Available online at www.sciencedirect.com
ScienceDirect

Comput. Methods Appl. Mech. Engrg. 357 (2019) 112598

Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

A finite element framework based on bivariate simplex splines on triangle configurations

Juan Cao[a,b], Zhonggui Chen[c,*], Xiaodong Wei[d,e], Yongjie Jessica Zhang[d]

[a] *School of Mathematical Sciences, Xiamen University, Xiamen, 361005, China*
[b] *Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computation, Xiamen University, Xiamen, 361005, China*
[c] *Department of Computer Science, Xiamen University, Xiamen, 361000, China*
[d] *Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA*
[e] *Institute of Mathematics, Ecole Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland*

Received 16 April 2019; received in revised form 14 August 2019; accepted 18 August 2019
Available online 27 August 2019

Highlights

- A TCB-splines based method is proposed for FEA on general domains.
- Local refinement is flexible without introducing undesirable propagation.
- Spline functions are globally C^1-continuous.
- Optimal convergence rates are numerically demonstrated.

Abstract

Recently, triangle configuration based bivariate simplex splines (referred to as TCB-spline) have been introduced to the geometric computing community. TCB-splines retain many attractive theoretic properties of classical B-splines, such as partition of unity, local support, polynomial reproduction and automatic inbuilt high-order smoothness. In this paper, we propose a computational framework for isogeometric analysis using TCB-splines. The centroidal Voronoi tessellation method is used to generate a set of knots that are distributed evenly over the domain. Then, knot subsets are carefully selected by a so-called link triangulation procedure (LTP), on which shape functions are defined in a recursive manner. To achieve high-precision numerical integration, triangle faces served as background integration cells are obtained by triangulating the entire domain restricted to all knot lines, i.e., line segments defined by any two knots in a knot subset. Various numerical examples are carried out to demonstrate the efficiency, flexibility and optimal convergence rates of the proposed method.

Keywords: Simplex spline; TCB-splines; Isogeometric analysis; Triangulation

1. Introduction

Classical univariate B-splines have been comprehensively studied and widely used in today's industrial applications due to their superior capability of representing both free-form and common analytical shapes. Their

[*] Corresponding author.
E-mail address: chenzhonggui@xmu.edu.cn (Z. Chen).

https://doi.org/10.1016/j.cma.2019.112598

bivariate generalizations have also found many applications in the area of data fitting, geometric shape modeling and numerical analysis of partial differential equations (PDEs). These bivariate generalizations can be classified into tensor product splines and non-tensor product splines. Tensor-product splines use tensor product to define bivariate B-spline basis functions, typical examples including non-uniform rational B-splines (NURBS) [1] and T-splines [2]. NURBS are the de facto industry standard in commercial computer-aided design (CAD) systems. Despite their popularity, tensor product splines have some serious drawbacks. For example, tensor product surfaces are restricted to four-sided domains and thus have difficulty in modeling shapes with arbitrary topology. Although algorithm-wise complicated, non-tensor product splines provide alternative powerful tools in modeling complex shapes and approximating scattered data and functions. In addition to four-side domains, they can be defined over triangles [3], multi-sided domains [4], triangulations [5–7] and higher order triangulations [8,9]. For example, triangle configuration based bivariate simplex splines, referred to as TCB-splines [10], are generalizations of univariate B-splines.

Isogeometric analysis (IGA) has been introduced to seamlessly integrate CAD and traditional finite element method (FEM) by employing the same basis functions on both sides [11]. The initial investigations of IGA focused on NURBS due to their dominance in industrial CAD system. However, the inherent tensor-produce structure of NURBS make it difficult to represent a complex geometry or perform local refinement. To support local refinement, several strategies have been introduced, including T-splines [2,12], hierarchical B-splines [13], LR-splines [14,15], THB-splines [16,17], and hierarchical T-splines [18]. However, due to their (local) tensor-product structures, there are still restrictions on the refinement. For example, local refinement of T-splines generally propagates beyond the region of interest, whereas hierarchical B-splines require a minimum size of the to-be-refined region.

To overcome the tensor-product restriction, many advances have been made in using non-tensor product splines in IGA and FEM, such as triangular/tetrahedral Bézier patches [19–21], Powell–Sabin splines [22–27], triangular NURPS [28,29] and so forth. The approach using rational triangular Bézier patches focuses on automatically generating meshes of C^0-continuous Bernstein–Bézier triangles [20]. The triangular B-spline based approach relies on a reproducing kernel approximation technique to improve the stability of triangular B-spline in solving equations [30]. Powell–Sabin (PS) splines based IGA represents solutions by quadratic C^1 functions defined on the PS refinement of a given triangulation [22]. Even though C^1 quadratic splines are the most popular PS B-splines, there also exist PS B-splines of higher order smoothness and degree [31–33]. The properties of PS splines are closely related to the so-called PS triangles, which are not uniquely defined. To obtain PS splines with satisfying properties (e.g., nonnegativity) one may solve optimization problems or resort to other computationally more efficient alternatives [22]. For triangular Bézier based IGA, continuity constraints between adjacent Bézier patches need to be imposed in constructing globally C^r-continuous splines [19,34].

An alternative to the above so-called macro-elements is the spline spaces spanned by compactly-supported smooth functions. A well-understood example of such functions is the so-called simplex spline. There are several constructions of spaces using simplex splines, e.g., triangular B-splines [30], Delaunay configuration B-splines [9,35] and the latest triangle configuration B-splines (TCB-splines) [10,36]. We refer the reader to the survey for existing constructions of simplex spline spaces [9]. The space spanned by simplex splines has been largely ignored by the engineering community, perhaps due to their computational complexity [37,38]. Even so, some of the existing simplex spline spaces have been applied to applications such as data fitting [39], FEA [30,40] and collocation [41]. In particular, the latest construction is the so-called TCB-splines, which share many merits with traditional B-splines. Moreover, TCB-splines have appealing properties superior to tensor-product splines and other simplex spline spaces. For example, compared to tensor-product splines, TCB-splines support local refinement and are more flexible to accommodate general parametric domains. For triangular B-splines, one has to explicitly add the auxiliary knots to the given set of knots in advance in order to form a valid knot sequence for the basis construction. These auxiliary knots could affect the stability of triangular B-splines in solving partial differential equations [30]. In contrast, TCB-splines are free of auxiliary knots. Delaunay configuration B-spline (DCB-spline) relies on the Delaunay triangulation in selecting knot subsets. TCB-splines perform better than DCB-splines in modeling sharply varying surfaces [39].

The attractive theoretical properties of TCB-splines make them an ideal basis for IGA. This paper focuses on directly solving PDEs on complex polygonal domains without considering geometric map, which will lay the foundation to isogeometric analysis using general geometries. In our finite element framework, the solution field is represented by C^{k-1}-continuous TCB-splines. In particular, we look into the construction of TCB-spline space, discuss the numerical integration scheme and present adaptive knot placement using TCB-splines. We investigate the performance of TCB-splines in both uniform and adaptive refinement by showing numerical results on various well-known benchmark examples. Our approach is fully automatic and applicable to polygonal domains and allows local refinement. Different from previous work of using non-tensor product splines, our approach can automatically build C^{k-1}-continuous basis functions over polygonal domains, with no continuity constraints needed on convex polygonal parametric domains. For non-convex polygonal parametric domains, we can construct C^1-continuous geometric mapping by imposing n continuity constraints, where n is the number of concave corners.

The remainder of this paper is organized as follows: Section 2 covers the preliminaries of TCB-splines. In Section 3, we discuss the computation of TCB-splines, including the knot placement method, construction of C^{k-1}-continuous basis functions on convex domains and C^1-continuous basis functions on concave domains, and the numerical integration scheme for TCB-spline basis functions. In Section 4, we present experimental results by solving the Poisson and biharmonic equations on various polygonal domains. We present our conclusions, limitations, and suggestions for future work in Section 5.

2. TCB-Splines

TCB-splines are formed as a linear combination of simplex splines [42]. These simplex splines are defined over the so-called triangle configurations [10,39]. This section first introduces simplex splines and triangle configurations, and then explains how the basis functions of TCB-splines are constructed based on them. A detailed introduction to TCB-splines can be found in [10].

2.1. Simplex splines

Simplex splines are piecewise polynomials defined by a set of points (referred to as knots) in R^2 [42]. Given a set of three non-collinear knots $V = \{\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2\}$, a degree zero simplex spline defined by the knot set $V = \{\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2\}$ is simply a normalized characteristic function of the triangle formed by $\{\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2\}$,

$$M(\boldsymbol{u}|\{\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2\}) = \begin{cases} 0, & \boldsymbol{u} \notin [V), \\ 1/|area(V)|, & \boldsymbol{u} \in [V), \end{cases} \tag{1}$$

where $[\cdots)$ denotes the half-open convex hull of a set of points [43], which is a generalization of the half-open domain in R. A degree k simplex spline can be recursively evaluated as a linear combination of degree $k-1$ simplex splines using the Micchelli recurrence relation [44]. Given a set of knots $V = \{\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{k+2}\}$ from R^2 and an arbitrarily chosen knot set $X = \{\boldsymbol{t}_{i_0}, \boldsymbol{t}_{i_1}, \boldsymbol{t}_{i_2}\}$ consisting of three non-collinear knots from V, the Micchelli recurrence relation for a degree k simplex spline $M(\boldsymbol{u}|V)$ defined over V has the form:

$$M(\boldsymbol{u}|V) = \sum_{j=0}^{2} \lambda_j(\boldsymbol{u}|X) M(\boldsymbol{u}|V \backslash \{\boldsymbol{t}_{i_j}\}), \quad \boldsymbol{u} \in R^2, \tag{2}$$

where $\lambda_j(\boldsymbol{u}|X)$ are barycentric coordinates of $\boldsymbol{u}$ with respect to X satisfying $\sum_{j=0}^{2} \lambda_j(\boldsymbol{u}|X) = 1$ and $\sum_{j=0}^{2} \lambda_j(\boldsymbol{u}|X)\boldsymbol{t}_{i_j} = \boldsymbol{u}$.

Simplex splines share many properties with univariate B-splines. For example, a simplex spline defined over a knot set V is non-negative and has local support on the convex hull of the knots in V. A simplex spline defined over a knot set $V = \{\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{k+2}\}$ is a piecewise polynomial of degree k over the partition induced by the knots in V. Knots are said to be in *general position* if there are no coalescent knots and no three knots are collinear. It is known that if V is in general position, this simplex spline is C^{k-1}-continuous across the so-called knot lines, which are the segments $[\boldsymbol{t}_i, \boldsymbol{t}_j]$ with $i, j \in \{0, \ldots, k+2\}$ and $i \neq j$. We call these segments *knot lines*. On the other hand, the continuity order of a simplex function decreases if coalescent knots or collinear knots appear in V, which resembles the coalescent knots in the univariate B-spline case. If s knots in V are collinear, then the simplex is C^{k+1-s} continuous across this line. In particular, a degree k simplex spline (defined by $k+3$ knots) with $k+2$ collinear knots, when restricted on this line, degenerates to a univariate B-spline defined on these $k+2$ knots. A detailed theory of simplex splines can be found in [45].

Algorithm 1 Link triangulation procedure

```
Input: Degree $k$ $t$-config family $\Gamma_k$
Output: Degree $k+1$ $t$-config family $\Gamma_{k+1}$
Initialize $\Gamma_{k+1} = \emptyset$
for each $v$-config $J \in V(\Gamma_k)$ do
    Compute the polygon $P$ formed by the link of $J$, i.e., $P = \sum lk(J, \Gamma_k)$
    if $P \neq \emptyset$ then
        Partition the polygon $P$ into a set of triangles $C$
        $\Gamma_{k+1} \longleftarrow \Gamma_{k+1} \cup \{(T, J) | T \in C\}$
    end if
end for
```

2.2. *Link triangulation procedure (LTP)*

There are two main ingredients in the construction of TCB-spline: simplex splines and triangle configurations. The triangle configurations are knot subsets selected from a given knot set which are used for defining simplex splines. The *link triangulation procedure (LTP)* is a recursive algorithm that computes the family of triangle configurations such that the simplex splines defined on them span a space retaining the fundamental properties of univariate B-splines. To facilitate the explanation of LTP, we first introduce several necessary terminologies [10,39].

We denote the cardinality of a set X by $\#X$. Given a set of knots $K \subset R^2$ in general positions, a degree k *triangle-configuration* (t-config) is a pair of knot subsets (T, I) with $T, I \subset K$, $\#T = 3$ and $\#I = k$, where knot subsets T and I are referred to as the *first* and *second knot subsets* of the t-config (T, I), respectively. We denote a family of degree k t-configs by Γ_k. Note that an arbitrary triangulation of given knot set K corresponds to a family of degree zero t-configs Γ_0, where in each t-config, the first knot subset is formed by three vertices of a triangle and the second knot subset is empty; see Fig. 1(a). LTP starts with an arbitrary triangulation Γ_0 and recursively computes t-config families of higher degrees. For a degree k t-config (T, I), the first knot subset T corresponds to a counter-clockwise oriented triangle, and we further define the following terminologies.

- ***v*-config** is a knot subset of size $k+1$ formed by the union of a knot $\boldsymbol{v} \in T$ and the knot subset I, i.e., $\{\boldsymbol{v}\} \cup I$. Therefore, a degree k t-config can generate three degree $k+1$ v-configs. For example, a degree zero t-config $T = (\{\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2\}, \emptyset)$ generates three degree-one v-configs $\{\boldsymbol{t}_0\}$, $\{\boldsymbol{t}_1\}$ and $\{\boldsymbol{t}_2\}$.
- ***e*-config** is an oriented edge (of the oriented triangle formed by T) that is opposite to a knot $\boldsymbol{v} \in T$, denoted by ${}_{\boldsymbol{v}}T$. A degree k t-config can generate three e-configs. For example, a degree one t-config $T = (\{\boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3\}, \{\boldsymbol{t}_0\})$ generates three e-configs ${}_{t_1}T = \overrightarrow{\boldsymbol{t}_2\boldsymbol{t}_3}$, ${}_{t_2}T = \overrightarrow{\boldsymbol{t}_3\boldsymbol{t}_1}$ and ${}_{t_3}T = \overrightarrow{\boldsymbol{t}_1\boldsymbol{t}_2}$; see Fig. 1(c). A pair of opposite e-configs are two e-configs with the same endpoints but opposite orientations.
- ***e*-*v*-config pair** is a pair of an e-config and a v-config induced by a knot $\boldsymbol{v} \in T$, i.e., $({}_{\boldsymbol{v}}T, \{\boldsymbol{v}\} \cup I)$. A degree k t-config can generate three different e-v-config pairs. For example, a zero t-config $T = (\{\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2\}, \emptyset)$ generates e-v-config pairs $(\overrightarrow{\boldsymbol{t}_0\boldsymbol{t}_1}, \{\boldsymbol{t}_2\})$, $(\overrightarrow{\boldsymbol{t}_1\boldsymbol{t}_2}, \{\boldsymbol{t}_0\})$ and $(\overrightarrow{\boldsymbol{t}_2\boldsymbol{t}_0}, \{\boldsymbol{t}_1\})$.

Different degree k t-configs may have e-v-config pairs sharing a common v-config J. For example, there are five e-v-config pairs $(\overrightarrow{\boldsymbol{t}_1\boldsymbol{t}_2}, \{\boldsymbol{t}_0\})$, $(\overrightarrow{\boldsymbol{t}_2\boldsymbol{t}_3}, \{\boldsymbol{t}_0\})$, $(\overrightarrow{\boldsymbol{t}_3\boldsymbol{t}_4}, \{\boldsymbol{t}_0\})$, $(\overrightarrow{\boldsymbol{t}_4\boldsymbol{t}_5}, \{\boldsymbol{t}_0\})$ and $(\overrightarrow{\boldsymbol{t}_5\boldsymbol{t}_6}, \{\boldsymbol{t}_0\})$ are induced by t-configs $(\{\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2\}, \emptyset)$, $(\{\boldsymbol{t}_0, \boldsymbol{t}_2, \boldsymbol{t}_3\}, \emptyset)$, $(\{\boldsymbol{t}_0, \boldsymbol{t}_3, \boldsymbol{t}_4\}, \emptyset)$, $(\{\boldsymbol{t}_0, \boldsymbol{t}_4, \boldsymbol{t}_5\}, \emptyset)$ and $(\{\boldsymbol{t}_0, \boldsymbol{t}_5, \boldsymbol{t}_6\}, \emptyset)$, respectively. These five e-v-config pairs share a common v-configs $\{\boldsymbol{t}_0\}$; see Fig. 1(b), where e-configs are marked by red dash arrows. The set of e-configs from the e-v-config pairs sharing a common v-config J is called the *link* of the v-config J, denoted by

$$lk(J, \Gamma_k) := \{ {}_{v}T | v \in T, v \in J, (T, J \backslash \{v\}) \in \Gamma_k \}.$$

For a link $lk(J, \Gamma_k)$, we remove the opposite e-config pair if it exists. It has been proved that the remaining e-configs form a simple polygon (denoted by $\sum lk(J, \Gamma_k)$) by connecting the oriented edges at the coincident source and destination endpoints [10,46]. In particular, the link of a v-config J, which consists of a single vertex, is the set of oriented edges opposite to this vertex; see an example in Fig. 1(b), where $J = \{\boldsymbol{t}_0\}$ and the polygon

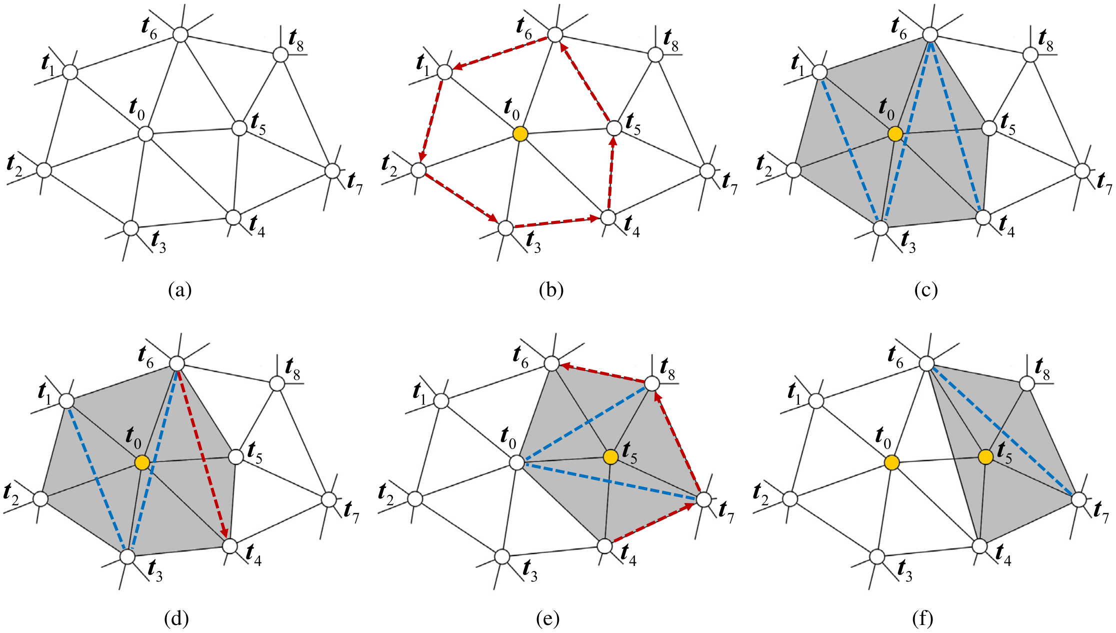


Fig. 1. The link triangulation procedure (LTP) and t-configs. (a) Initial triangulation, which corresponds to a family of degree-zero t-configs $\{(\{t_0, t_1, t_2\}, \emptyset), (\{t_0, t_2, t_3\}, \emptyset), (\{t_0, t_3, t_4\}, \emptyset), \ldots\}$; (b) a degree-one v-config $\{t_0\} \in V(\Gamma_0)$ marked by the solid yellow point and the e-v-config pairs $(\overrightarrow{t_1t_2}, \{t_0\})$, $(\overrightarrow{t_2t_3}, \{t_0\})$, $(\overrightarrow{t_3t_4}, \{t_0\})$, $(\overrightarrow{t_4t_5}, \{t_0\})$, $(\overrightarrow{t_5t_6}, \{t_0\})$ and $(\overrightarrow{t_6t_1}, \{t_0\})$ induced by degree zero t-config $(\{t_0, t_1, t_2\}, \emptyset)$, $(\{t_0, t_2, t_3\}, \emptyset)$, $(\{t_0, t_3, t_4\}, \emptyset)$, $(\{t_0, t_4, t_5\}, \emptyset)$, $(\{t_0, t_5, t_6\}, \emptyset)$ and $(\{t_0, t_6, t_1\}, \emptyset)$, respectively, where the e-configs are marked by red dash arrows; (c) the polygon $\sum lk(\{t_0\}, \Gamma_0) = [t_1, t_2, t_3, t_4, t_5, t_6]$ (filled in gray) formed by the links $lk(\{t_0\}, \Gamma_0) = \{\overrightarrow{t_1t_2}, \overrightarrow{t_2t_3}, \overrightarrow{t_3t_4}, \overrightarrow{t_4t_5}, \overrightarrow{t_5t_6}, \overrightarrow{t_6t_1}\}$ and its triangulation (blue dash lines), which yields four degree-one t-configs: $(\{t_1, t_2, t_3\}, \{t_0\})$, $(\{t_1, t_3, t_6\}, \{t_0\})$, $(\{t_6, t_3, t_4\}, \{t_0\})$ and $(\{t_6, t_4, t_5\}, \{t_0\})$; (d) the e-v-config pair $(\overrightarrow{t_6t_4}, \{t_0, t_5\})$ induced by degree-one t-config $(\{t_6, t_4, t_5\}, \{t_0\})$, where the e-config is marked by red dash arrows; (e) the e-v-config pairs $(\overrightarrow{t_8t_6}, \{t_0, t_5\})$, $(\overrightarrow{t_7t_8}, \{t_0, t_5\})$ and $(\overrightarrow{t_4t_7}, \{t_0, t_5\})$ sharing a common v-config $\{t_0, t_5\}$ are induced by degree-one t-configs $(\{t_6, t_0, t_8\}, \{t_5\})$, $(\{t_8, t_0, t_7\}, \{t_5\})$, $(\{t_7, t_0, t_4\}, \{t_5\})$, respectively, where e-configs are marked by red dash arrows; These three e-v-config pairs share a common v-config $\{t_0, t_5\}$ with the e-v-config pair $(\{t_6, t_4, t_5\}, \{t_0\})$ shown in (d); (f) the polygon $[t_8, t_6, t_4, t_7]$ (filled in gray) formed by link $lk(\{t_0, t_5\}, \Gamma_1)$ and its triangulation (blue dash line), generating two degree two t-configs $(\{t_8, t_6, t_7\}, \{t_0, t_5\})$ and $(\{t_7, t_6, t_4\}, \{t_0, t_5\})$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$\sum lk(\{t_0\}, \Gamma_0)$ is formed by connecting the one-ring neighbors of t_0 counterclockwise. We denote the collection of all v-configs of a degree k t-config family Γ_k by $V(\Gamma_k)$. LTP takes a family of degree k t-config Γ_k as input, and produces a family of degree $k+1$ t-config Γ_{k+1} as output. For each v-config $J \in V(\Gamma_k)$, we first obtain the polygon P formed by the link of J, i.e., $P = \sum lk(J, \Gamma_k)$; see Fig. 1(b) and (f) for examples of $\sum lk(\{t_0\}, \Gamma_k)$ and $\sum lk(\{t_0, t_5\}, \Gamma_k)$, respectively. If $P \neq \emptyset$, then we triangulate or partition P into a set of triangles C; see Fig. 1(c) and (f) for triangulation of polygons $\sum lk(\{t_0\}, \Gamma_0)$ and $\sum lk(\{t_0, t_5\}, \Gamma_1)$, respectively. Then, each triangle $T \in C$ and J form a degree $k+1$ t-config (see Fig. 1(c) and (f)), which is included into the $k+1$ t-config family, i.e., $\Gamma_{k+1} \longleftarrow \Gamma_{k+1} \cup \{(T, J) | T \in C\}$. The pseudo code of LTP is shown in Algorithm 1. We start our LTP with an arbitrary triangulation Γ_0 and repeatedly apply LTP k times to obtain degree k t-config family Γ_k. Fig. 1 shows examples of LTP up to degree 2, where the first (resp. second) row shows the procedure of generating degree-one (resp. two) t-configs from degree zero (resp. one) t-configs.

Remark 2.1. It is worth pointing out that LTP can yield different t-config families for a given set of knots if different triangulation methods are used to obtain the initial triangulation Γ_0 or to partition the polygon that appears in LTP. As a special case of TCB-splines, LTP yields the so-called Delaunay configurations when the Delaunay triangulation is adopted for both initial triangulation and polygon partition. The resulting t-config family is thus called Delaunay configuration family, and the associated simplex splines are called DCB-splines [9,35]. The criteria for an appropriate triangulation of LTP may vary with different applications. From geometric point of view, a feature sensitive triangulation method has been tailored for TCB-spline based surface fitting [39]. In Section 3.2, we will propose a triangulation method to generate TCB-spline basis functions that are globally at least C^1-continuous such that they can be used as a basis to solve high-order PDEs.

2.3. Definition of TCB-splines

With the family of degree k t-configs Γ_k, we can define simplex splines $M(u|T \cup I)$ for each degree k t-config (T, I). Then TCB-spline basis functions are linear combinations of these simplex splines. We denote by $\mathfrak{I}$ the set of all second knot subsets in Γ_k. Given $I \in \mathfrak{I}$, $X_I = \{(T, I^*)|I^* = I, (T, I^*) \in \Gamma_k\}$ is the set of t-configs sharing the common second knot subset I. The degree k TCB-spline basis function associated with a knot subset $I \in \mathfrak{I}$ is defined as [10]

$$B_I(u) = \sum_{(T,I)\in X_I} area(T)M(u|T \cup I), u \in R^2. \tag{3}$$

The Greville site ξ_I of $B_I(u)$ is defined as the average of the second knot subset I, i.e., $\xi_I = \frac{\sum_{v\in I} v}{k}$, which is the generalization of the univariate B-spline case. Fig. 2 shows examples of degree one to three TCB-spline basis functions. It has been proved that TCB-spline basis functions form a nonnegative partition of unity, possess automatic smoothness, have local support and reproduce polynomials.

Remark 2.2. TCB-splines may be linearly dependent. Although there is a method to eliminate linear dependence of TCB-splines in practice [39], there is no rule to guarantee linear independence of TCB-splines so far, leading to possibly singular stiffness matrix in analysis. There are two approaches to handle this problem. First, we could use Conjugate Gradients (CG) when the coefficient matrix has spurious zero eigenvalues. The CG method only requires products of the stiffness matrix with vectors. CG converges to the largest eigenvalues first, and then sweeps down through the lower modes in order. CG cannot converge to modes with zero eigenvalues. This approach has been used successfully in analogous situations. Second, we could use an LDU decomposition. Before performing forward reductions and back substitutions, we can check how many pivots (i.e., diagonal entries of the diagonal matrix D) are zero. The number of zero pivots equals to the number of zero eigenvalues. For non-zero pivots, the system is nonsingular and the solution can proceed. Otherwise, we need to switch to the CG approach. Note that we never found linear dependence in all examples of this paper when applying linear dependent elimination [39].

3. Computation of TCB-splines

In this section, we first introduce how to place knots for TCB-spline construction on a 2D bounded domain, which is the solution domain of a partial differential equation (PDE). Second, we describe LTP to guarantee C^{k-1}-continuity in convex regions and C^1-continuity in concave regions. Third, we explain in detail the integration of TCB-splines, which is a key step in assembling the stiffness matrix in IGA. Finally, we propose an error-guided knot placement method for TCB-splines to achieve adaptive refinement.

3.1. Evenly-distributed knot placement

To define the TCB-spline space, we first need to place a set of knots K. Here we use the centroidal Voronoi tessellations (CVT) [47] to generate knots in general positions that are evenly distributed within a given domain. CVT is a Voronoi tessellation with generators coinciding with the centroids for each Voronoi region. The generators of CVT, which are in general positions, are taken as the knots to define the TCB-spline space. As the CVT method generates evenly spaced knots in the interior domain, the associated degree k ($\geq$ 2) TCB-splines are C^{k-1}-continuous, and these TCB-splines as well as their derivatives vanish on the domain boundary. To guarantee the partition of unity property of TCB-splines on the domain boundary $\partial\Omega$, which is similar to the univariate case, we introduce coalescent and collinear knots along the boundary and compute the TCB-splines as follows.

- **Step 1.** The evenly distributed boundary knots and interior knots are generated by adjusting the CVT results. In particular, the knots (i.e., centroids) of the boundary Voronoi cells are projected onto Ω. Then, CVT is applied again to all the knots (i.e., both boundary and interior knots), where boundary knots are restricted to the boundary; see the first column of Fig. 2 as an example of the resulting knots.

- **Step 2.** The knot at a corner is repeated $k+1$ times for degree k TCB-splines. In other words, there are $k+1$ knots sharing the same position at the corner; such knots are coalescent.
- **Step 3.** The coalescent knots at each corner are perturbed such that they are in general positions. This step provides a topological structure (i.e., connectivity in the initial triangulation Γ_0) to compute the t-config family, and it will be further discussed in the next section.
- **Step 4.** We compute the t-config family and construct the corresponding simplex splines and let the perturbation size vanish. In other words, we merely treat coalescent knots separately in t-config family computation; the unperturbed knots are used in the computation of associated simplex splines.

The above knot placement method gives rise to a collection of TCB-splines that perform equivalently to univariate B-splines along the boundaries. Fig. 3 shows an example of a cubic TCB-spline basis function with five collinear knots on the boundaries, whose restriction to the boundary is the classical univariate cubic B-spline basis function. Fig. 4 shows an example of degree-two TCB-spline basis function defined on coalescent knots (with a multiplicity of three) at the upper left corner, whose restrictions to the boundary are the univariate quadratic B-spline basis functions on knots with a multiplicity of three.

Remark 3.1. The most straightforward approach to obtain evenly spaced knots is to use a regular grid of knots. However, knots forming such a grid are inappropriate for TCB-splines because it involves collinear knots, leading to reduced continuity of TCB-splines. To avoid collinear knots in the interior domain, we use the CVT method to generate knots that are distributed in a both uniform and random manner. Other methods, such as the Poisson-disk sampling method [48], may be also viable to generate knots in general positions.

Remark 3.2. In practice, collinear knots and coalescent knots are undesired in the interior of parametric domain as they reduce the continuity order of simplex splines. In our framework, we use the CVT method to generate evenly spaced knots given a pre-specified density. Although there is no guarantee that the CVT method can avoid collinear knots, numerically we never find collinear knots in the interior of parametric domain in our experiments. If in any case collinear knots appear, we can either perturb these knots or perform several more steps of Lloyd's iteration locally to numerically eliminate them.

3.2. C^1-continuity around corners

Note that TCB-splines defined with the perturbed knots may be only C^0-continuous in the interior domain of Ω. Similar to univariate B-splines of coalescent knots, if a t-config (T, I) contains both interior knots and coalescent knots of multiplicity $s > 1$, the continuity of the associated simplex will be reduced by s at the knot lines formed by interior knots and coalescent knots. To guarantee that TCB-splines possess C^1-continuity on the entire domain, we need to add several constraints to the triangulation referred in LTP from Algorithm 1.

We here use the quadratic case as an example to illustrate what these constraints are, but the same method also works for higher-degree cases. Fig. 5 shows a convex corner of the domain. With degree two, there are three coalescent knots ($\boldsymbol{t}_1, \boldsymbol{t}_2$ and $\boldsymbol{t}_3$) sharing the same position at this corner. To compute t-configs, we perturb these knots and consider two different initial triangulations shown in Fig. 5(a) and (b). From LTP we observe that, for a quadratic t-config, knots in the second knot subset form an edge of the initial triangulation, and knots in the first knot subset belong to the one-ring neighborhood of this edge; see the second row in Fig. 2 as an example. Knots in the second knot subset can also form a knot line of the associated simplex spline. Therefore, if we adopt the initial triangulation in Fig. 5(a), there may exist a quadratic t-config whose second knot subset is $\{\boldsymbol{t}_1, \boldsymbol{t}_5\}$ and the first knot subset contains $\boldsymbol{t}_2$ and/or $\boldsymbol{t}_3$. In other words, the associated simplex spline has a knot line $[\boldsymbol{t}_1, \boldsymbol{t}_5]$, where $\boldsymbol{t}_1$ is of multiplicity three. Thus, the continuity order of this quadratic basis function is decreased to C^{-1} across this knot line.

To achieve C^1-continuity, we prevent coalescent knots and its one-ring neighbors from connecting with interior knots in LTP. First, we introduce two new knots (orange points $\boldsymbol{t}_7, \boldsymbol{t}_8$ in Fig. 5(b)) on adjacent boundary edges of the corner such that the triangle $[\boldsymbol{t}_1, \boldsymbol{t}_7, \boldsymbol{t}_8]$ contains only two perturbed knots (i.e., $\boldsymbol{t}_2$ and $\boldsymbol{t}_3$) in its interior; see Fig. 5(b). Second, the triangle $[\boldsymbol{t}_1, \boldsymbol{t}_7, \boldsymbol{t}_8]$ and the quadrilateral $[\boldsymbol{t}_4, \boldsymbol{t}_6, \boldsymbol{t}_8, \boldsymbol{t}_7]$ (formed by the newly added knots and existing knots $\boldsymbol{t}_4$, $\boldsymbol{t}_6$) are further triangulated.; see Fig. 5(b). Thus, the interior knots are beyond the two-ring neighborhood of the coalescent knots, leading to t-configs that do not contain both coalescent knots and interior

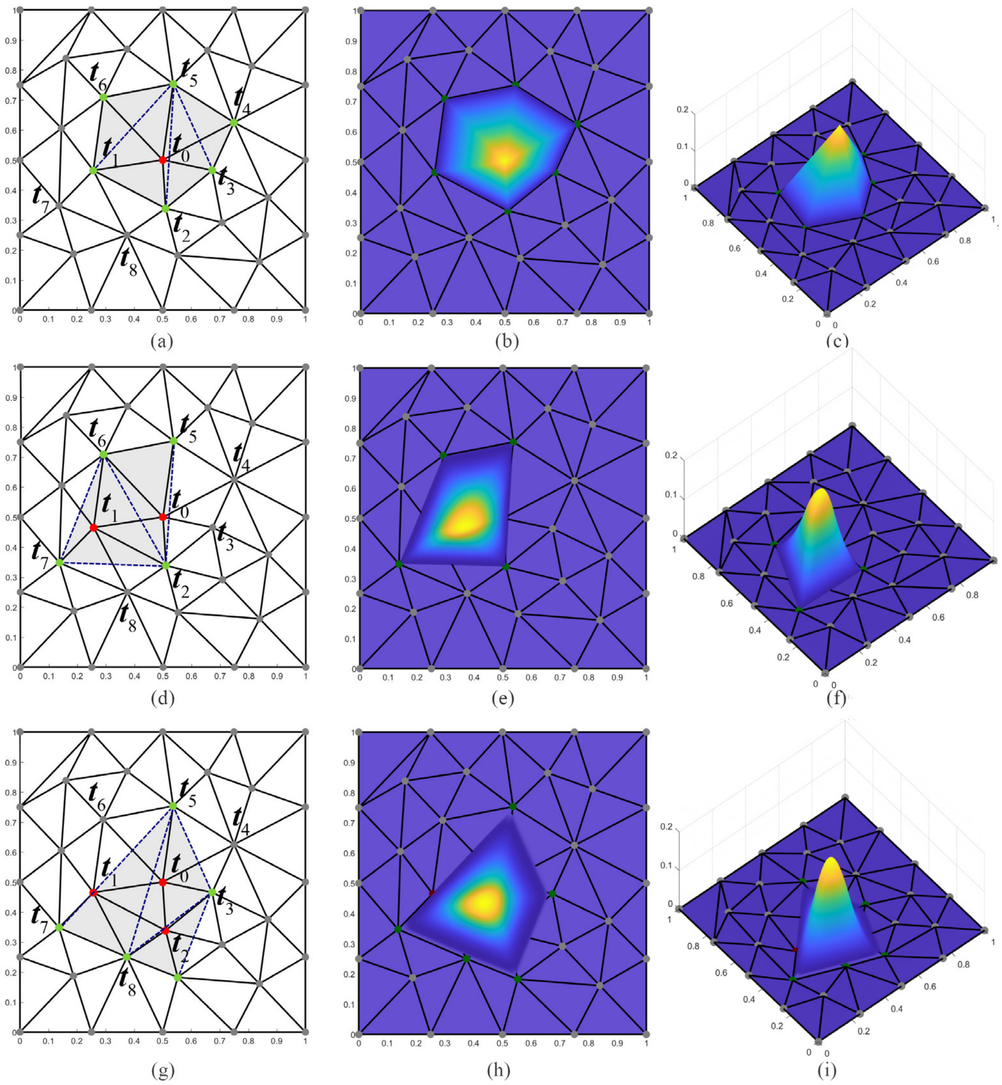


Fig. 2. Interior TCB-spline basis functions of degree one (a–c), degree two (d–f) and degree three (g–i). Columns from left to right: t-configs sharing a common second knot subset (where knots in the first and the second knot subsets are marked in green and red, respectively, and the triangles formed by the first knot subsets are marked in blue dash lines), top view of the associated TCB-spline basis functions and perspective view of the basis functions. (a–c) the linear TCB-spline basis function associated with the knot subset $I = \{\boldsymbol{t}_0\}$, where X_I consists of four degree-one t-configs: $(\{\boldsymbol{t}_5, \boldsymbol{t}_6, \boldsymbol{t}_1\}, \{\boldsymbol{t}_0\})$, $(\{\boldsymbol{t}_5, \boldsymbol{t}_1, \boldsymbol{t}_2\}, \{\boldsymbol{t}_0\})$, $(\{\boldsymbol{t}_5, \boldsymbol{t}_2, \boldsymbol{t}_3\}, \{\boldsymbol{t}_0\})$ and $(\{\boldsymbol{t}_5, \boldsymbol{t}_3, \boldsymbol{t}_4\}, \{\boldsymbol{t}_0\})$; (d–f) a quadratic TCB-spline basis function associated with the knot subset $I = \{\boldsymbol{t}_0, \boldsymbol{t}_1\}$, where X_I consists of two degree-two t-configs: $(\{\boldsymbol{t}_6, \boldsymbol{t}_7, \boldsymbol{t}_2\}, \{\boldsymbol{t}_0, \boldsymbol{t}_1\})$ and $(\{\boldsymbol{t}_6, \boldsymbol{t}_2, \boldsymbol{t}_5\}, \{\boldsymbol{t}_0, \boldsymbol{t}_1\})$ and (g–h) a cubic TCB-spline basis function associated with the knot subset $I = \{\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2\}$, where X_I consists of three degree-three t-configs: $(\{\boldsymbol{t}_5, \boldsymbol{t}_7, \boldsymbol{t}_8\}, \{\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2\})$, $(\{\boldsymbol{t}_5, \boldsymbol{t}_8, \boldsymbol{t}_3\}, \{\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2\})$ and $(\{\boldsymbol{t}_3, \boldsymbol{t}_8, \boldsymbol{t}_2\}, \{\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2\})$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

knots simultaneously. In this manner, simplex splines are guaranteed to be C^1-continuous around a convex corner. Consequently, the TCB-splines, as linear combinations of simplex splines, are globally C^1-continuous.

However, this idea cannot be directly adapted to define C^1 basis functions at a concave domain, because coalescent knots at a concave corner are inevitably connected to interior knots in a triangulation (see $\boldsymbol{t}_8$ in Fig. 6), giving rise to simplex splines with continuity reduced inside domain Ω. Let us consider again the quadratic case at a concave corner, as shown in Fig. 6. Knots $\boldsymbol{t}_1$, $\boldsymbol{t}_2$ and $\boldsymbol{t}_3$ are three coalescent knots at this corner, where $\boldsymbol{t}_2$ and $\boldsymbol{t}_3$ are perturbed. The two closest boundary knots to the concave corner are $\boldsymbol{t}_4$ and $\boldsymbol{t}_6$; see Fig. 6(a). The idea to achieve C^1 at a concave corner is to first restrict discontinuity of simplex splines to a single segment (i.e., $[\boldsymbol{t}_1, \boldsymbol{t}_5]$

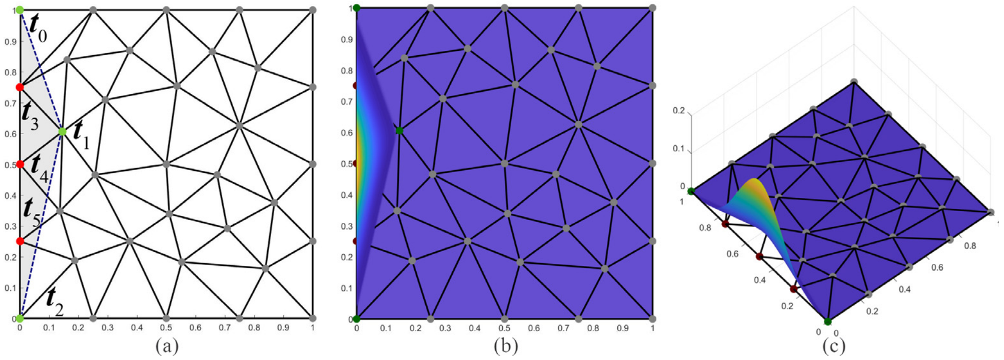


Fig. 3. A degree three TCB-spline basis at domain boundary. From left to right: t-configs, top view of the associated TCB-spline basis functions and perspective view of the basis function. In this example, the TCB-spline basis is the linear combination of a simplex spline defined over a single t-config ($\{t_0, t_1, t_2\}, \{t_3, t_4, t_5\}$), where the knots in the first and second knot subsets are marked by green and red dots, respectively, and the triangle formed by the first knot subset is marked by blue dash lines. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

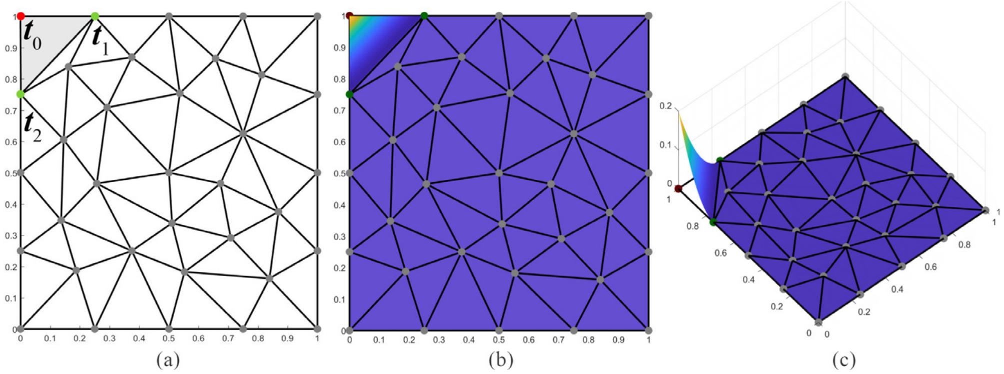


Fig. 4. A degree-two TCB-spline basis at a corner. From left to right: t-configs, top view of the associated TCB-spline basis functions and perspective view of the basis function. In this example, knots at the upper left corner have a multiplicity four and the TCB-spline basis is the linear combination of a simplex spline defined over a single t-config ($\{t_0, t_1, t_2\}, \{t_0, t_0\}$), where the knot in the second knot subset is marked by red (i.e., triple-knots at the upper left corner) and the knots in the first knot subset are marked by green with one hidden at the upper left corner, and the triangle formed by the first knot subset is filled in gray. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

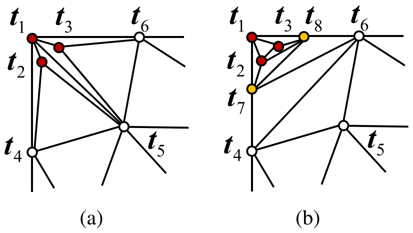


Fig. 5. C^1 conditions at a convex corner. (a) Coalescent knots (red dots) and its triangulation at a concave corner; and (b) coalescent knots are beyond the two-ring neighborhood of interior knots in the constrained triangulation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in Fig. 6(b)) and then to adjust the linear combination of these simplex splines to achieve a C^1 TCB-spline. More specifically, we first add a knot t_5 in the interior of the domain such that the concave region is split into two convex regions and no other knots (except the perturbed knots) locate in the interior of the polygonal region $[t_1, t_4, t_5, t_6]$;

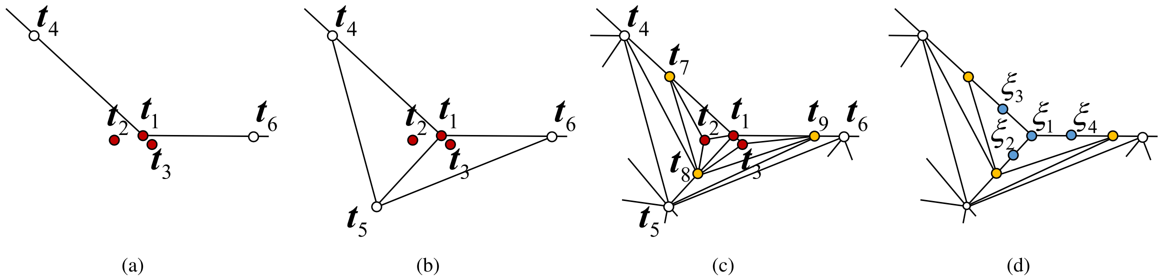


Fig. 6. C^1-conditions at a concave corner. (a) Coalescent knots at a concave corner; (b) a tailored knot line $[\boldsymbol{t}_1, \boldsymbol{t}_5]$, across which the simplex splines may not be C^1-continuous; (c) triangulation around the concave corner; and (d) Greville sites of C^0 basis functions.

see Fig. 6(b). For each convex corner $\widehat{\boldsymbol{t}_4\boldsymbol{t}_1\boldsymbol{t}_5}$ or $\widehat{\boldsymbol{t}_5\boldsymbol{t}_1\boldsymbol{t}_6}$, we add the new knots and perform constrained triangulation as we did in Fig. 5. In particular, the new knots $\boldsymbol{t}_7$, $\boldsymbol{t}_8$ and $\boldsymbol{t}_9$ are inserted on segments $[\boldsymbol{t}_1, \boldsymbol{t}_4]$, $[\boldsymbol{t}_1, \boldsymbol{t}_5]$ and $[\boldsymbol{t}_1, \boldsymbol{t}_6]$, respectively, such that the perturbed knots ($\boldsymbol{t}_2$ and $\boldsymbol{t}_3$) locate in the interior of polygonal region $[\boldsymbol{t}_1, \boldsymbol{t}_7, \boldsymbol{t}_8, \boldsymbol{t}_9]$. Thus $\boldsymbol{t}_2$ and $\boldsymbol{t}_3$ split triangles $[\boldsymbol{t}_1, \boldsymbol{t}_7, \boldsymbol{t}_8]$ and $[\boldsymbol{t}_1, \boldsymbol{t}_8, \boldsymbol{t}_9]$, respectively; see Fig. 6(c). In this way, the one-ring or two-ring neighboring knots of the coalescent knots $\boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3$ locate either on the boundary or on the segment $[\boldsymbol{t}_1, \boldsymbol{t}_5]$. Similar to the case of a convex corner, quadratic simplex splines are C^1-continuous in the interior of the convex region $\widehat{\boldsymbol{t}_4\boldsymbol{t}_1\boldsymbol{t}_5}$ or $\widehat{\boldsymbol{t}_5\boldsymbol{t}_1\boldsymbol{t}_6}$. In other words, simplex splines are only discontinuous across the segment $[\boldsymbol{t}_1, \boldsymbol{t}_5]$. The resulting TCB-splines, defined as linear combinations of these simplex splines, are C^0-continuous across the segment $[\boldsymbol{t}_1, \boldsymbol{t}_5]$ because discontinuous simplex splines cancel out in each linear combination [10,39]). Moreover, these C^0 basis functions are linearly combined to define a C^1 TCB-spline basis function. Assume that there are M C^0-continuous TCB-splines B_{I_i} ($i = 1, \dots, M$) with Greville sites $\xi_1, \dots, \xi_M$, where M is an integer depending on the degree and triangulation and in our experiments we have $M \in \{3, 4, 5\}$; see Fig. 6(d) as an example of $M = 4$. The linear combination of these basis functions

$$\sum_{i=1}^{M} c_i B_{I_i}$$

is not C^1-continuous for a general choice of coefficients $c_1, \dots, c_M$. However, if the points $(\xi_i, c_i) \in R^3$ are coplanar, the linearly combined splines can achieve C^1, which is similar to imposing C^1-constraints on Bézier patches [49]. Note that, coefficients satisfying coplanarity conditions are not unique. In our experiment, coplanarity conditions to ensure C^1 continuity around concave corners are explicitly enforced. In particular, the coefficients of C^0-continuous TCB-splines with Greville sites on the boundary are determined by boundary conditions, whereas coefficients of the remaining C^0-continuous TCB-splines are further determined by the coplanarity conditions at the concave corner. For example, we assign the coefficients c_i of the TCB-splines B_{I_i} with Greville sites ξ_i for $i = 1, 3, 4$ shown in Fig. 6(d) according to boundary conditions. To define a C^1 spline, c_2 is chosen such that point (ξ_2, c_2) is located on the plane formed by those three points corresponding to the boundary (ξ_i, c_i) ($i = 1, 3, 4$).

Discussion 3.1. *The number of C^1 TCB-splines after linear combinations is $\#\mathfrak{I} - \#\mathfrak{I}_{C^0} + n_{con}$, where $\#\mathfrak{I}$ and $\#\mathfrak{I}_{C^0}$ are numbers of original TCB-spline basis functions and C^0 basis functions, respectively, and n_{con} is the number of concave corners. As the coefficients are determined by the boundary conditions, there is no guarantee that C^1-continuous functions formed by the linear combination of C^0-continuous functions at each concave corner are non-negative.*

The idea can be generalized to other higher order TCB-splines to achieve C^1-continuity around concave corners. For a general polygonal domain, the number of coplanarity conditions is equal to the number of concave corners. It is also possible to achieve higher order continuity around concave corners by imposing stronger continuity conditions, which should be analogous to the case of Bézier patches, but discussion is beyond the scope of this paper.

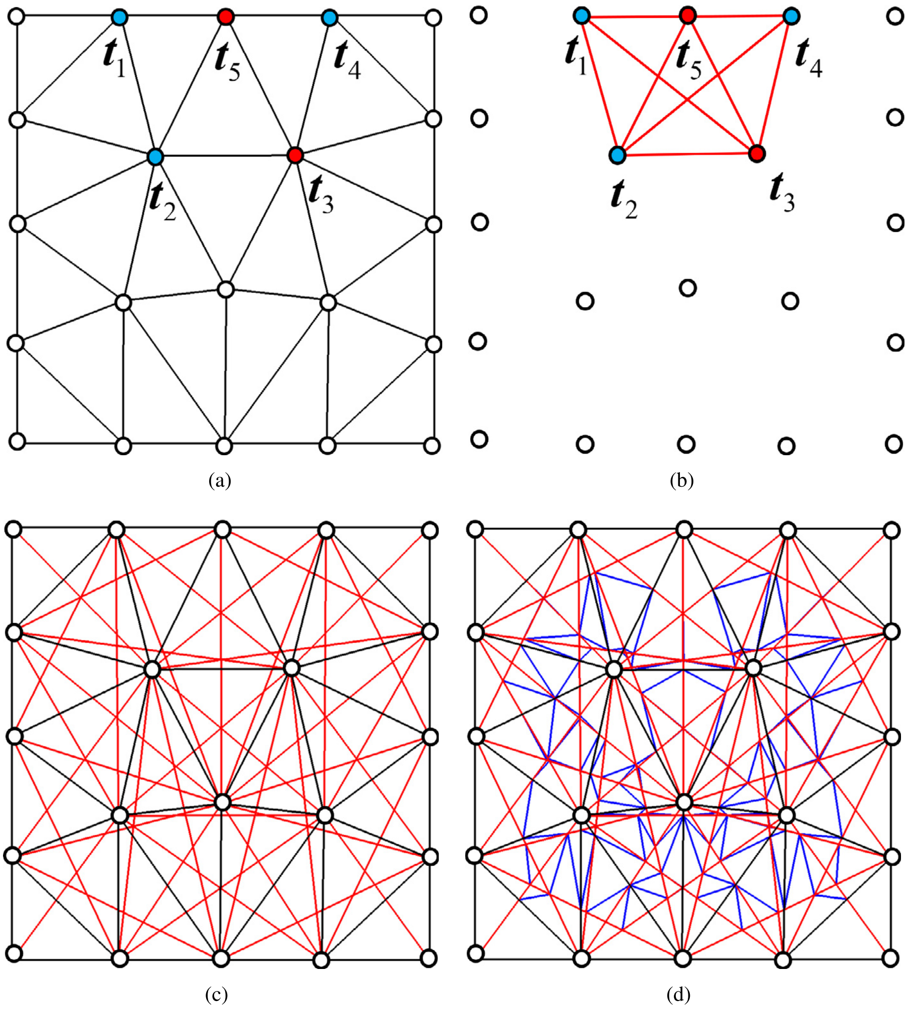


Fig. 7. Background integration cells. (a) Initial triangulation (24 triangles) of a knot set and a quadratic t-config ($\{\boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_4\}$, $\{\boldsymbol{t}_3, \boldsymbol{t}_5\}$), where knots in the first and second knot subsets are marked by blue and red dots, respectively; (b) knot lines (marked in orange) of the simplex spline associated with t-config in (a) and the partition of its support $[\boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3, \boldsymbol{t}_4, \boldsymbol{t}_5]$ induced by the knot lines; (c) knot lines of all simplex splines, which serve as the constraints in the triangulation to generate integration cells; and (d) triangulation (372 triangles) of knots restricted to all knot lines, where all the triangles serve as background integration cells in the numerical integration. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.3. Numerical integration for TCB-splines

Accurate numerical integration is important when using TCB-splines in IGA. It is not sufficient to only use quadrature points on triangles in the initial triangulation because simplex splines are only piecewise polynomials on each of the triangles. Fig. 7(a) shows an initial triangulation of a set of knots and a quadratic t-config ($\{\boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_4\}$, $\{\boldsymbol{t}_3, \boldsymbol{t}_5\}$), where knots in the first and second knot subsets are marked by blue and red dots, respectively. The simplex spline associated with this t-config is a piecewise quadratic polynomial over its support $[\boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3, \boldsymbol{t}_4, \boldsymbol{t}_5]$ (i.e., the convex hull of all the knots in the t-config). Note that, the simplex spline is a quadratic polynomial on each polygonal/triangular region partitioned by all its knot lines; see the orange lines and their induced polygonal regions in Fig. 7(b). To achieve high-precision numerical integration of this simplex spline, we further triangulate each polygonal region in Fig. 7(b) and apply the quadrature rule to each of the resulting triangles. More generally, we triangulate the entire domain restricted to all knot lines of all the simplex splines; see Fig. 7(c). Thus, any simplex spline on each resulting triangle is a polynomial, rather than a piecewise polynomial. We here adopt the constrained Delaunay triangulations from the Computational Geometry Algorithms Library (CGAL) to generate integration cells [50]. Fig. 7(c) shows the final triangulation restricted to all the knot lines. In each triangular integration cell, the 4-point rule [51] is adopted and it is exact for polynomials up to degree 5. It is obvious that the triangulation (372 triangles) serving as background integration cells in Fig. 7(d) is much finer than the initial triangulation (24 triangles) in Fig. 7(a).

3.4. Adaptive knot placement

B-splines or NURBS do not support local refinement due to their tensor-product structure. The generalizations of NURBS such as T-splines [2,12], truncated T-splines [52], hierarchical B-splines [13], LR-splines [14,15] and THB-splines [16,17] support local refinement. However, they are still based on local tensor product, and refinement in a specified region tends to propagate to adjacent regions where refinement is not needed. On the other hand, TCB-splines can flexibly insert knots to a specified region without introducing knots or refinement in other regions, and thus the refinement can be well controlled within a desired region. To demonstrate the flexibility of TCB-splines in adapting to functions that vary rapidly in local regions, we here present an adaptive knot placement strategy to generate localized knots and apply it to solving problems whose solutions contain singularity or sharp gradient.

We start from a set of evenly distributed knots K_0 of size N on domain Ω and use the approximation error to guide the adaptive knot insertion by the following iterative algorithm:

- **Step 1.** The problem (i.e., a certain PDE) is solved on the current set of knots K_n ($n \geq 0$). For each face T_i in the initial triangulation Γ_0^n of K_n, we compute the average L_2-norm error (denoted by $e_{L_2}|_{T_i}$) and the average H_1-norm error (denoted by $e_{H_1}|_{T_i}$) as follows:
$$e_{L_2}|_{T_i} = \frac{\sqrt{\int_{T_i} \|u - u_h\|^2 d\Omega}}{area(T_i)}, \quad e_{H_1}|_{T_i} = \frac{\sqrt{\int_{T_i} \|\nabla(u - u_h)\|^2 d\Omega} + \sqrt{\int_{T_i} \|u - u_h\|^2 d\Omega}}{area(T_i)}. \tag{4}$$
Then all the errors in $\{e_{L_2}|_{T_i}\}$ and $\{e_{H_1}|_{T_i}\}$ are normalized with respect to the total L_2 and H_1-norm errors, respectively.
- **Step 2.** $\alpha \cdot N$ faces T_i with the highest approximation error $e_i = \beta e_{L_2}|_{T_i} + (1-\beta) e_{H_1}|_{T_i}$ in the initial triangulation Γ_0^n are identified, where N is the number of faces in Γ_0. Their circumcenters, collected in a set U_i, are included into the knot set in the next step, i.e., $K_{n+1} = K_n \cup U_n$. Note that α is a factor to adjust the number of newly added knots U_n and β is a factor to balance the influence of two kinds of errors.
- **Step 3.** One step of Lloyd's relaxation, which moves a point towards the centroid of its Voronoi region [53], is applied to the knot set K_{n+1}. In this relaxation step, the old knots in K_n are fixed and the newly added knots in U_n are slightly adjusted such that the interior knots in K_{n+1} are in general positions. Then, a degree k t-config family is computed by applying LTP, where the triangulation is subject to the constraint that all the old knots and edges in Γ_0^n must stay in Γ_0^{n+1}.

The above three steps are repeatedly carried out to adaptively refine the knot set. In the end, we obtain TCB-splines supporting both evenly distributed and adaptive knot placement. The constructed TCB-splines are C^{k-1}-continuous in convex regions and C^1-continuous in concave regions, which are analysis-suitable for IGA applications.

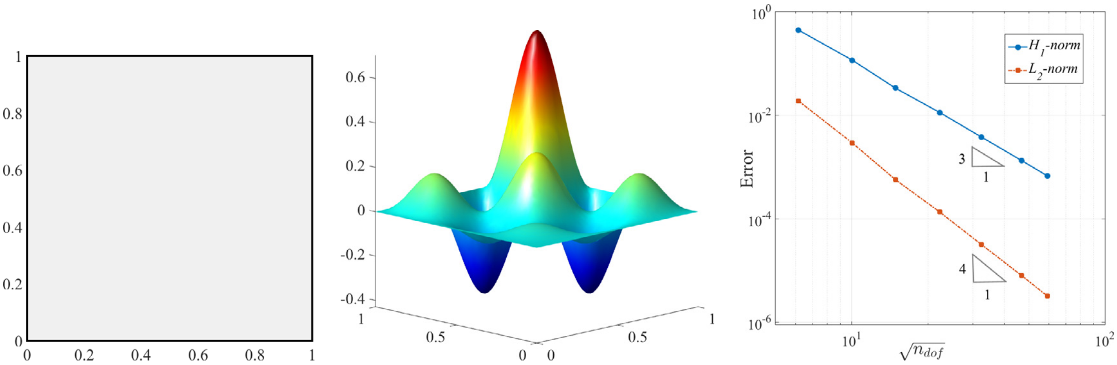
4. Numerical results

In this section, we construct TCB-splines on domains of widely varying shapes and use them to numerically solve the Poisson's and biharmonic equations. These examples are aimed to demonstrate that TCB-splines can serve as a candidate of IGA basis functions, flexibly accommodating complex geometries with automatic C^1-continuity and local adaptivity.

4.1. 2D Poisson equation

4.1.1. Model problem

We consider the following 2D Poisson's equation with the Dirichlet boundary condition:
$$\begin{cases} -\Delta u &= f \quad \text{on} \quad \Omega \\ u &= g \quad \text{on} \quad \partial\Omega \end{cases} \tag{5}$$



(a) Domain Ω (b) Exact solution (c) Convergence plot

Fig. 8. A Poisson problem on a square domain and approximation errors in the L_2-norm and H_1-norm versus $\sqrt{n_{dof}}$ obtained for evenly distributed knots using cubic TCB-splines.

where Ω is the solution domain and Δ is the Laplace operator. In its weak form, this amounts to find $u \in \{w : w \in H^1(\Omega)$ and $w = g$ on $\partial\Omega\}$ such that

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v d\Omega \quad \text{for all} \quad v \in w : w \in H^1(\Omega) \quad \text{and} \quad w = 0 \quad \text{on} \quad \partial\Omega. \tag{6}$$

Following the standard Galerkin approximation scheme, we discretize both u and v in terms of TCB-splines, that is, $u^h = \sum_{I\in\Im_v\cup\Im_b} c_I B_I$ and $v^h = \sum_{I\in\Im_v} d_I B_I$, where $c_I, d_I \in R$, $\Im_v$ is the index set of TCB-splines vanishing on the boundary, and $\Im_b$ contains those non-zero ones on the boundary. Substituting the discretized functions into the weak form and following the arbitrariness of v^h (i.e., arbitrary b_I), we obtain a system of linear equations to be solved for the unknown coefficients c_I:

$$A\begin{pmatrix} c_{I_0} \\ \vdots \\ c_{I_n}\end{pmatrix} = \begin{pmatrix} f_{I_0} \\ \vdots \\ f_{I_n}\end{pmatrix} \quad \text{with} \quad A_{I_i,I_j} = \int_\Omega \nabla B_{I_i} \cdot \nabla B_{I_j} d\Omega \quad \text{and} \quad f_{I_i} = \int_\Omega f B_{I_i} d\Omega. \tag{7}$$

The numerical integrations of (7) are performed via quadrature rule for triangles, where the triangular integration cells are generated by following the method described in Section 3.3. In what follows, we use cubic TCB-splines to solve the above problem.

4.1.2. Uniformly refined mesh

To study the convergence performance, we solve the problem on a series of meshes by increasing the number of knots, which are evenly distributed by the CVT method. Both the L_2-norm error and the H_1-norm error are calculated.

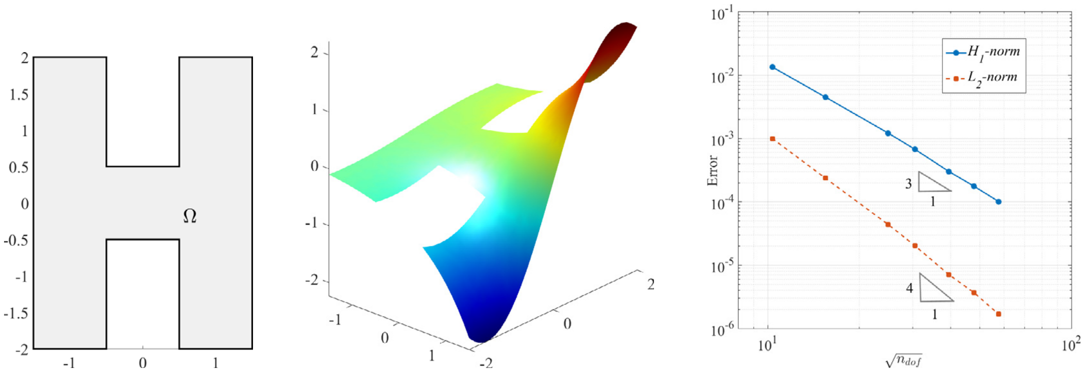
Example 1. We start with a square domain $\Omega = [0, 1] \times [0, 1]$ with the following manufactured solution:

$$u(x, y) = xy \sin(3\pi x) \sin(3\pi y).$$

Here we only have a homogeneous Dirichlet boundary condition. The domain Ω, exact solution and the resulting errors in L_2-norm and H_1-norm with respect to square root of degrees of freedom (DOF) are presented in Fig. 8. We observe that the optimal convergence rates (4 for the L_2 error and 3 for the H_1 error) are achieved by using cubic TCB-splines.

Example 2. In the second example, we consider an H-shape domain as shown in Fig. 9(a), with the following manufactured solution

$$u(x, y) = \frac{1}{2} e^x \sin(y).$$



(a) Domain Ω (b) Exact solution (c) Convergence plot

Fig. 9. A Poisson problem on an H-shape domain and approximation errors in the L_2-norm and H_1-norm versus $\sqrt{n_{dof}}$ obtained for evenly distributed knots using cubic TCB-splines.

Exact solution and the resulting errors in L_2-norm and H_1-norm with respect to square root of DOF are presented in Fig. 9(b–c). The convergence rates displayed in Fig. 9(c) again demonstrate the optimal approximation power of TCB-splines.

4.1.3. Adaptively refined mesh

In the following two examples, we apply the refinement strategy in Section 3.4 to solving two problems with known analytic solutions that contain singularity and sharp gradient.

Example 3. In this example, we solve a Poisson problem (5) on a unit square with exact solution given by

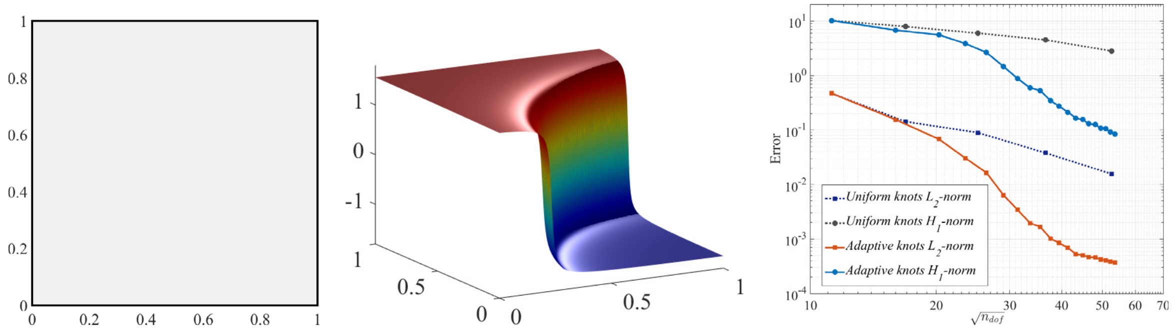
$$u(x, y) = arctan(100(\sqrt{(x-1.25)^2+(y+0.25)^2} - \frac{\pi}{3})). \tag{8}$$

We choose $\alpha = 0.8$ and $\beta = 1$, i.e., we use error in L_2 norm to control the distribution of new knots. As depicted in Fig. 10(b), the solution changes rapidly across a circular band in the domain. Although the solution field is smooth, optimal convergence rates can only be achieved when the size of triangle is small enough. In other words, a large number of knots are needed if we use the evenly distributed knot placement method. In order to solve this problem efficiently, we adopt the adaptive knot placement method to obtain a much faster convergence with a relatively small number of DOF. In Fig. 11, we observe that the errors dominate in the vicinity of the circular band. Adaptive knot placement iterations and distribution of the newly added knots are consistent with the errors. Fig. 12 shows knot sets in different refinement steps in the adaptive knot placement algorithm. We can observe clearly that the triangles are locally refined in the circular band. Results in Fig. 10(c) show that TCB-splines based on adaptive refinement converge much faster and yield much better accuracy per DOF than that based on uniform refinement.

Example 4. In this example, we solve the Laplace equation (5) on an L-shaped domain with the following exact solution

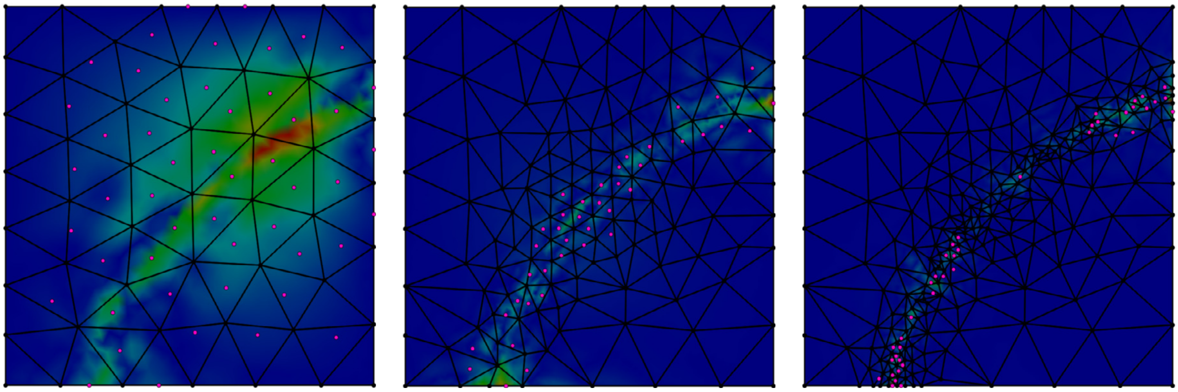
$$u(r, \theta) = r^{\frac{2}{3}} \sin \frac{(2\theta+\pi)}{3}, \tag{9}$$

where the right hand side $f = 0$ and g is given by the exact solution. The exact solution has a singularity at the origin, with a partial derivative not well defined. For such problems, convergence rates are controlled by the singularity rather than by the polynomial degree. Hence, optimal convergence rates cannot be achieved by using uniformly refined knot sets; see Fig. 14(c). Alternatively, the adaptive knot placement method is adopted to provide sufficient refinement around the singularity point, so a much faster convergence can be achieved. In this example, we choose $\alpha = 0.8$ and $\beta = 0$, i.e., we use error in H_1-norm to control the distribution of new knots. As has been shown in Fig. 15, our adaptive knot placement method is able to locally refine the triangles around the singular



(a) Square domain Ω (b) Exact solution (c) Convergence plot

Fig. 10. A Poisson problem with a sharp internal layer and the approximation errors measured in the L_2-norm and H_1-norm versus $\sqrt{n_{dof}}$ for evenly distributed knots and adaptive knot placement method using cubic TCB-splines.



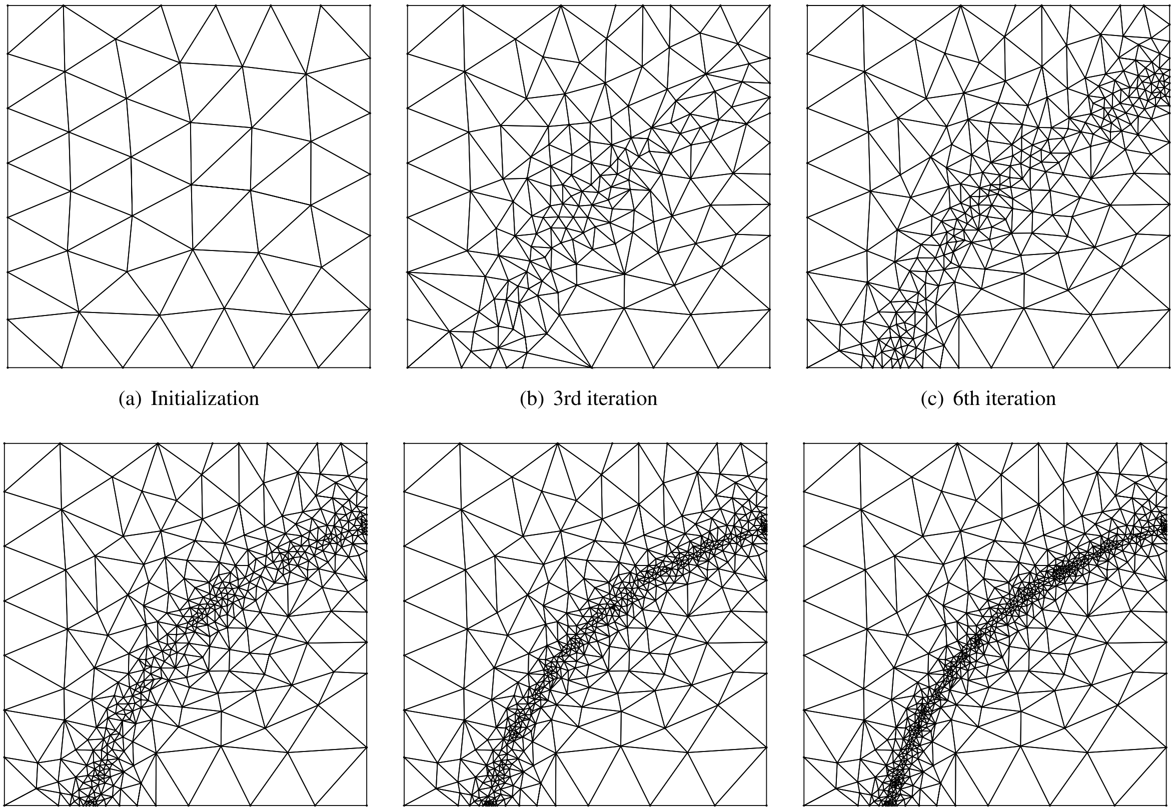
(a) U_0 (b) U_3 (c) U_6

Fig. 11. Newly added knots in the adaptive knot placement method in Example 3, where newly added knots U_n ($\#U_n = 40$) are marked in pink, and old knots K_n and degree zero t-configs Γ_0^n are marked in black dots and black solid lines, respectively.

region. Results in Fig. 14(c) shows that adaptive knot placement enables us to achieve much faster convergence rates than the evenly distributed knot placement method.

Discussion 4.1. *In adaptive mesh refinement, we use either the L_2- or the H_1-norm error to guide the refinement. In Example 4, the exact solution has a singularity at the concave corner, with a partial derivative not well defined. We use an approximation error in H_1-norm to guide the distribution of knots, where the error of the first order derivatives dominates. Thus, newly added knots are apt to concentrate around the singularity to effectively reduce the H_1-norm error. On the other hand, as the exact solution itself varies smoothly, the region away from the singularity should also be refined to further reduce the L_2-norm error, which, however, is not happening as we primarily use the H_1 error to guide the refinement. Hence, the error in L_2-norm levels off in this example.*

Discussion 4.2. *Narrow triangles are usually undesired in finite element method for solving partial different equations to avoid the numerical stability problem. However, we observe that the construction of TCB-spline basis and their application in solving the Poisson equation are not influenced by such triangles. We consider an extreme case as an example shown in Fig. 13, where 40 knots are evenly distributed and we use Delaunay triangulation in Fig. 13(a). Some edges are flipped locally to create a narrow triangle in Fig. 13(b). The condition number of the obtained stiffness matrices on both triangulations are 287.9311 (Fig. 13(a)) and 285.9580 (Fig. 13(b)), respectively. Numerically, narrow triangles do not seem to be an issue in our experiment. However, our understanding of the relationship between the knot mesh (including knot positions and their connectivity), numerical accuracy*


(a) Initialization

(b) 3rd iteration

(c) 6th iteration

(d) 9th iteration

(e) 12th iteration

(f) 15th iteration

Fig. 12. The adaptively placed knot set and Γ_0^n. A total of 50 evenly distributed knots are used in the initialization stage (a) and 40 new knots are added in each following iteration step.

and stiffness matrix condition number is far from complete. More sufficient theoretical analysis and numerical experiments on the relationship between knot meshes and matrix condition are needed.

4.2. Biharmonic equations

4.2.1. Model problem

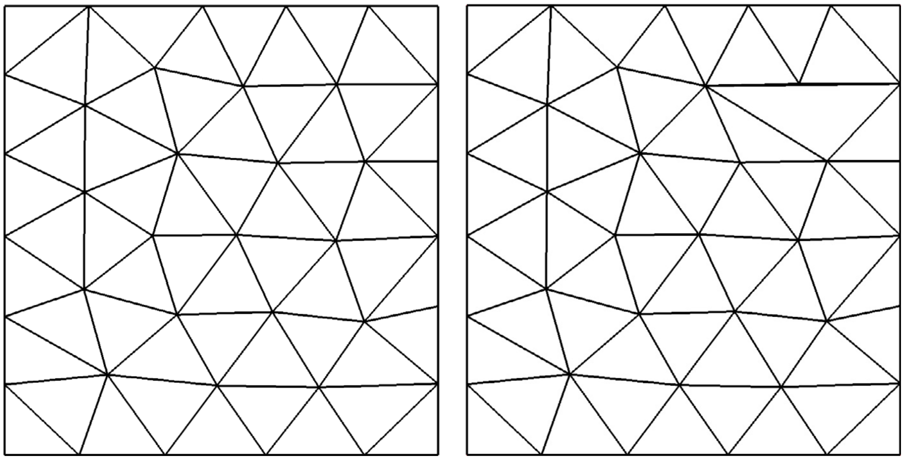
Global C^1-continuity is necessary for solving fourth-order PDEs. The biharmonic equation is a typical example of such PDEs. In this section, we construct C^1 TCB-splines following the strategies in Section 3.2 and solve the biharmonic equation on a domain Ω subject to homogeneous boundary conditions on the domain boundary $\partial\Omega$:

$$\left\{ \begin{array}{rcll} \Delta^2 u & = & f & \text{on} \quad \Omega \\ u & = & 0 & \text{on} \quad \partial\Omega \\ \frac{\partial u}{\partial \mathbf{n}} & = & 0 & \text{on} \quad \partial\Omega \end{array} \right. \tag{10}$$

where $\mathbf{n}$ is the outward normal on the boundary $\partial\Omega$. In our tests, the data f is obtained from a certain manufactured solution u. Reformulating Eq. (10) in the weak form, we find $u \in H_0^2(\Omega)$ such that

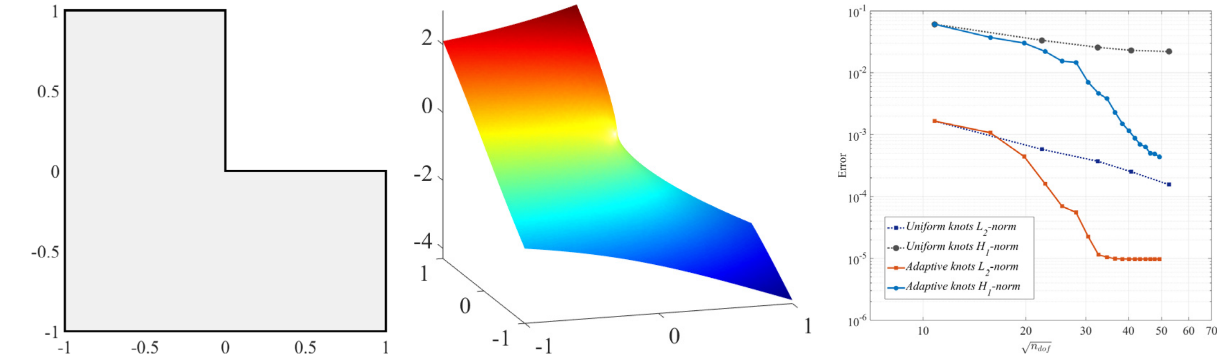
$$\int_\Omega \Delta u \Delta v dxdy = \int_\Omega f v dxdy, \tag{11}$$

for all $v \in H_0^2(\Omega)$, $v = 0$ and $\Delta v \cdot \mathbf{n} = 0$ on $\partial\Omega$. Following the same Galerkin procedure as described in Section 4.1.1, we end up with a system of linear equations. In what follows, we solve the biharmonic equations on a square domain and an L-shape domain.



(a) (b)

Fig. 13. Two different triangulations on the same set of knots. (a) Delaunay triangulation (condition no.: 287.9311); and (b) triangulation with a thin triangle (condition no.: 285.9585).



(a) L-shape domain Ω (b) Exact solution (c) Convergence plot

Fig. 14. A Poisson problem with a singularity point on the boundary and approximation errors in the L_2-norm and the H_1-norm versus $\sqrt{n_{dof}}$ for evenly distributed knots and adaptive knot placement using cubic TCB-splines.

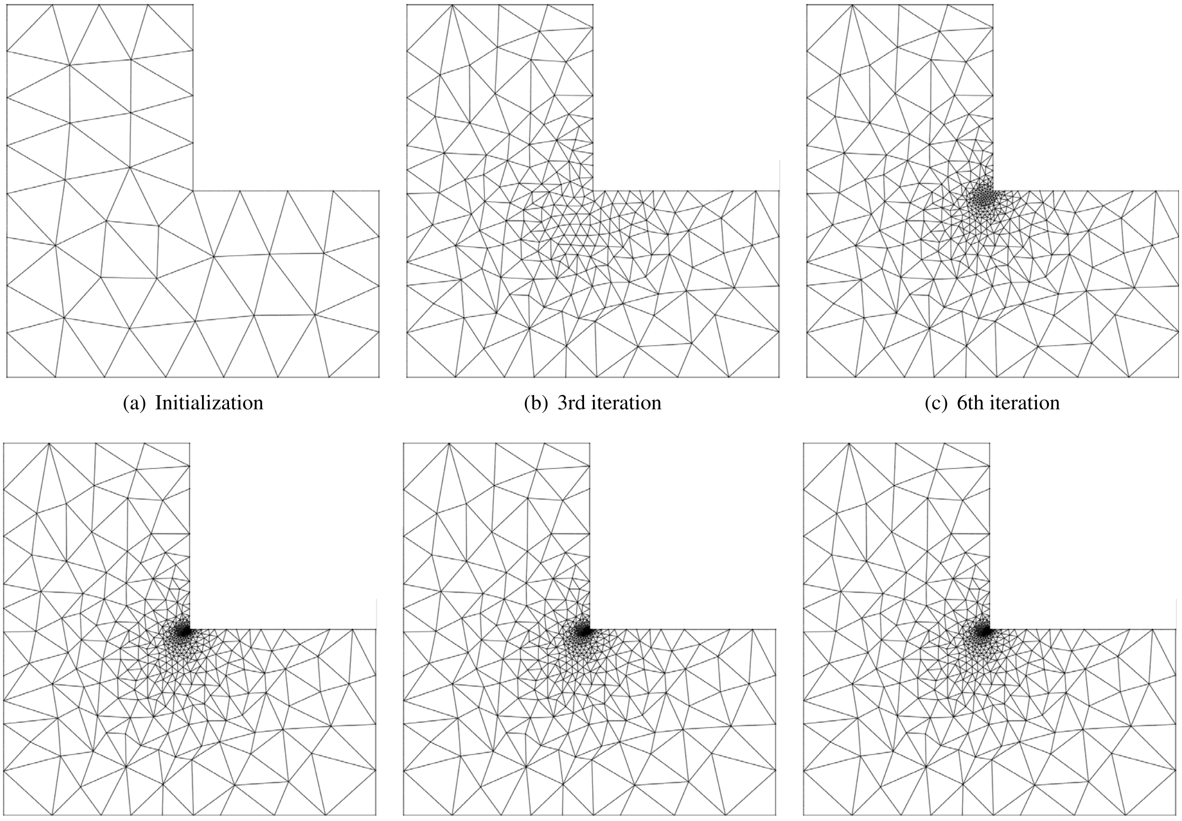
4.2.2. Uniformly refined mesh

To study the convergence performance, we solve the problem on a series of meshes by increasing the number of knots. The knots are evenly spaced using by the CVT method. Both the L_2-norm error and the H_1-norm error are reported.

Example 5. We consider the biharmonic equation (10) on a square domain with $\Omega := [-1, 1] \times [-1, 1]$ and f is given by the exact solution

$$u(x, y) = \frac{25}{\pi^4}(1 + \cos(\pi x))(1 + \cos(\pi y)).$$

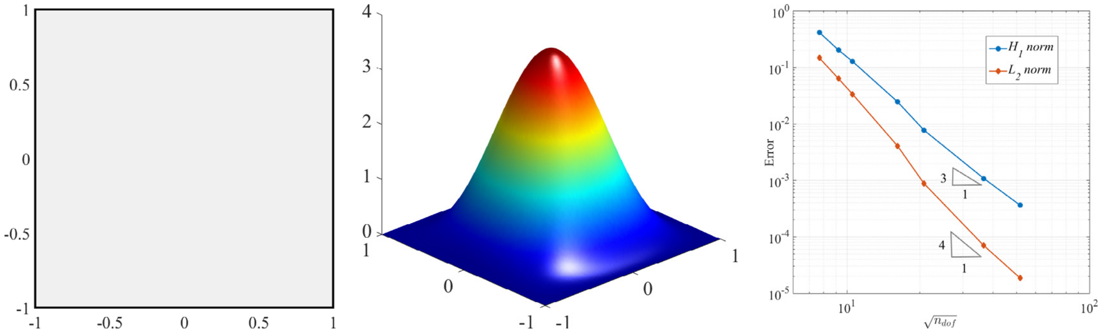
We generate different numbers of evenly distributed knots on Ω using the CVT method and solve the equation using cubic TCB-splines. Fig. 16 displays the solution domain, exact solution and L_2 and H_1 errors with respect to root square of DOF. We again observe optimal convergence rates for all the errors.



(a) Initialization (b) 3rd iteration (c) 6th iteration

(d) 9th iteration (e) 12th iteration (f) 15th iteration

Fig. 15. The adaptively placed knot set and Γ_0^n in Example 4. A total of 50 evenly distributed knots are used in the initialization stage (a) and 40 new knots are added in each following iteration step.



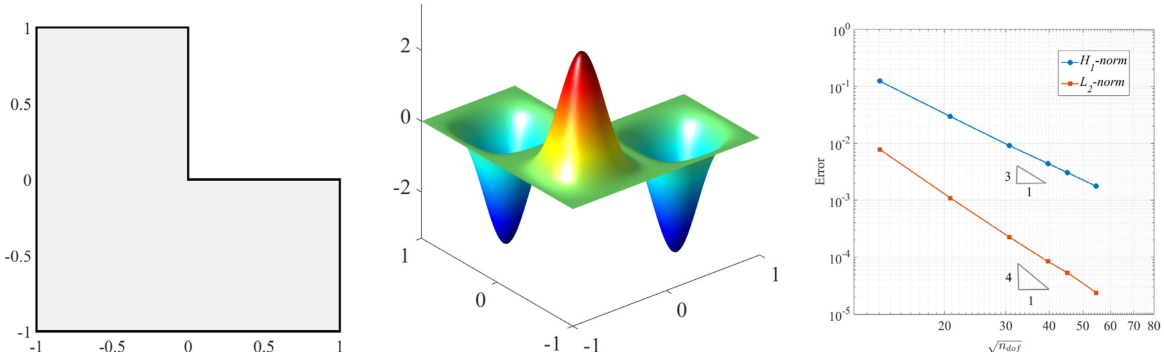
(a) Square domain Ω (b) Exact solution (c) Convergence plot

Fig. 16. A biharmonic problem on a square domain and approximation errors of cubic TCB-splines defined over evenly distributed knots in the L_2-norm and H_1-norm versus $\sqrt{n_{dof}}$.

Example 6. We next consider the biharmonic equation (10) on an L-shape domain $\Omega := [-1, 1] \times [-1, 1] \setminus [0, 1] \times [0, 1]$, where f is given by the exact solution

$$u(x, y) = \sin(\pi x) \sin(\pi y)(\cos^2(\pi x) - 1)(\cos^2(\pi y) - 1).$$

As shown in Fig. 17(c), the optimal convergence rates can be achieved.



(a) Square domain Ω (b) Exact solution (c) Convergence plot

Fig. 17. A biharmonic problem on an L-shaped domain and approximation errors of cubic TCB-splines defined over evenly distributed knots in the L_2-norm and H_1-norm versus $\sqrt{n_{dof}}$.

To summarize, we have numerically studied the TCB-spline based IGA in solving both Poisson and bi-harmonic problems over various domains. Numerical experiments have shown optimal convergence rates in terms of both H_1- and L_2-norms in all the numerical tests. We have further applied adaptive knot placement to solve the Poisson problem whose solution has singularity or sharp gradient. We found that TCB-splines based on adaptive knot placement clearly exhibits superior convergence over those based on uniform knot placement, where the adaptive case achieves the same accuracy with much fewer degrees of freedom.

Discussion 4.3. *Convergence rate is widely investigated to evaluate the performance of numerical schemes for solving PDEs. Several papers in the IGA community have reported numerical evidence of optimal convergence rates on general 2D domains with a globally smooth construction (i.e., at least C^1 everywhere). Among them, the work related to tensor-product splines includes Hermite splines [54], manifold splines [55], analysis-suitable G^1 parameterizations [56] and splines based on degenerated Bézier patches [57,58], whereas rational triangular Bézier spline [34] is the work related to non-tensor product splines. The investigation on general 3D domains is limited. In tensor product splines, only one literature shows optimal convergence rates for 3D with a blended B-spline construction [59], which, however, only possesses C^0 continuity in irregular regions. In non-tensor product splines, rational trivariate Bézier–Bernstein polynomials [21] can achieve optimal convergence rates with a global C^1 space using degree-5 splines. PS B-spline is also a C^1 non-tensor product spline that can achieve optimal convergence rates [22]. Theoretical analysis of the optimal convergence behavior with TCB-splines in IGA is not trivial, which is part of our future work.*

5. Conclusion and future work

In this paper, we have presented a new IGA framework based on TCB-splines. Due to the superior properties of TCB-splines, our framework is able to solve equations on general domains. Local refinement is flexible without introducing undesirable propagation in neighboring regions. Unlike Bézier patches based IGA, which requires imposing continuity constraints between adjacent patches [19], our method can construct globally C^{k-1}-continuous basis on convex domain automatically and C^1-continuous basis functions on concave regions by imposing a small number of continuity constraints (which equals to the number of concave corners). The application of TCB-splines in IGA has been demonstrated by solving several examples, including the Poisson's equation using evenly distributed knots and adaptively placed knots, and the biharmonic equation using evenly distributed knots. For all these applications, numerical results have shown optimal convergence rates.

Despite the generally promising results shown in the paper, our framework suffers from two major limitations: (1) TCB-splines may possess linear dependency and may limit their use in applications that rely heavily on the property of linear independence; and (2) to achieve high-precision integration, the triangulation used for integration is much finer than the triangulation used for defining basis, leading to a great computational burden. Hence, one of our future work is to study how to identify and theoretically eliminate the linear dependency that may exist in

TCB-splines. Another aspect we would like to improve is integration. To achieve efficient computation, a more sophisticated evaluation method and data structure are desired to make TCB-splines more appealing for practical industrial applications. Moreover, the influence of narrow triangles on their resulting stiffness matrices needs further investigation especially in theory. Finally, theoretical analysis of optimal convergence rates and extending TCB-splines to 3D are two other potential directions to explore.

Acknowledgments

The research of Juan Cao and Zhonggui Chen was supported by the National Natural Science Foundation of China (Nos. 61872308, 61972327, 61572020), the Natural Science Foundation of Fujian Province of China (Nos. 2018J01104, 2019J01011703), and the Fundamental Research Funds for the Central Universities, China (Nos. 20720190011, 20720190063). The research of Xiaodong Wei and Yongjie Jessica Zhang was supported in part by the PECASE award N00014-16-1-2254, NSF CAREER Award OCI-1149591, NSF CBET-1804929, CMU Manufacturing Futures Initiative and CMU-PITA (USA).

References

[1] L. Piegl, W. Tiller, The NURBS Book, Springer Science & Business Media, 2012.
[2] T.W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCs, ACM Trans. Graph. 22 (3) (2003) 477–484.
[3] G. Farin, Triangular Bernstein–Bézier patches, Comput. Aided Geom. Design 3 (2) (1986) 83–127.
[4] J. Gregory, N-sided surface patches, Math. Surf. (1986) 217–232.
[5] M.-J. Lai, L.L. Schumaker, Spline Functions on Triangulations, Cambridge University Press, 2007.
[6] W. Dahmen, C.A. Micchelli, H.-P. Seidel, Blossoming begets spline bases built better by patches, Math. Comp. 59 (199) (1992) 97–115.
[7] J. Cao, J. Zheng, Bivariate splines over triangular meshes for freeform surface modeling with sharp features, Comput.-Aided Des. Appl. 14 (4) (2017) 498–506.
[8] M. Neamtu, Bivariate simplex B-splines: a new paradigm, in: IEEE Spring Conference on Computer Graphics, 2001, pp. 71–78.
[9] M. Neamtu, What is the natural generalization of univariate splines to higher dimensions, in: Mathematical Methods for Curves and Surfaces: Oslo 2000, Citeseer, 2001, pp. 355–392.
[10] Y. Liu, J. Snoeyink, Quadratic and cubic B-splines by generalizing higher-order Voronoi diagrams, in: Symposium on Computational Geometry, 2007, pp. 150–157.
[11] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (39) (2005) 4135–4195.
[12] T.W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, J. Zheng, T. Lyche, T-spline simplification and local refinement, ACM Trans. Graph. 23 (3) (2004) 276–283.
[13] D.R. Forsey, R.H. Bartels, Hierarchical B-spline refinement, SIGGRAPH Comput. Graph. 22 (4) (1988) 205–212.
[14] T. Dokken, T. Lyche, K.F. Pettersen, Polynomial splines over locally refined Box-partitions, Comput. Aided Geom. Design 30 (3) (2013) 331–356.
[15] K.A. Johannessen, T. Kvamsdal, T. Dokken, Isogeometric analysis using LR B-splines, Comput. Methods Appl. Mech. Engrg. 269 (2014) 471–514.
[16] C. Giannelli, B. JüTtler, H. Speleers, THB-splines: The truncated basis for hierarchical splines, Comput. Aided Geom. Design 29 (7) (2012) 485–498.
[17] C. Giannelli, B. Jüttler, S.K. Kleiss, A. Mantzaflaris, B. Simeon, J. Špeh, THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 299 (2016) 337–365.
[18] E.J. Evans, M.A. Scott, L. Xin, D.C. Thomas, Hierarchical analysis-suitable T-splines: Formulation, Bézier extraction, and application as an adaptive basis for isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 284 (8) (2014) 559–560.
[19] N. Jaxon, X. Qian, Isogeometric analysis on triangulations, Comput. Aided Des. 46 (2014) 45–57.
[20] L. Engvall, J.A. Evans, Isogeometric triangular Bernstein–Bézier discretizations: automatic mesh generation and geometrically exact finite element analysis, Comput. Methods Appl. Mech. Engrg. 304 (2016) 378–407.
[21] S. Xia, X. Qian, Isogeometric analysis with Bézier tetrahedra, Comput. Methods Appl. Mech. Engrg. 316 (2017) 782–816.
[22] H. Speleers, C. Manni, F. Pelosi, M.L. Sampoli, Isogeometric analysis with Powell–Sabin splines for advection-diffusion-reaction problems, Comput. Methods Appl. Mech. Engrg. 221–222 (2012) 132–148.
[23] H. Speleers, P. Dierckx, S. Vandewalle, Numerical solution of partial differential equations with Powell–Sabin splines, J. Comput. Appl. Math. 189 (1) (2006) 643–659.
[24] H. Speleers, C. Manni, Optimizing domain parameterization in isogeometric analysis based on Powell–Sabin splines, J. Comput. Appl. Math. 289 (2015) 68–86.
[25] G. Giorgiani, H. Guillard, B. Nkonga, E. Serre, A stabilized Powell–Sabin finite-element method for the 2D Euler equations in supersonic regime, Comput. Methods Appl. Mech. Engrg. 340 (2018) 216–235.
[26] S. May, J. Vignollet, R.d. Borst, Powell–Sabin B-splines and unstructured standard T-splines for the solution of the Kirchhoff–Love plate theory exploiting Bézier extraction, Internat. J. Numer. Methods Engrg. 107 (3) (2016) 205–233.

[27] S. May, R. de Borst, J. Vignollet, POwell–Sabin B-splines for smeared and discrete approaches to fracture in quasi-brittle materials, Comput. Methods Appl. Mech. Engrg. 307 (2016) 193–214.

[28] H. Speleers, C. Manni, F. Pelosi, From NURBS to NURPS geometries, Comput. Methods Appl. Mech. Engrg. 255 (2013) 238–254.

[29] L. Beirão Da Veiga, T.J.R. Hughes, J. Kiendl, C. Lovadina, J. Niiranen, A. Reali, H. Speleers, A locking-free model for Reissner–Mindlin plates: analysis and isogeometric implementation via NURBS and triangular NURPS, Math. Models Methods Appl. Sci. 25 (08) (2015) 1519–1551.

[30] Y. Jia, Y. Zhang, G. Xu, X. Zhuang, T. Rabczuk, Reproducing kernel triangular B-spline-based FEM for solving PDEs, Comput. Methods Appl. Mech. Engrg. 267 (2013) 342–358.

[31] H. Speleers, Construction of normalized B-splines for a family of smooth spline spaces over Powell–Sabin triangulations, Constr. Approx. 37 (1) (2013) 41–72.

[32] J. Grošelj, A normalized representation of super splines of arbitrary degree on Powell–Sabin triangulations, BIT Numer. Math. 56 (4) (2016) 1257–1280.

[33] J. Grošelj, H. Speleers, Construction and analysis of cubic Powell–Sabin B-splines, Comput. Aided Geom. Design 57 (2017) 1–22.

[34] S. Xia, X. Wang, X. Qian, Continuity and convergence in rational triangular Bézier spline based isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 297 (C) (2015) 292–324.

[35] M. Neamtu, Delaunay configurations and multivariate splines: a generalization of a result of B.N. Delaunay, Trans. Amer. Math. Soc. 359 (7) (2007) 2993–3004.

[36] Y. Liu, Computations of Delaunay and Higher Order Triangulations, with Applications to Splines (Ph.D. thesis), University of North Carolina at Chapel Hill, 2007.

[37] T.A. Grandine, The computational cost of simplex spline functions, SIAM J. Numer. Anal. 24 (4) (1987) 887–890.

[38] T.A. Grandine, The stable evaluation of multivariate simplex splines, Math. Comp. 50 (181) (1988) 197–205.

[39] Y. Zhang, J. Cao, Z. Chen, X. Zeng, Surface reconstruction using simplex splines on feature-sensitive configurations, Comput. Aided Geom. Design 50 (2017) 14–28.

[40] C.R. Traas, Practice of bivariate quadratic simplicial splines, in: Computation of Curves and Surfaces, Springer, 1990, pp. 383–422.

[41] B. Dembart, D. Gonsor, M. Neamtu, Bivariate quadratic B-splines used as basis functions for collocation, Mathematics for Industry: Challenges and Frontiers, SIAM, Philadelphia, PA, 2005, pp. 178–198.

[42] C. De Boor, Splines as linear combinations of B-splines. A survey, in: G. G. Lorentz, C. Chui, L. L. Schumaker (Eds.), Approximation Theorry II, Academic Press, New York, 1976, pp. 1–47.

[43] M.G.J. Franssen, Evaluation of DMS-splines (Master's thesis), Eindhoven University of Technology, 1995.

[44] C.A. Micchelli, A constructive approach to Kergin interpolation in R^k: multivariate B-splines and Lagrange interpolation, Rocky Mt. J. Math. (1980) 485–497.

[45] C.A. Micchelli, Geometric methods for piecewise polynomial surfaces, in: Mathematical Aspects of Geometric Modeling, 1995, pp. 149–206.

[46] W. El Oraiby, D. Schmitt, J.-C. Spehner, Centroid triangulation from K-sets, Int. J. Comput. Geom. Appl. 21 (06) (2011) 635–659.

[47] Q. Du, V. Faber, M. Gunzburger, Centroidal voronoi tessellations: applications and algorithms, SIAM Rev. 41 (4) (1999) 637–676.

[48] R.L. Cook, Stochastic sampling in computer graphics, ACM Trans. Graph. 5 (1) (1986) 51–72.

[49] G. Farin, Curves and Surfaces for CAGD: A Practical Guide, fifth ed., Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[50] M. Yvinec, 2D triangulation, in: CGAL User and Reference Manual, 4.13 ed., CGAL Editorial Board, 2018.

[51] G.R. Cowper, Gaussian quadrature formulas for triangles, Internat. J. Numer. Methods Engrg. 7 (3) (1973) 405–408.

[52] X. Wei, Y. Zhang, L. Liu, T.J. Hughes, Truncated T-splines: fundamentals and methods, Comput. Methods Appl. Mech. Engrg. 316 (C) (2017) 349–372.

[53] S. Lloyd, Least squares quantization in PCM, IEEE Trans. Inf. Theory 28 (2) (1982) 129–137.

[54] M. Wu, B. Mourrain, A. Galligo, B. Nkonga, Hermite type spline spaces over rectangular meshes with complex topological structures, Commun. Comput. Phys. 21 (3) (2017) 835–866.

[55] M. Majeed, F. Cirak, Isogeometric analysis using manifold-based smooth basis functions, CoRR abs/1605.00423 (2016) (2016).

[56] A. Collin, G. Sangalli, T. Takacs, Analysis-suitable G^1 multi-patch parametrizations for C^1 isogeometric spaces, Comput. Aided Geom. Design 47 (2016) 93–113, SI: New Developments Geometry.

[57] T. Nguyen, J. Peters, Refinable C^1 spline elements for irregular quad layout, Comput. Aided Geom. Design 43 (2016) 123–130.

[58] D. Toshniwal, H. Speleers, T.J. Hughes, Smooth cubic spline spaces on unstructured quadrilateral meshes with particular emphasis on extraordinary points: geometric design and isogeometric analysis considerations, Comput. Methods Appl. Mech. Engrg. 327 (2017) 411–458.

[59] X. Wei, Y.J. Zhang, D. Toshniwal, H. Speleers, X. Li, C. Manni, J.A. Evans, T.J. Hughes, Blended B-spline construction on unstructured quadrilateral and hexahedral meshes with optimal convergence rates in isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 341 (2018) 609–639.