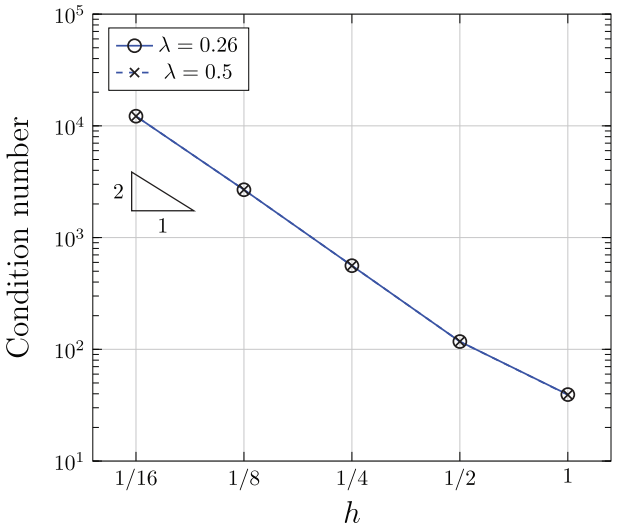


FIGURE 17 Condition numbers with $\lambda = 0.26$ and 0.5

overlaid with one another in most regions, and with a smaller λ, the isoparametric lines (blue curves) are more bent towards the extraordinary surface point $s(0, 0)$. Equivalently speaking, a smaller λ yields smaller refined irregular elements in the physical domain. Therefore, the mesh around $s(0, 0)$ becomes denser than that using a larger λ, and as a result, the asymptotic approximation error controlled by $s(0, 0)$ can be reduced using such a denser mesh. Ideally, optimal convergence rates can be achieved by reducing λ, which indeed is the case in all our numerical tests when $\lambda = 0.26$.

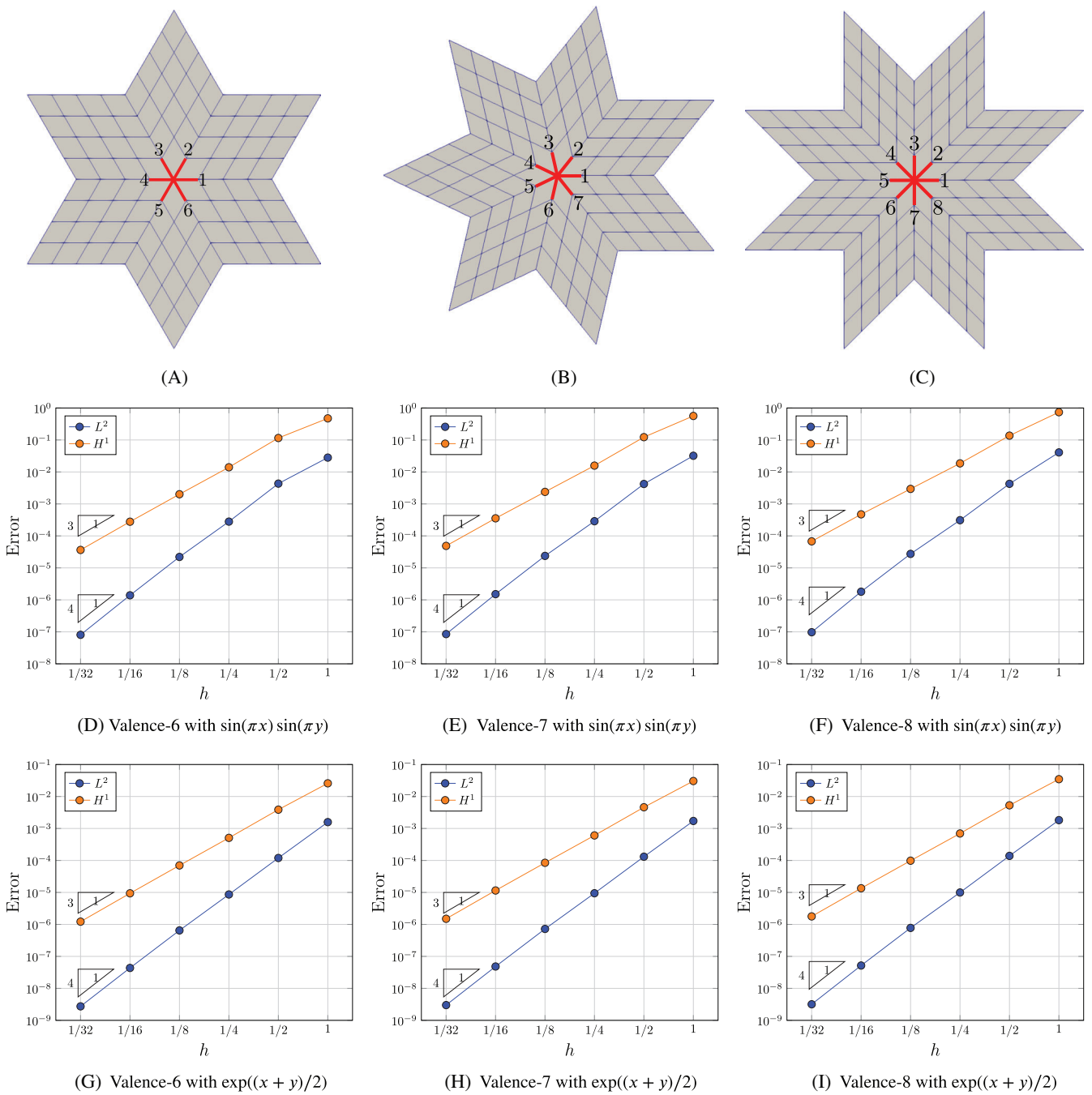


FIGURE 18 Convergence plots using meshes with high-valence EPs. (A–C) The configurations of knot intervals around EPs, (D–F) convergence plots with the solution $u(x,y) = \sin(\pi x)\sin(\pi y)$, and (G–I) convergence plots with the solution $u(x,y) = \exp((x+y)/2)$. EPs, extraordinary points

Remark 8. Although the globally smooth tHNUS basis functions can be applied to solve 4th-order partial differential equations (PDEs), our preliminary tests only show suboptimal convergence in solving the biharmonic equation, where obtained convergence rates in terms of L^2-, H^1- and H^2-norm errors are around 2, 2, and 1, respectively. This is consistent with the result reported in Reference 38, where a thin-shell problem was solved and the reported convergence rates in L^2- and energy norm errors are 2 and 1, respectively. In other words, reducing λ alone is not sufficient for high-order PDEs. We conjecture that to recover the optimal convergence in this case, we may need more degrees of freedom around EPs following similar ideas in References 24,27. However,

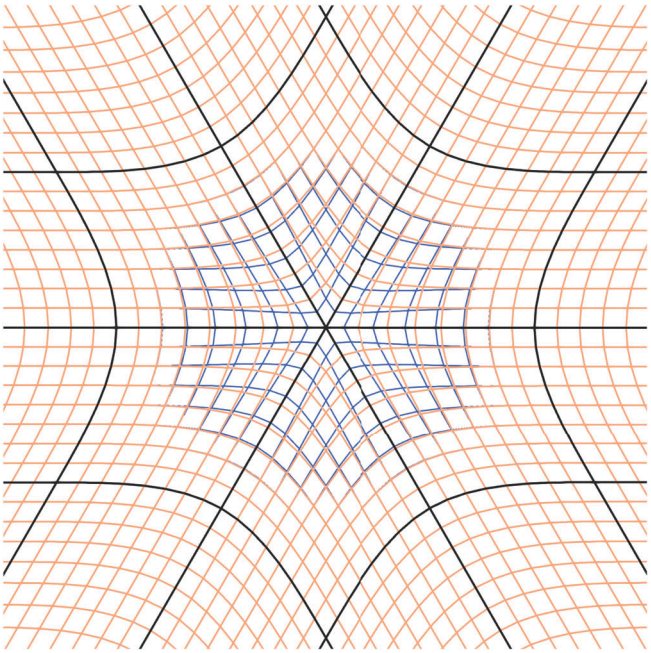


FIGURE 19 Isoparametric lines around the valence-6 extraordinary point under uniform parameterization. Orange and blue curves are isoparametric lines corresponding to $\lambda = 0.5$ and $\lambda = 0.26$, respectively. Blue curves are not visible in most regions because they are overlaid with orange ones. Black curves indicate element boundaries in the physical domain

this would further complicate the current subdivision framework, so we postpone related results in a follow-up work.

6 | CONCLUSIONS AND FUTURE WORK

We have presented a tuned version of HNUS, tHNUS, by introducing a parameter $\lambda \in \left(\frac{1}{4}, 1\right)$, which is also the second and third eigenvalues of the subdivision matrix. The tHNUS surface is proved to be G^1-continuous for any positive knot intervals and extraordinary vertices of any valence. The tHNUS surface has satisfactory shape quality for any λ under nonuniform parameterization. However, the highest shape quality is achieved when $\lambda = 0.5$. In other words, the original HNUS generally performs better in geometric modeling than tHNUS. On the other hand, tHNUS basis functions can achieve optimal convergence rates when λ is reduced to 0.26, regardless of which quadrature scheme is used and whether parameterizations are uniform or nonuniform around EPs.

In the future, we can extend tHNUS in the following three issues. First, converting an input quad mesh to its hybrid counterpart can be restricted locally to irregular regions without introducing zero-measure faces throughout the entire mesh. This can be performed by allowing T-junctions[4] in the hybrid mesh, but support of T-junctions in a hybrid mesh requires a much more sophisticated data structure to accommodate both nonquad faces and quads with T-junctions. Second, tHNUS can be adapted to hierarchical splines[3] due to its refinability property, where the initial level corresponds to the initial hybrid mesh. The construction of hierarchical tHNUS essentially follows those proposed for truncated hierarchical Catmull–Clark subdivision surfaces,[17,18] but the differences lie in dealing with hybrid meshes and nonuniform knot intervals. Third, improving tHNUS to achieve optimal convergence rates in solving high-order PDEs is another challenging but very interesting direction to pursue. Currently, we can only show optimal convergence in solving the second-order PDE. In the case of high-order PDEs, our preliminary tests suggest that it is not sufficient to tune λ alone and additional treatment is needed. We plan to our investigation by adding more control points around extraordinary vertices.

ACKNOWLEDGEMENTS

X.L. was supported by the National Key R&D Program of China(2020YFB1708900), NSF of China (No.61872328) and the Youth Innovation Promotion Association CAS. X.W. was supported in part by ERC AdG project CHANGE n. 694515 and the Swiss NSF project HOGAEMS n.200021_188589. Y.J.Z. was supported in part by the NSF grants CMMI-1953323 and CBET-1804929. T.J.R.H. was partially supported by the Office of Naval Research, USA (Grant Nos. N00014-17-1-2119 and N00014-13-1-0500).

ORCID

Xin Li https://orcid.org/0000-0003-0477-7098
Yongjie J. Zhang https://orcid.org/0000-0001-7436-9757

REFERENCES

1. Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Comput Methods Appl Mech Eng*. 2005;194:4135-4195.
2. Cottrell JA, Hughes TJR, Bazilevs Y. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Hoboken, NJ: Wiley; 2009.
3. Vuong AV, Giannelli C, Jüttler B, Simeon B. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comput Methods Appl Mech Eng*. 2011;200:3554-3567.
4. Sederberg TW, Cardon DL, Finnigan GT, North NS, Zheng J, Lyche T. T-spline simplification and local refinement. *ACM Trans Graph*. 2004;23(3):276-283.
5. Li X, Zheng J, Sederberg TW, Hughes TJR, Scott MA. On the linear independence of T-splines blending functions. *Comput Aided Geom Des*. 2012;29:63-76.
6. Scott MA, Li X, Sederberg TW, Hughes TJR. Local refinement of analysis-suitable T-splines. *Comput Methods Appl Mech Eng*. 2012;213-216:206-222.
7. Veiga LB, Buffa A, Cho D, Sangalli G. Analysis-suitable T-splines are dual-compatible. *Comput Methods Appl Mech Eng*. 2012;249-252:42-51.
8. Zhang J, Li X. On the linear independence and partition of unity of arbitrary degree analysis-suitable T-splines. *Commun Math Stat*. 2015;3(3):353-364.
9. Li X, Zhang J. AS++ T-splines: linear independence and approximation. *Comput Methods Appl Mech Eng*. 2018;333:462-474.
10. Wei X, Zhang Y, Liu L, Hughes TJR. Truncated T-splines: fundamentals and methods. *Comput Methods Appl Mech Eng*. 2017;316:349-372.
11. Zhang J, Li X. Local refinement of analysis-suitable++ T-splines. *Comput Methods Appl Mech Eng*. 2018;342:32-45.
12. Deng J, Chen F, Li X, et al. Polynomial splines over hierarchical T-meshes. *Graph Model*. 2008;74:76-86.
13. Dokken T, Lyche T, Pettersen KF. Polynomial splines over locally refined box-partitions. *Comput Aided Geom Des*. 2013;30:331-356.
14. Li X, Wei X, Zhang Y. Hybrid non-uniform recursive subdivision with improved convergence rates. *Comput Methods Appl Mech Eng*. 2019;352:606-624.
15. Cirak F, Ortiz M, Schröder P. Subdivision surfaces: a new paradigm for thin shell analysis. *Int J Numer Methods Eng*. 2000;47:2039-2072.
16. Burckhart D, Umlauf BHG. Iso-geometric finite element analysis based on Catmull-Clark subdivision solids. *Comput Graph Forum*. 2010;29(5):1575-1584.
17. Wei X, Zhang Y, Hughes TJR, Scott MA. Truncated hierarchical Catmull-Clark subdivision with local refinement. *Comput Methods Appl Mech Eng*. 2015;291:1-20.
18. Wei X, Zhang Y, Hughes TJR, Scott MA. Extended truncated hierarchical Catmull-Clark subdivision. *Comput Methods Appl Mech Eng*. 2016;299:316-336.
19. Riffnaller-Schiefer A, Augsdorfer U, Fellner D. Isogeometric shell analysis with NURBS compatible subdivision surfaces. *Appl Math Comput*. 2016;272:139-147.
20. Ma Y, Ma W. A subdivision scheme for unstructured quadrilateral meshes with improved convergence rate for isogeometric analysis. *Graph Model*. 2019;106:101043.
21. Collin A, Sangalli G, Takacs T. Analysis-suitable G^1 multi-patch parametrizations for C^1 isogeometric spaces. *Comput Aided Geom Des*. 2016;47:93-113.
22. Kapl M, Buchegger F, Bercovier M, Jüttler B. Isogeometric analysis with geometrically continuous functions on planar multi-patch geometries. *Comput Methods Appl Mech Eng*. 2017;316:209-234.
23. Nguyen T, Peters J. Refinable C^1 spline elements for irregular quad layout. *Comput Aided Geom Des*. 2016;43:123-130.
24. Toshniwal D, Speleers H, Hughes T. Smooth cubic spline spaces on unstructured quadrilateral meshes with particular emphasis on extraordinary points: geometric design and isogeometric analysis considerations. *Comput Methods Appl Mech Eng*. 2017;327:411-458.
25. Casquero H, Wei X, Toshniwal D, et al. Seamless integration of design and Kirchhoff-Love shell analysis using analysis-suitable unstructured T-splines. *Comput Methods Appl Mech Eng*. 2020;360:112765.
26. Majeed M, Cirak F. Isogeometric analysis using manifold-based smooth basis functions. *Comput Methods Appl Mech Eng*. 2017;326:547-567.
27. Wei X, Zhang YJ, Toshniwal D, et al. Blended B-spline construction on unstructured quadrilateral and hexahedral meshes with optimal convergence rates in isogeometric analysis. *Comput Methods Appl Mech Eng*. 2018;341:608-639.
28. Jüttler B, Mantzaflaris A, Perl R, Rumpf M. On numerical integration in isogeometric subdivision methods for PDEs on surfaces. *Comput Methods Appl Mech Eng*. 2016;302:131-146.
29. Barendrecht PJ, Bartoň M, Kosinka J. Efficient quadrature rules for subdivision surfaces in isogeometric analysis. *Comput Methods Appl Mech Eng*. 2018;340:1-23.
30. Sederberg TW, Zheng J, Sewell D, Sabin M. Non-uniform recursive subdivision surfaces. Paper presented at: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques; 1998:387-394; ACM Press/Addison-Wesley Publishing Co, New York, NY.

31. Müller K, Reusche L, Fellner D. Extended subdivision surfaces: building a bridge between NURBS and Catmull-Clark surfaces. *ACM Trans Graph*. 2006;25:268-292.
32. Cashman TJ, Augsdörfer UH, Dodgson NA, Sabin MA. NURBS with extraordinary points: high-degree, non-uniform, rational subdivision schemes. *ACM Trans Graph*. 2009;28(3):1-9.
33. Müller K, Funfzig C, Reusche L, Hansford D, Farin G, Hagen H. DINUS-double insertion non-uniform stationary subdivision surfaces. *ACM Trans Graph*. 2010;29:1-21.
34. Li X, Finnigan GT, Sederberg TW. G^1 non-uniform catmull-clark surfaces. *Trans Graph*. 2016;35(4).
35. Halstead M, Kass M, DeRose T. Efficient,. Fair interpolation using Catmull-Clark surfaces. *SIGGRAPH*. 20th Annual Conference and Exhibition on Computer Graphics and Interactive Techniques. Anaheim CA: Association for Computing Machinery; 1993:35-44.
36. Kobbelt L. A variational approach to subdivision. *Comput Aided Geom Des*. 1996;13(8):743-761.
37. Augsdorfer UH, Dodgson NA, Sabin MA. Tuning subdivision by minimising gaussian curvature variation near extraordinary vertices. *Comput Graph Forum*. 2006;25(3):263-272.
38. Zhang Q, Sabin M, Cirak F. Subdivision surfaces with isogeometric analysis adapted refinement weights. *Comput Aided Des*. 2018;102:104-114.
39. Reif U. A unified approach to subdivision algorithms near extraordinary vertices. *Comput Aided Geom Des*. 1995;12:153-174.
40. Stam J. Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values. Paper presented at: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques; 1998:395-404.
41. Boier-Martin I, Zorin D. Differentiable parameterization of Catmull-Clark subdivision surfaces. Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing; 2004:155-164.
42. Nguyen T, Karčiauskas K, Peters J. A comparative study of several classical, discrete differential and isogeometric methods for solving Poisson's equation on the disk. *Axioms*. 2014;3:280-300.
43. Kovacs D, Bisceglio J, Zorin D. Dyadic T-mesh subdivision. *ACM Trans Graph*. 2015;34(4):143.

How to cite this article: Wei X, Li X, Zhang YJ, Hughes TJR. Tuned hybrid nonuniform subdivision surfaces with optimal convergence rates. *Int J Numer Methods Eng*. 2021;122:2117–2144. https://doi.org/10.1002/nme.6608

APPENDIX 1

The control points $P_i^{j,k}$ $(1 \le j,k \le 3)$ of the characteristic map are listed as follows.

$$P_i^{1,1} = -\frac{(\lambda-2)((32\lambda^3-100\lambda^2+31\lambda-2)p-12\lambda^2(v+w))}{(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)},$$

$$P_i^{1,2} = \frac{\begin{array}{l}6\lambda^2((256\lambda^3+136\lambda^2-2251\lambda-232)v+2(128\lambda^3-184\lambda^2-149\lambda+10)w)\\+(-4096\lambda^6+18688\lambda^5-14576\lambda^4-22112\lambda^3+8275\lambda^2-608\lambda+20)p\end{array}}{(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)},$$

$$P_i^{1,3} = \frac{\begin{array}{l}6\lambda^2(2(320\lambda^3-919\lambda^2-800\lambda+7)v+(160\lambda^3-314\lambda^2-13\lambda+2)w)\\+(-2560\lambda^6+15904\lambda^5-29348\lambda^4+3535\lambda^3+928\lambda^2-62\lambda+2)p\end{array}}{(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)},$$

$$P_i^{2,1} = \frac{\begin{array}{l}6\lambda^2((256\lambda^3-368\lambda^2-298\lambda+20)v+(256\lambda^3+136\lambda^2-2251\lambda-232)w)+\\(-4096\lambda^6+18688\lambda^5-14576\lambda^4-22112\lambda^3+8275\lambda^2-608\lambda+20)p\end{array}}{(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)},$$

$$P_i^{2,2} = \frac{\begin{array}{l}6(4096\lambda^4+33024\lambda^3-96320\lambda^2-11271\lambda-1160)\lambda^2(v+w)+\\(-65536\lambda^7-194560\lambda^6+2362752\lambda^5-4183824\lambda^4+1165584\lambda^3-71505\lambda^2+976\lambda+100)p\end{array}}{(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)},$$

$$P_i^{2,3} = \frac{6\lambda^2((74240\lambda^4-139104\lambda^3-145684\lambda^2-8118\lambda-7)v+(58880\lambda^4-106224\lambda^3-32914\lambda^2-2208\lambda-127)w)}{2(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$

$$+\frac{(-942080\lambda^7+4816384\lambda^6-6465888\lambda^5+394032\lambda^4+345858\lambda^3-33192\lambda^2+1313\lambda-1)p}{2(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)},$$

$$P_i^{3,1}=\frac{\begin{array}{l}6\lambda^2((160\lambda^3-314\lambda^2-13\lambda+2)v+2(320\lambda^3-919\lambda^2-800\lambda+7)w)\\+(-2560\lambda^6+15904\lambda^5-29348\lambda^4+3535\lambda^3+928\lambda^2-62\lambda+2)p\end{array}}{(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)},$$

$$P_i^{3,2}=\frac{6\lambda^2((58880\lambda^4-106224\lambda^3-32914\lambda^2-2208\lambda-127)v+(74240\lambda^4-139104\lambda^3-145684\lambda^2-8118\lambda-7)w)}{2(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$+\frac{(-942080\lambda^7+4816384\lambda^6-6465888\lambda^5+394032\lambda^4+345858\lambda^3-33192\lambda^2+1313\lambda-1)p}{2(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)},$$

$$P_i^{3,3}=\frac{\begin{array}{l}6(25600\lambda^4-30144\lambda^3-61166\lambda^2-5685\lambda-236)\lambda^2(v+w)\\+(-409600\lambda^7+1923584\lambda^6-1925280\lambda^5-1010388\lambda^4+470631\lambda^3-36036\lambda^2+1060\lambda+16)p\end{array}}{(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}.$$

The expressions of all $S_2^{j,k}$ $(0\le j,k\le 3)$ are listed as follows.

$$S_2^{0,0}=\frac{3\lambda(28672\lambda^4-44160\lambda^3-104888\lambda^2-4242\lambda-59)v}{2(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$+\frac{3\lambda(-20480\lambda^4+125568\lambda^3+2296\lambda^2-2886\lambda-107)w}{2(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$+\frac{(-589824\lambda^7+2557952\lambda^6-1417856\lambda^5-723088\lambda^4+318940\lambda^3-25264\lambda^2+853\lambda+3)p}{4(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$

$$S_2^{1,0}=-\frac{6\lambda(14336\lambda^4-53888\lambda^3+58744\lambda^2+2009\lambda+30)v}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$-\frac{12\lambda(5120\lambda^4-11808\lambda^3+1904\lambda^2-234\lambda+41)w}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$+\frac{2(81920\lambda^7-467968\lambda^6+851904\lambda^5-540792\lambda^4+116238\lambda^3-7683\lambda^2+211\lambda+1)p}{(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$

$$S_2^{2,0}=-\frac{(163840\lambda^5-360448\lambda^4+20512\lambda^3+2732\lambda^2+571\lambda+47)w}{4(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$-\frac{(458752\lambda^6-1579008\lambda^5+1185408\lambda^4+775632\lambda^3-25056\lambda^2+567\lambda-4)v}{8(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$+\frac{(2621440\lambda^8-14696448\lambda^7+25499648\lambda^6-13590144\lambda^5+1352256\lambda^4+248436\lambda^3-23683\lambda^2+897\lambda-5)p}{24(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)},$$

$$S_2^{0,1}=\frac{18\lambda(2048\lambda^4-1856\lambda^3-7028\lambda^2-669\lambda-13)v}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$+\frac{18\lambda(-2048\lambda^4+12288\lambda^3+1904\lambda^2-556\lambda-17)w}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$-\frac{3(32768\lambda^7-129024\lambda^6+47872\lambda^5+108496\lambda^4-38716\lambda^3+3018\lambda^2-95\lambda-1)p}{(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$

$$S_2^{1,1}=-\frac{6\lambda(12288\lambda^4-38912\lambda^3+43904\lambda^2+4127\lambda+76)v}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$-\frac{12\lambda(6144\lambda^4-14080\lambda^3+1036\lambda^2-299\lambda+80)w}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$+\frac{2(98304\lambda^7-553984\lambda^6+985152\lambda^5-637296\lambda^4+138348\lambda^3-9009\lambda^2+216\lambda+4)p}{(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)},$$

$$S_2^{2,1} = -\frac{3(32768\lambda^5 - 93184\lambda^4 + 70336\lambda^3 + 61032\lambda^2 - 210\lambda + 7)v}{2(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} - \frac{3(32768\lambda^5 - 70656\lambda^4 - 4768\lambda^3 + 888\lambda^2 + 236\lambda + 15)w}{2(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} + \frac{(524288\lambda^8 - 2883584\lambda^7 + 4827136\lambda^6 - 2557248\lambda^5 + 206544\lambda^4 + 64080\lambda^3 - 6056\lambda^2 + 233\lambda - 1)p}{4(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$

$$S_2^{0,2} = \frac{6\lambda(4096\lambda^4 - 1024\lambda^3 - 19348\lambda^2 - 1877\lambda - 117)v}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} + \frac{6\lambda(-4096\lambda^4 + 21888\lambda^3 + 20552\lambda^2 - 3786\lambda - 97)w}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} + \frac{(-65536\lambda^7 + 215040\lambda^6 + 122880\lambda^5 - 518496\lambda^4 + 160368\lambda^3 - 11706\lambda^2 + 295\lambda + 9)p}{(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$

$$S_2^{1,2} = -\frac{6\lambda(8192\lambda^4 - 28928\lambda^3 + 34888\lambda^2 + 4205\lambda + 228)v}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} - \frac{48\lambda(1024\lambda^4 - 2272\lambda^3 - 868\lambda^2 - 65\lambda + 39)w}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} + \frac{2(65536\lambda^7 - 393216\lambda^6 + 768768\lambda^5 - 550584\lambda^4 + 120330\lambda^3 - 6531\lambda^2 + 32\lambda + 12)p}{(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$

$$S_2^{2,2} = -\frac{(32768\lambda^5 - 100352\lambda^4 + 78176\lambda^3 + 82984\lambda^2 + 1862\lambda + 7)v}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} - \frac{(32768\lambda^5 - 63488\lambda^4 - 49120\lambda^3 + 2596\lambda^2 + 845\lambda + 43)w}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} + \frac{(524288\lambda^8 - 2998272\lambda^7 + 5300224\lambda^6 - 2834112\lambda^5 - 22440\lambda^4 + 158118\lambda^3 - 13784\lambda^2 + 513\lambda - 1)p}{6(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$

$$S_2^{0,3} = \frac{(393216\lambda^7 + 233472\lambda^6 - 2491392\lambda^5 - 285864\lambda^4 - 21918\lambda^3 - 258\lambda^2)v}{24(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} + \frac{(-393216\lambda^7 + 1769472\lambda^6 + 3983616\lambda^5 - 415440\lambda^4 - 81576\lambda^3 + 1449\lambda^2 - 12\lambda)w}{24(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} + \frac{(-1048576\lambda^8 + 2555904\lambda^7 + 6412288\lambda^6 - 14197248\lambda^5 + 3735840\lambda^4 - 144900\lambda^3 - 5573\lambda^2 + 717\lambda - 1)p}{24(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$

$$S_2^{1,3} = -\frac{(65536\lambda^5 - 262144\lambda^4 + 334720\lambda^3 + 52988\lambda^2 + 3718\lambda + 41)v}{2(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} - \frac{(65536\lambda^5 - 139264\lambda^4 - 137984\lambda^3 - 23368\lambda^2 + 8098\lambda - 7)w}{2(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} + \frac{(1048576\lambda^8 - 6782976\lambda^7 + 14569472\lambda^6 - 11210496\lambda^5 + 2223408\lambda^4 - 32748\lambda^3 - 8632\lambda^2 + 615\lambda + 1)p}{12(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$

$$S_2^{2,3} = -\frac{(65536\lambda^5 - 219136\lambda^4 + 173056\lambda^3 + 229220\lambda^2 + 9745\lambda + 131)v}{3(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} - \frac{(65536\lambda^5 - 108544\lambda^4 - 208832\lambda^3 - 11944\lambda^2 + 6694\lambda + 239)w}{3(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} + \frac{(1048576\lambda^8 - 6291456\lambda^7 + 11761664\lambda^6 - 6099072\lambda^5 - 1021344\lambda^4 + 664680\lambda^3 - 54730\lambda^2 + 1881\lambda + 7)p}{18(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}.$$

The expressions of all $T_2^{j,k}$ $(0 \le j,k \le 3)$ are listed as follows.

$$T_2^{0,0} = -\frac{3\lambda(20480\lambda^4 - 125568\lambda^3 - 2296\lambda^2 + 2886\lambda + 107)v}{2(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$

$$
\begin{aligned}
&- \frac{3\lambda(-28672\lambda^4+44160\lambda^3+104888\lambda^2+4242\lambda+59)w}{2(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&+ \frac{(-589824\lambda^7+2557952\lambda^6-1417856\lambda^5-723088\lambda^4+318940\lambda^3-25264\lambda^2+853\lambda+3)p}{4(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)},
\end{aligned}
$$

$$
\begin{aligned}
T_2^{1,0} = &- \frac{18\lambda(2048\lambda^4-12288\lambda^3-1904\lambda^2+556\lambda+17)v}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&- \frac{18\lambda(-2048\lambda^4+1856\lambda^3+7028\lambda^2+669\lambda+13)w}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&- \frac{3(32768\lambda^7-129024\lambda^6+47872\lambda^5+108496\lambda^4-38716\lambda^3+3018\lambda^2-95\lambda-1)p}{(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}
\end{aligned}
$$

$$
\begin{aligned}
T_2^{2,0} = &- \frac{6\lambda(4096\lambda^4-21888\lambda^3-20552\lambda^2+3786\lambda+97)v}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&- \frac{6\lambda(-4096\lambda^4+1024\lambda^3+19348\lambda^2+1877\lambda+117)w}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&+ \frac{(-65536\lambda^7+215040\lambda^6+122880\lambda^5-518496\lambda^4+160368\lambda^3-11706\lambda^2+295\lambda+9)p}{(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)},
\end{aligned}
$$

$$
\begin{aligned}
T_2^{3,0} = &- \frac{(-65536\lambda^5-38912\lambda^4+415232\lambda^3+47644\lambda^2+3653\lambda+43)w}{4(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&- \frac{(131072\lambda^6-589824\lambda^5-1327872\lambda^4+138480\lambda^3+27192\lambda^2-483\lambda+4)v}{8(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&+ \frac{(-1048576\lambda^8+2555904\lambda^7+6412288\lambda^6-14197248\lambda^5+3735840\lambda^4-144900\lambda^3-5573\lambda^2+717\lambda-1)p}{24(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)},
\end{aligned}
$$

$$
\begin{aligned}
T_2^{0,1} = &- \frac{12\lambda(5120\lambda^4-11808\lambda^3+1904\lambda^2-234\lambda+41)v}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&- \frac{6\lambda(14336\lambda^4-53888\lambda^3+58744\lambda^2+2009\lambda+30)w}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&+ \frac{2(81920\lambda^7-467968\lambda^6+851904\lambda^5-540792\lambda^4+116238\lambda^3-7683\lambda^2+211\lambda+1)p}{(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)},
\end{aligned}
$$

$$
\begin{aligned}
T_2^{1,1} = &- \frac{12\lambda(6144\lambda^4-14080\lambda^3+1036\lambda^2-299\lambda+80)v}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&- \frac{6\lambda(12288\lambda^4-38912\lambda^3+43904\lambda^2+4127\lambda+76)w}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&+ \frac{2(98304\lambda^7-553984\lambda^6+985152\lambda^5-637296\lambda^4+138348\lambda^3-9009\lambda^2+216\lambda+4)p}{(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)},
\end{aligned}
$$

$$
\begin{aligned}
T_2^{2,1} = &- \frac{48\lambda(1024\lambda^4-2272\lambda^3-868\lambda^2-65\lambda+39)v}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&- \frac{6\lambda(8192\lambda^4-28928\lambda^3+34888\lambda^2+4205\lambda+228)w}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&+ \frac{2(65536\lambda^7-393216\lambda^6+768768\lambda^5-550584\lambda^4+120330\lambda^3-6531\lambda^2+32\lambda+12)p}{(\lambda-4)\lambda(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)},
\end{aligned}
$$

$$
\begin{aligned}
T_2^{3,1} = &- \frac{(65536\lambda^5-139264\lambda^4-137984\lambda^3-23368\lambda^2+8098\lambda-7)v}{2(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&- \frac{(65536\lambda^5-262144\lambda^4+334720\lambda^3+52988\lambda^2+3718\lambda+41)w}{2(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&+ \frac{(1048576\lambda^8-6782976\lambda^7+14569472\lambda^6-11210496\lambda^5+2223408\lambda^4-32748\lambda^3-8632\lambda^2+615\lambda+1)p}{12(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)},
\end{aligned}
$$

$$
\begin{aligned}
T_2^{0,2} = &\frac{(-983040\lambda^7+2162688\lambda^6-123072\lambda^5-16392\lambda^4-3426\lambda^3-282\lambda^2)v}{24(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)} \\
&+ \frac{(-1376256\lambda^7+4737024\lambda^6-3556224\lambda^5-2326896\lambda^4+75168\lambda^3-1701\lambda^2+12\lambda)w}{24(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}
\end{aligned}
$$

$$+\frac{(2621440\lambda^8-14696448\lambda^7+25499648\lambda^6-13590144\lambda^5+1352256\lambda^4+248436\lambda^3-23683\lambda^2+897\lambda-5)p}{24(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)},$$

$$T_2^{1,2}=-\frac{3(32768\lambda^5-70656\lambda^4-4768\lambda^3+888\lambda^2+236\lambda+15)v}{2(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$-\frac{3(32768\lambda^5-93184\lambda^4+70336\lambda^3+61032\lambda^2-210\lambda+7)w}{2(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$+\frac{(524288\lambda^8-2883584\lambda^7+4827136\lambda^6-2557248\lambda^5+206544\lambda^4+64080\lambda^3-6056\lambda^2+233\lambda-1)p}{4(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)},$$

$$T_2^{2,2}=-\frac{(32768\lambda^5-63488\lambda^4-49120\lambda^3+2596\lambda^2+845\lambda+43)v}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$-\frac{(32768\lambda^5-100352\lambda^4+78176\lambda^3+82984\lambda^2+1862\lambda+7)w}{(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$+\frac{(524288\lambda^8-2998272\lambda^7+5300224\lambda^6-2834112\lambda^5-22440\lambda^4+158118\lambda^3-13784\lambda^2+513\lambda-1)p}{6(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)},$$

$$T_2^{3,2}=-\frac{(65536\lambda^5-108544\lambda^4-208832\lambda^3-11944\lambda^2+6694\lambda+239)v}{3(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$-\frac{(65536\lambda^5-219136\lambda^4+173056\lambda^3+229220\lambda^2+9745\lambda+131)w}{3(\lambda-4)(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}$$
$$+\frac{(1048576\lambda^8-6291456\lambda^7+11761664\lambda^6-6099072\lambda^5-1021344\lambda^4+664680\lambda^3-54730\lambda^2+1881\lambda+7)p}{18(\lambda-4)\lambda^2(4\lambda-1)(8\lambda-1)(16\lambda-1)(32\lambda-1)(64\lambda-1)}.$$

The following mathematica code has been used to compute the above control points.

```
clear;
a1 = 2*(1 - la)/la;
a2 = 6*(2 - la)*(1 - la)/(8*la - 1)/la;
a3 = 2*(2 - la)*(1 - la)*(1 + la)/(8*la - 1)/la/la;
b2 = 6*(1 - 2*la)/(16*la - 1);
b3 = (1 - 2*la)*(1 + 2*la)/la/(16*la - 1);
p1 = p + a1*v; p2 = (1 + b2)*p + (a1 + a2 - b2) * v;
p3 = (1 + b2 + b3)*p + (a1 + a2 + a3 - b2 - b3) * v;
p4  = p + a1*w; p5 = (1 + b2)*p + (a1 + a2 - b2) * w;
p6 = (1 + b2 + b3)*p + (a1 + a2 + a3 - b2 - b3) * w;
T = Solve[a*la == a*la*la/4 + p1 * la*(2 - la)/4 + p4 * la*(2 - la)/4 +
      p * (2 - la)*(2 - la)/4 && (b - a)*la == ( a + p1*3)/8 + (p1*3 + p2*3 + a + b)/32 -
      a*la*3/4 && (d - a)*la == ( a + p4*3)/8 + (p4*3 + p5*3 + a + d)/32 -
      a*la*3/4 && (c - b)*la == (p1*3 + p2*3 + a + b)*3/32 - a*la/4 - ( a + p1*3)/8 &&
      (g - d)*la == (p4*3 + p5*3 + a + d)*3/32 - a*la/4 - ( a + p4*3)/8 &&
      (e - a)*la == (p1*30 + p2 * 6 + p4 * 30 + a * 120 + b * 22 + p5 * 6 +
      d * 22 + e * 4)/256 - (a*la*la + p1 * la*(2 - la) + p4 * la*(2 - la) +
      p * (2 - la)*(2 - la))*15/64 && (f - c)*la == (a + b)/4 + (a + b + d + e)/16 -
      (p1 * 3 + p2 * 3 + a + b)*3/32 && (h - g)*la == (a + d)/4 + (a + b + d + e)/16 -
      (p4 * 3 + p5 * 3 + a + d)*3/32 && (k - f)*la == (a + b + d + e)*3/16 - (a + b)/4 -
      (p1 * 3 + p2 * 3 + a + b)/32, {a, b, c, d, e, f, g, h, k}];
a = T[[1, 1, 2]]; b = T[[1, 2, 2]]; c = T[[1, 3, 2]];
d = T[[1, 4, 2]]; e = T[[1, 5, 2]]; f = T[[1, 6, 2]];
g = T[[1, 7, 2]]; h = T[[1, 8, 2]]; k = T[[1, 9, 2]];
b11 = (a * 4 + b * 2 + d * 2 + e  * 1 )/9;
b21 = (a * 2 + b * 4 + d * 1 + e  * 2 )/9;
b12 = (a * 2 + b * 1 + d * 4 + e  * 2 )/9;
b22 = (a * 1 + b * 2 + d * 2 + e  * 4 )/9;
b00 = (p + p1 + p4  + a )/4;
b10 = (p1 * 2 + p2 * 1 + a * 2 + b * 1)/6;
b20 = (p1 * 1 + p2 * 2 + a * 1 + b * 2)/6;
b30 = (p2 * 2 + p3 * 1 + b * 2 + c * 1)/6;
b01 = (p4 * 2 + p5 * 1 + a * 2 + d * 1)/6;
b02 = (p4 * 1 + p5 * 2 + a * 1 + d * 2)/6;
b03 = (p5 * 2 + p6 * 1 + d * 2 + g * 1)/6;
b31 = (b* 4 + c * 2 + e * 2 + f  * 1 )/9;
b32 = (b* 2 + c * 1 + e * 4 + f  * 2 )/9;
b33 = (e* 4 + f * 2 + h * 2 + k  * 1 )/9;
b13 = (d* 4 + e * 2 + g * 2 + h  * 1 )/9;
b23 = (d* 2 + e * 4 + g * 1 + h  * 2 )/9;
b00 = (b00 + b10 + b01 + b11)/4;
b30 = (b30 + b20 + b21 + b31)/4;
b03 = (b03 + b13 + b02 + b12)/4;
b33 = (b33 + b32 + b23 + b22)/4;
b10 = (b11 + b10)/2;b20 = (b21 + b20)/2;
b13 = (b12 + b13)/2;b23 = (b22 + b23)/2;
b01 = (b11 + b01)/2;b02 = (b12 + b02)/2;
b31 = (b31 + b21)/2;b32 = (b32 + b22)/2;
Collect[Simplify[{{b00, b10, b20, b30}, {b01, b11, b21, b31}, {b02, b12, b22, b32}, {b03, b13, b23, b33}}], {p, v
    , w}]
Collect[Simplify[{{b01 - b00, b11 - b10, b21 - b20, b31 - b30}, {b02 - b01, b12 - b11, b22 - b21,
    b32 - b31}, {b03 - b02, b13 - b12, b23 - b22, b33 - b32}}], {p, v,w}]
Collect[Simplify[{{b10 - b00, b20 - b10, b30 - b20}, {b11 - b01, b21 - b11, b31 - b21}, {b12 - b02, b22 - b12,
    b32 - b22}, {b13 - b03, b23 - b13, b33 - b23}}], {p, v, w}]
```