SCIENCE CHINA
Mathematics

Progress of Projects Supported by NSFC
• REVIEWS •

April 2016 Vol. 59 No. 4: 617–644
doi: 10.1007/s11425-015-5063-8

A survey on the local refinable splines

LI Xin*, CHEN FaLai, KANG HongMei & DENG JianSong

School of Mathematical Sciences, University of Science and Technology of China, Hefei 230026*, China*
Email: lixustc@ustc.edu.cn, chenfl@ustc.edu.cn, khm@mail.ustc.edu.cn, dengjs@ustc.edu.cn

Received September 11, 2014; accepted May 15, 2015; published online December 30, 2015

Abstract This paper provides a survey of local refinable splines, including hierarchical B-splines, T-splines, polynomial splines over T-meshes, etc., with a view to applications in geometric modeling and iso-geometric analysis. We will identify the strengths and weaknesses of these methods and also offer suggestions for their using in geometric modeling and iso-geometric analysis.

Keywords geometric modeling, isogeometric analysis, B-splines, hierarchical B-splines, T-splines, polynomial splines over hierarchical T-meshes (PHT-splines), locally refined (LR) B-splines, local refinement

MSC(2010) 65D07, 47A15

Citation: Li X, Chen F L, Kang H M, et al. A survey on the local refinable splines. Sci China Math, 2016, 59: 617–644, doi: 10.1007/s11425-015-5063-8

1 Introduction

Geometric modeling is concerned with the mathematical representation of shapes on a computer, which has been used in computer-aided design, engineering and manufacturing, as well as in computer graphics and animation. Among all the technologies, non-uniform rational B-splines (NURBS) provides an intuitive, easy-to-use, tractable scheme for creating mathematically well-defined freeform curves and surfaces, which become current international de facto standard representation. However, a significant disadvantage of NURBS is that they are based on a tensor-product structure, which means that NURBS models may have a large number of superfluous control points only to satisfy the topological requirement. What is more, in traditional NURBS-based design, modeling a complicated engineering design often requires hundreds, if not thousands, of NURBS patches which are usually discontinuous across patch boundaries. The complexity of the patch layout coupled with the manual enforcement of smoothness across patch boundaries (via control point positioning) makes NURBS design of complicated geometries time-consuming, error prone, and tedious.

After creating the geometric models embodied in computer aided design (CAD) systems, the design description needs to be translated to an analysis-suitable geometry for mesh generation and use in a finite element analysis (FEA) code in order to simulate the physical behavior. This task is far from trivial. For complex engineering designs it is now estimated to take over 80% of overall analysis time. One vision to solve this integration problem of CAD and FEA is to breakdown the barriers between engineering design and analysis and to reconstitute the entire process which focus on one, and only one, geometric model. This new vision is formalized into a new notion, *iso-geometric analysis* (IGA), developed in [35] and described in detail in [12]. With IGA, traditional design-through-analysis procedures such as

*Corresponding author

geometry clean-up, defeaturing, and mesh generation are simplified or eliminated entirely. Additionally, the higher-order smoothness provides substantial gains to analysis in terms of accuracy and robustness of finite element solutions [13, 21, 54]. However, a global geometric discretization, based on NURBS, is usually not suitable as a basis for analysis.

With the needs both from geometric modeling and iso-geometric analysis, a very natural question arises that how to define a spline space such that we can adaptively refine the spline space in the localized changes of the control net. A lot of researchers have tackled this issue and many different methods have been developed in these years, including hierarchical B-splines [26, 87], T-splines [53, 70, 72], polynomial splines over T-meshes [14, 16], and LR B-splines [17], etc. In this paper, we provide a survey of all these local refinable splines with a view to applications in geometric modeling and iso-geometric analysis. We will identify the strengths and weaknesses of these methods and also offer suggestions for their using in geometric modeling and iso-geometric analysis.

The paper is structured as follows. Pertinent background on NURBS and IGA is reviewed in Section 2. In Sections 3–6, we introduce the basic concepts for hierarchical B-splines, T-splines, polynomial splines over T-meshes and LR B-splines, respectively. The last section is the summary and conclusion.

2 NURBS and iso-geometric analysis

In this section, we briefly introduce bivariate tensor-product B-splines space [22, 23] and iso-geometric analysis [12, 35] with the aim of recalling a few results that we will require in the next several sections.

2.1 Tensor-produce B-spline spaces

There are two separate ways to define a bivariate tensor-product B-spline space. Given two polynomial degrees $(d_1; d_2)$ and the horizontal and vertical knots vectors

$$U = \{u_0 \leqslant u_1 \leqslant \cdots \leqslant u_{n+d_1+1}\}, \quad V = \{v_0 \leqslant v_1 \leqslant \cdots \leqslant v_{m+d_2+1}\},$$

a bivariate tensor-product B-spline space $\mathbb{B}$ can be defined as a linear space spanned by a set of tensor-product B-spline functions $N_{i,j}(s,t) = N_{i,U}^{d_1}(s) N_{j,V}^{d_2}(t)$. Here $N_{i,U}^{d_1}(s)$ and $N_{j,V}^{d_2}(t)$ are univariate B-spline functions defined by U, d_1 and V, d_2 as follows:

$$N_{i,U}^0(s) = \begin{cases} 1, & u_i \leqslant s < u_{i+1}, \\ 0, & \text{otherwise}, \end{cases}$$

$$N_{i,U}^p(s) = \frac{s - u_i}{u_{i+p} - u_i} N_{i,U}^{p-1}(s) + \frac{u_{i+p+1} - s}{u_{i+p+1} - u_{i+1}} N_{i+1,U}^{p-1}(s).$$

The B-spline basis can be generalized to the rational case by associating a positive weight $\omega_{i,j}$ to each control point, which gives the NURBS basis functions $R_{i,j}(s,t) = \frac{\omega_{i,j} N_{i,j}(s,t)}{\sum_{i=0}^{n} \sum_{j=0}^{m} \omega_{i,j} N_{i,j}(s,t)}$, then any function $f(s,t)$ is represented as $f(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{m} d_{i,j} R_{i,j}(s,t)$. Here $d_{i,j}$ form the control net associated to the parametric representation.

The bivariate tensor-product B-spline space $\mathbb{B}$ can be considered in another way. Rewrite the two knot vectors U and V by getting rid of the multiplicities as

$$\{u_{i_0} < u_{i_1} < \cdots < u_{i_p}\}, \quad \{v_{j_0} < v_{j_1} < \cdots < v_{j_q}\},$$

where $u_{i_k} = u_{i_k+1} = \cdots = u_{i_{k+1}-1}$ and $v_{j_k} = v_{j_k+1} = \cdots = v_{j_{k+1}-1}$. Denote $\mu_k = i_{k+1} - i_k$ and $\nu_k = j_{k+1} - j_k$ to be the multiplicities of knots u_{i_k} and v_{j_k}. With these notations, the spline space $\mathbb{B}$ can also be defined in terms of the smoothness and polynomial orders,

$$\mathbb{B} := \{f(s,t) \mid f(s,t)|_{[u_{i_k}, u_{i_{k+1}}] \times [v_{j_l}, v_{j_{l+1}}]} \in \mathbb{P}_{mn},\ f \text{ is } C^{d_1 - \mu_k} \text{ at } s = u_{i_k} \text{ and is } C^{d_2 - \nu_l} \text{ at } t = v_{j_l}\},$$

where $\mathbb{P}_{mn}$ is the space of all the polynomials with bi-degree (m, n).

2.2 Iso-geometric analysis

Given a Lipschitz continuous domain Ω, which is designed in a NURBS-based CAD program with the form $G : \Omega_0 := [0,1]^2 \mapsto \Omega$ as illustrated in Figure 1,

$$G(s,t) := \sum_{i=0}^{n}\sum_{j=0}^{m} R_{i,j}(s,t)P_{i,j}. \tag{2.1}$$

Let L be an operator on Ω with boundary $\Gamma = \Gamma_D \cup \Gamma_N$. We want to solve the partial differential equation

$$Lu = f \quad \text{on} \quad \Omega \tag{2.2}$$

with the boundary conditions

$$u = 0 \quad \text{on} \quad \Gamma_D \quad \text{and} \quad \langle \nabla u, n\rangle = h \quad \text{on} \quad \Gamma_N$$

for unknown $u : \Omega \to \mathbb{R}$ with given data f and the outer normal vector n on Γ_N. Without loss of generality, we assume the zero Dirichlet boundary conditions on Γ_D.

The weak form of (2.2) takes the standard form: find $u \in V$ such that

$$a(u,v) = l(v) \quad \text{for all} \quad v \in V, \tag{2.3}$$

where the appropriate function space is given by

$$V := \{u \in H^1(\Omega), u|_{\Gamma_D} = 0\}.$$

As usual, $a : V \times V \to \mathbb{R}$ denotes the symmetric bilinear form that corresponds to the operator L and $l : V \to \mathbb{R}$ is a linear functional that contains the right-hand side term f and the Neumann term h. The bilinear form a is assumed to be continuous and coercive such that a unique solution to (2.3) exists.

The IGA-based Galerkin projection is the compositions of the shape functions with the inverse of the geometry function, $\phi_{(m+1)i+j} = R_{i,j} \circ G^{-1}$. With the standard Galerkin finite element approach, the approximation solution $u_h = \sum_{i=1}^{(m+1)(n+1)} q_i\phi_i$ has to satisfy a linear system $Aq = b$, where the matrix $A_{ij} = a(\phi_i, \phi_j)$ and the right term $b_i = l(\phi_i)$ for $i, j = 1, 2, \ldots, (m+1)(n+1)$, respectively.

We should emphasize that both geometric modeling and iso-geometric analysis based on NURBS offer the possibility for *knot insertion* (*h-refinement*) by inserting new knots in the knot vectors and for *degree elevation* (*p-refinement*) by increasing the degree of the NURBS. Additionally, in IGA, the so-called *k-refinement* is available that combines the *h*- and *p-refinement* strategy [11, 21]. However, all the three refinements are global in nature, and it is clear that there is a need for more general basis functions that feature local subdivision (the survey only focuses on the problem of h-refinement) while still maintaining the favorable properties of NURBS.

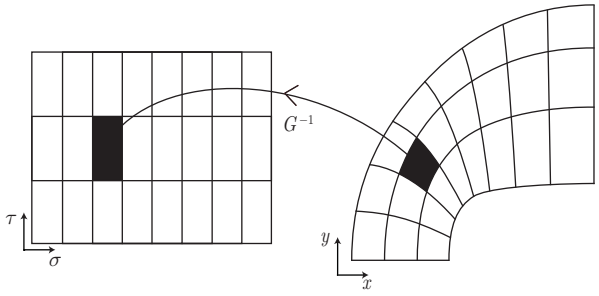


Figure 1 The iso-geometric analysis mapping

3 Hierarchical *B*-splines

The concept of hierarchical refinement of B-splines was introduced by Forsey and Bartels for local surface refinement in CAD [26]. However, it does not have a well development for applications in CAD. But in the framework of iso-geometric analysis, hierarchical refinement of NURBS has recently attracted increasing attention due to the following important advantages. First, hierarchical B-splines rely on the principle of B-spline subdivision, which makes it easy to maintain linear independence and the maximal smoothness throughout the refinement process. Second, hierarchical B-splines rely on a local tensor product structure, and they can be easily generalized to arbitrary dimensions. The rigidity and simplicity of the tensor product structure also facilitate automation of the refinement process. Third, very similar refinement techniques based on a hierarchical split of standard finite element bases have existed in the finite element analysis community for a long time and the tree-like data structure for hierarchical B-splines is also well-established in computer science. These existing technologies can help one to become familiar with hierarchical B-spline refinement. Despite being appealing for analysis purposes, it is unclear whether hierarchical B-splines will be adopted in a commercial CAD environment because they do not have a natural control grid.

A hierarchical B-splines surface can be locally refined using overlays. Based on such hierarchical model, complex surfaces can be created from simple NURBS surfaces with hierarchical editing. Later researches mainly focus on how to construct basis functions of hierarchical B-spline spaces. The first specific basis selection mechanism was proposed by Kraft [41], and extended in [87] to satisfy some nice properties, such as non-negativity, linearly independence and local support. Later, Vuong et al. [87] normalized the hierarchical B-splines proposed by reducing the support of basis functions defined on coarse grids, according to finer levels in the hierarchy of splines [29]. They called such hierarchical B-splines as truncated hierarchical B-splines (THB-splines for short).

Surface reconstruction schemes for solving interpolation and approximation problems by using multi-level B-splines were originally discussed by Forsey and Bartels [27], and additionally investigated in [30]. A quasi-interpolation algorithm based on hierarchical B-splines was developed in [41]. There is an increasing interest in hierarchical splines coming from recent studies related to iso-geometric analysis [87], where a set of linearly independent weighted basis functions are constructed to define the geometry and are used in analysis. In order to have a self-contained presentation, next, we briefly summarize from [29, 40, 87] the construction of hierarchical B-splines and THB-splines. Generalization of hierachical B-splines over regular triangular partitions is discussed in [39].

Let $\mathcal{S}^k, k = 0, \ldots, N$ be a nested sequence of tensor-product B-spline function spaces defined on the domain Ω_0,

$$\mathcal{S}^0 \subseteq \mathcal{S}^1 \subseteq \cdots \subseteq \mathcal{S}^N.$$

Each spline space $\mathcal{S}^k, k = 0, 1, \ldots, N$ is spanned by a given tensor-product B-spline basis $\mathcal{B}^k$ defined on two knot sequences U^k, V^k, containing the horizontal and vertical knots, respectively. These knot sequences are also nested, namely,

$$U^0 \subseteq U^1 \subseteq \cdots \subseteq U^N, \quad V^0 \subseteq V^1 \subseteq \cdots \subseteq V^N.$$

The degrees of $\mathcal{B}^k$ is (p^k, q^k), and $p^{k+1} \geqslant p^k, q^{k+1} \geqslant q^k, k = 0, \ldots, N-1$. In order to obtain nested spaces, it is assumed that

$$\mu(U^{k+1}, u) - \mu(U^k, u) \geqslant p^{k+1} - p^k, \quad \mu(V^{k+1}, v) - \mu(V^k, v) \geqslant q^{k+1} - q^k,$$

for all $u \in U, v \in V$ with $k = 0, \ldots, N-1$. Here, $\mu(U, u)$ is the multiplicity of the parameter value u in the knot vector U (the multiplicity $\mu(U, u)$ is zero if the given value u is not a knot in U). These conditions are necessary and sufficient.

In addition, let

$$\Omega_0 \supseteq \Omega_1 \cdots \supseteq \Omega_N, \quad \Omega_{N+1} = \emptyset$$

be a sequence of nested domains. Each $\Omega_k \in \mathbb{R}^2, k = 0, \ldots, N$ represents the region selected to be refined at level k and its boundary $\partial\Omega_k$ is may be aligned with the knot lines of $\mathcal{S}^{k-1}$(strong condition) or $\mathcal{S}^k$ (weak condition).

Finally, the support of a function f is defined as

$$\text{supp } f = \{\boldsymbol{x} : f(\boldsymbol{x}) \neq 0 \wedge \boldsymbol{x} \in \Omega_0\}.$$

Definition 3.1. The hierarchical B-spline basis $\mathcal{H}$ is recursively constructed as follows (see [87]):

1. Initialization: $\mathcal{H}^0 = \{\beta \in \mathcal{B}^0 : \text{supp } \beta \neq \emptyset\}$.
2. Recursive construction: $\mathcal{H}^{l+1} = \mathcal{H}_A^{l+1} \cup \mathcal{H}_B^{l+1}$, for $l = 0, \ldots, N-1$, where

$$\mathcal{H}_A^{l+1} = \{\beta \in \mathcal{H}^l : \text{supp } \beta \nsubseteq \Omega_{l+1}\},$$

and

$$\mathcal{H}_B^{l+1} = \{\beta \in \mathcal{B}^{l+1} : \text{supp } \beta \subseteq \Omega_{l+1}\}.$$

3. $\mathcal{H} = \mathcal{H}^N$.

THB-splines improve hierarchical B-splines in two aspects: Normalizing hierarchical B-splines and making the support of a basis function smaller. It is based on the following representation of a basis $\tau \in \mathcal{B}^l$,

$$\tau = \sum_{\beta \in \mathcal{B}^{l+1}} c_\beta^{l+1}(\tau)\beta, \quad c_\beta^{l+1} \in \mathbb{R}. \tag{3.1}$$

Then the truncation of τ with respect to $\mathcal{B}^{l+1}$ and Ω_{l+1} is defined as

$$\text{trunc}^{l+1}\tau = \sum_{\beta \in \mathcal{B}^{l+1}, \text{supp } \beta \nsubseteq \Omega_{l+1}} c_\beta^{l+1}(\tau)\beta. \tag{3.2}$$

Definition 3.2. The hierarchical B-spline basis $\mathcal{T}$ is recursively constructed as follows (see [29]):

1. Initialization: $\mathcal{T}^0 = \mathcal{H}^0$.
2. Recursive construction: $\mathcal{T}^{l+1} = \mathcal{T}_A^{l+1} \cup \mathcal{T}_B^{l+1}$, for $l = 0, \ldots, N-1$, where

$$\mathcal{T}_A^{l+1} = \{\text{trunc}^{l+1}\beta : \beta \in \mathcal{T}^l \wedge \text{supp } \beta \nsubseteq \Omega_{l+1}\},$$

and

$$\mathcal{T}_B^{l+1} = \mathcal{H}_B^{l+1}.$$

3. $\mathcal{T} = \mathcal{T}^N$.

In order to understand the difference between THB-splines and hierarchical B-splines more easily, Figure 2 shows how to define univariate HB-splines and THB-splines on a nested intervals.

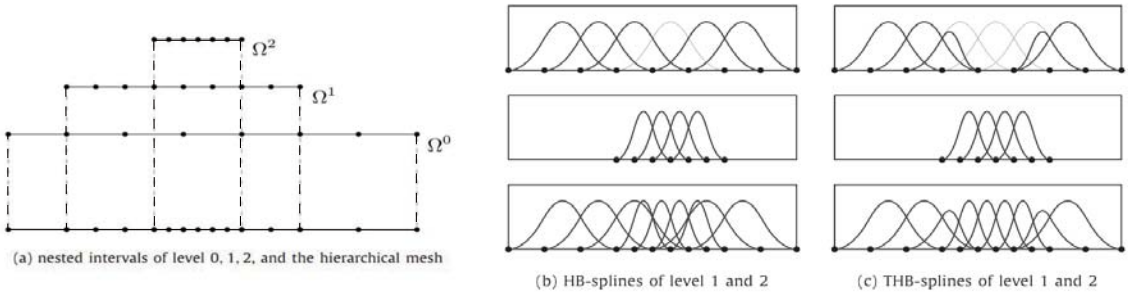


Figure 2 Univariate quadratic HB- and THB-splines defined on the hierarchical mesh shown in (a). For Cases (b) and (c), top: Basis functions of level 1 (B-splines of level 1 that are replaced or modified are depicted in grey lines); middle: Finer basis functions of level 2; bottom: Combination of basis functions from these two hierarchical levels

4 *T*-splines

T-splines are introduced in [70,72] in the computer aided geometric design (CAGD) community to address the important limitations of NURBS. T-splines can model complicated engineering designs as a single, watertight geometry. Additionally, NURBS are a special case of T-splines, so existing technology based on NURBS extends to T-splines. Any trimmed NURBS model can be represented by a watertight trimless T-spline [71] and multiple NURBS patches can be merged into a single watertight T-spline [36,72]. Unlike NURBS, T-splines can be locally refined [68, 70]. These geometric properties are especially critical in the context of geometric modeling and iso-geometric analysis where the behavior and accuracy of the method are strongly influenced by the watertightness, smoothness, and the ability to refine the surface mesh in a localized region. Thus, T-splines are regarded as a technology both for design and analysis.

4.1 Definition

The concept of an arbitrary degree T-spline was defined in [3, 25, 70, 72] based on a T-mesh in the index domain which is referred as an index T-mesh. A T-mesh $\boldsymbol{T}$ for a bi-degree (d_1, d_2) T-spline is a collection of all the elements of a rectangular partition of the index domain $[0, c+d_1]\times[0, r+d_2]$, where all rectangle corners (or vertices) have integer coordinates. Each vertex has a unique pair of index coordinates (δ_i, τ_i). The valence of a vertex is the number of edges such that the vertex is an endpoint. The interior vertices allow valence three (called T-junctions) or four vertices. Notation $\vdash$, $\dashv$, $\perp$ and $\top$ indicate the four possible orientations for the T-junctions. The *active region* is the rectangle region $[p, c+d_1-p]\times[q, r+d_2-q]$, here p and q are the maximal integers equal or less than $\frac{d_1+1}{2}$ and $\frac{d_2+1}{2}$, respectively. The active region carries the anchors that will be associated with the blending functions while the other indices will be needed for the definition of the blending function when the anchor is close to the boundary. For example, Figure 3 is some example T-meshes, where active regions are marked with grey.

An *anchor* is a point in an index T-mesh which corresponds to one blending function. If both d_1 and d_2 are odd, the anchor is the vertex in the active region of the T-mesh, if both d_1 and d_2 are even, then the anchor is the barycenter of each face in the active region of the T-mesh. If d_1 is even and d_2 is odd, the anchor is the middle point of each horizonal edge in the active region of the T-mesh and if d_1 is odd and d_2 is even, then the anchor is the middle point of each vertical edge in the active region of the T-mesh.

For the i-th anchor, a *local index vector* $\vec{\delta_i}\times\vec{\tau_i}$ is used to define the blending function $T_i(s,t)$. The values of $\vec{\delta_i}=[\delta_i^0,\ldots,\delta_i^{d_1+1}]$ and $\vec{\tau_i}=[\tau_i^0,\ldots,\tau_i^{d_2+1}]$ are determined as follows. From the i-th anchor in the *index T-mesh*, a ray is shot in the s and t direction traversing the T-mesh to get a total of d_1+2 and d_2+2 knot indices, which form $\vec{\delta_i}$ and $\vec{\tau_i}$, as shown in Figure 3.

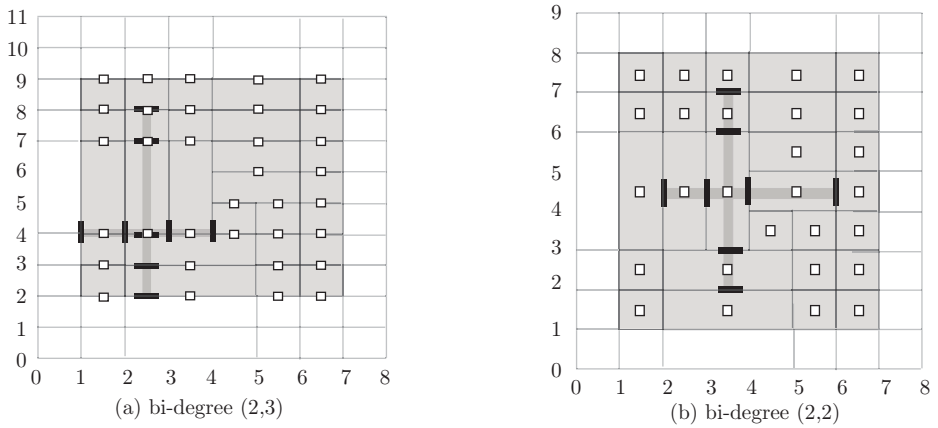


Figure 3 Example T-meshes and the local index vectors for two blending functions

For the given T-mesh, the knot vector indices correspond to two global knot vectors $\vec{\boldsymbol{s}} = [s_0, s_1, \ldots, s_{c+d_1}]$ and $\vec{\boldsymbol{t}} = [t_0, t_1, \ldots, t_{r+d_2}]$. The end condition knots for $\vec{\boldsymbol{s}}$ and $\vec{\boldsymbol{t}}$ may have multiplicity $d_1 + 1$ and $d_2 + 1$; all other knots are multiplicity $\leqslant d_1$ and $\leqslant d_2$, respectively.

The equation of a T-spline surface is defined as

$$\mathcal{T}(s,t) = \sum_{i=1}^{n_A} \mathcal{T}_i T_i(s,t), \tag{4.1}$$

where $\mathcal{T}_i = (\omega_i x_i, \omega_i y_i, \omega_i z_i, \omega_i) \in \mathbb{P}^3$ are homogeneous control points, $\omega_i \in \mathbb{R}$ are weights, and n_A is the number of control points or anchors. $T_i(s,t)$ is a tensor-product of degree d_1 and d_2 B-spline basis functions $T_i(s,t) = N[\vec{\boldsymbol{s}}_i](s)N[\vec{\boldsymbol{t}}_i](t)$, where

$$\vec{\boldsymbol{s}}_i = [s_{\sigma_i^0}, s_{\sigma_i^1}, \ldots, s_{\sigma_i^{d_1+1}}], \quad \vec{\boldsymbol{t}}_i = [t_{\tau_i^0}, t_{\tau_i^1}, \ldots, t_{\tau_i^{d_2+1}}]$$

are subsequences of $\vec{\boldsymbol{s}}$ and $\vec{\boldsymbol{t}}$, respectively.

4.2 Arbitrary topology *T*-splines

T-splines can be generalized to arbitrary topology, i.e., for which extraordinary points are allowed in the control grid, see Figure 4 for an example T-mesh. An extraordinary point is a vertex whose valence is not four, and which is not a T-junction. However, with extraordinary points, the rules for the definition of blending functions are not suitable for some vertices. There are two ways to generalize T-splines to arbitrary topology, the rules based on subdivision surfaces [72, 73] and patch-based methods [69]. Both methods can only handle bi-cubic T-splines as a special cases and only the surface patches associated with the extraordinary faces (a face which at least has one extraordinary point as the vertex) are different. The subdivision-based approach generates elements near the extraordinary point which are comprised of an infinite sequence of piecewise polynomials. Thus, they are not backward compatible with NURBS which can be avoided by patch-based methods. Figure 5 illustrates a T-spline models from Rhinoers T-splines plugin using the patch-based approach.

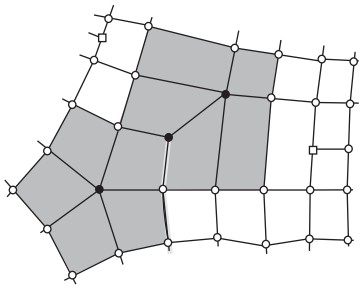


Figure 4 An arbitrary topological T-mesh, black circle are extraordinary points

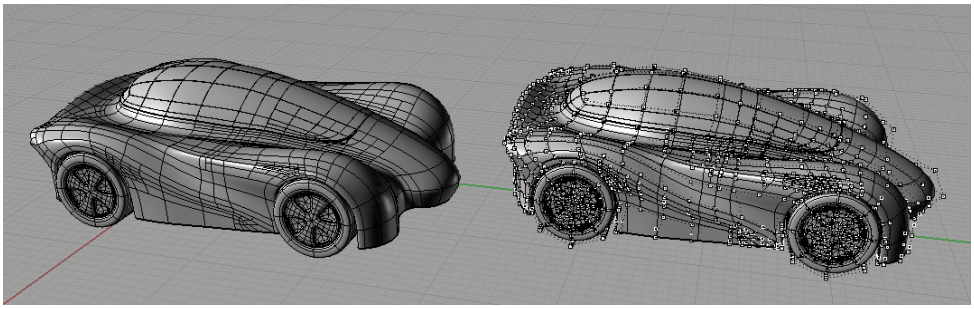
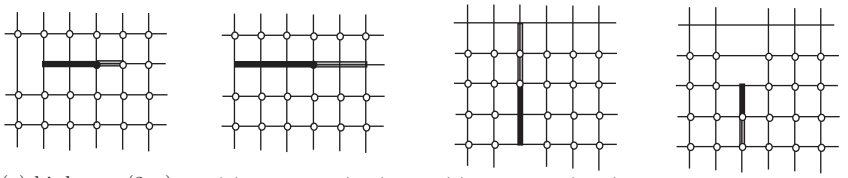


Figure 5 An arbitrary topological T-spline surface



(a) bi-degree $(3,*)$ (b) bi-degree $(5,*)$ (c) bi-degree $(*,4)$ (d) bi-degree $(*,2)$

Figure 6 The extension for four different kinds of T-junctions

4.3 Analysis-suitable *T*-splines

Analysis-suitable T-splines form a practically useful subset of T-splines, which maintain the important mathematical properties of the NURBS basis functions while providing an efficient and highly localized refinement capability. Analysis-suitable T-splines are defined in terms of T-junction extensions. For example, for an i-th T-junction (δ_i, τ_i) of type $\vdash$ or $\dashv$, the extension for the T-junction, denoted as $\mathrm{ext}(T_i)$ is line segment $[\underline{i}, \overline{i}] \times \{\tau_i\}$. $\underline{i}$ and $\overline{i}$ are determined such that edges $[\underline{i}, \delta_i) \times \{\tau_i\}$ intersect the T-mesh $[\frac{d_1+1}{2}]$ times and edges $(\delta_i, \overline{i}] \times \{\tau_i\}$ intersect the T-mesh $[\frac{d_1}{2}]$ times for T-junction of type $\vdash$. For T-junction of type $\dashv$, one can similarly define the extension except the number of intersections are exchanged. Also, the extensions for the other kinds of T-junctions $\perp$, $\top$ can be defined while using degree d_2 instead of d_1. All these extension examples are illustrated in Figure 6.

Definition 4.1. For a bi-degree (d_1, d_2) T-spline, a T-mesh is called analysis-suitable (for short, AS) T-mesh if the extensions for all the T-junctions $\vdash$ and $\dashv$, do not intersect the extensions for all the T-junctions $\perp$ and $\top$. A T-spline defined on an analysis-suitable T-mesh is called analysis-suitable T-spline, for short AS T-spline.

Several important properties of AS T-splines have been proven as follows:

- The blending functions are locally linearly independent for *any* choice of knots [53].
- The basis constitutes a partition of unity [108].
- Each basis function is non-negative.
- They can be generalized to arbitrary degree [81, 108].
- An affine transformation of an analysis-suitable T-spline is obtained by applying the transformation to the control points. We refer to this property as affine invariance. This implies that all "patch tests" are satisfied *a priori*.
- They obey the convex hull property.
- They can be locally refined [68, 70, 72].
- A dual basis can be constructed [80, 81].
- The AS T-spline space can be characterize in terms of piecewise polynomial [52].

4.4 Modified *T*-splines

Modified T-splines [38] are a variant of T-splines. For a given T-mesh $\boldsymbol{T}$, a set of non-negative, linearly independent basis functions are constructed for each anchor of the T-mesh, which are called Modified T-splines basis functions. The construction consists of the following two major steps (see Figure 7).

In the first step, an extended T-mesh $\boldsymbol{T}'$ is constructed by extending T-junctions of $\boldsymbol{T}$ such that the T-spline blending functions $T_i(s,t)$ over $\boldsymbol{T}'$ are linearly independent and form a partition of unity. For example, Figure 7(a) is an example T-mesh and Figure 7(b) is one possible extended T-mesh $\boldsymbol{T}'$ and the corresponding local knot vectors for some basis functions are shown in Figure 7(c). In the second step, we distribute the basis functions $T_i(s,t)$ over $\boldsymbol{T}'$ to those for T-mesh $\boldsymbol{T}$. Let $\{v_i'\}_{i=1}^{l}$ be the new vertices generated by the T-vertex extensions. We distribute the basis function $T_i(s,t)$ of v_i' to the basis functions at the neighboring old vertices of $\boldsymbol{T}'$. In Figure 7(d), the basis function of v_1' is distributed to the basis functions of v_2 and v_5, and the basis function of v_2' is distributed to the basis functions of v_2 and v_6.

Finally, the basis function over $\boldsymbol{T}$ is constructed as a linear combination of the basis functions $T_i(s,t)$ at the neighboring vertices of v_i. Figure 7(e) illustrates the basis function of v_2.

4.5 Control point insertion algorithm

Local control points insertion algorithm is the key algorithm for T-splines. The algorithm in [70] is a recursive procedure to refine all the possible influence blending functions. An important key to understand the algorithm is that the blending functions and the T-mesh anchors are tightly coupled. Every anchor corresponds to a blending function, and each blending function's knot vectors are defined from the T-mesh. For the given T-spline and some new insertion control points, the algorithm outputs a new T-spline that contains all the insertion control points and keeps the geometry unchanged by the following steps:

1. Modify the topology for the T-mesh by insertion all the new control points.

2. Add all the blending functions into a list L. If L is not empty, select a blending function from L and repeat the following steps:

(a) If the blending function is coupled with current T-mesh, continue.

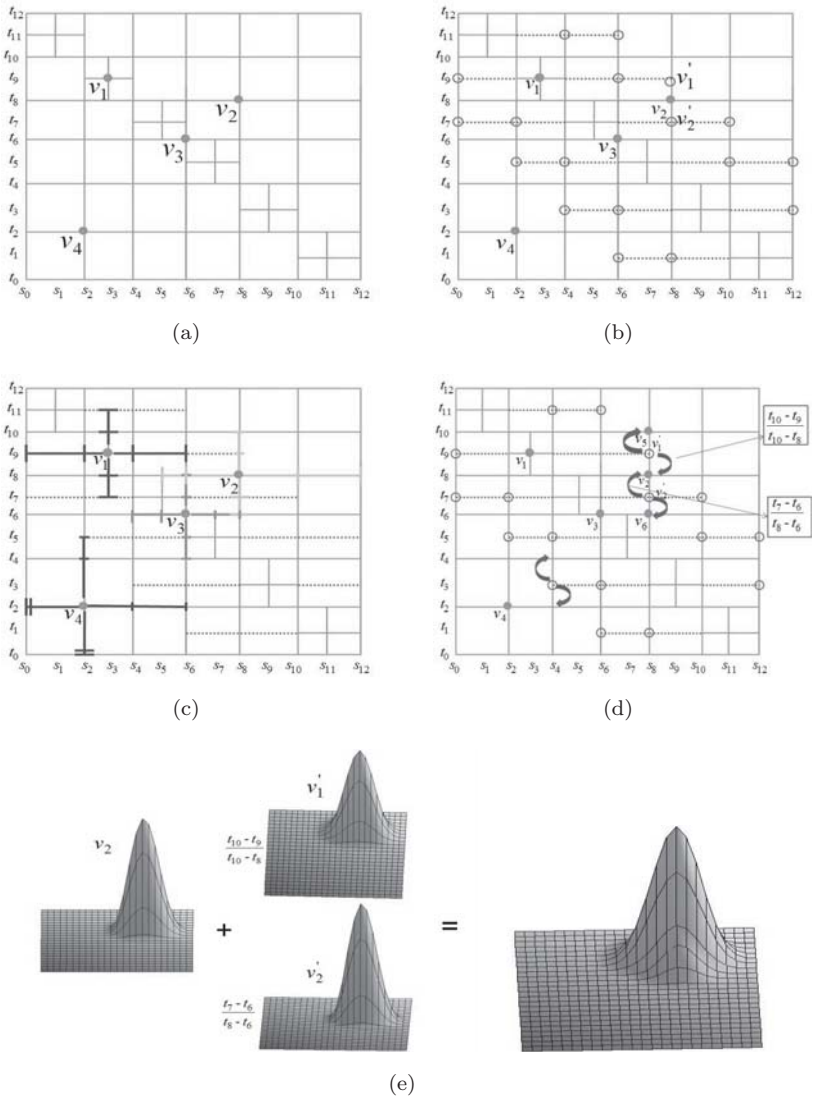


Figure 7 Modified T-splines Construction. (a) Origin T-mesh T. (b) Extended T-mesh T'. Circles are new vertices. (c) Knot vectors of basis functions over T'. (d) Basis functions at new vertices are distributed. (e) Basis function at v_2 is constructed

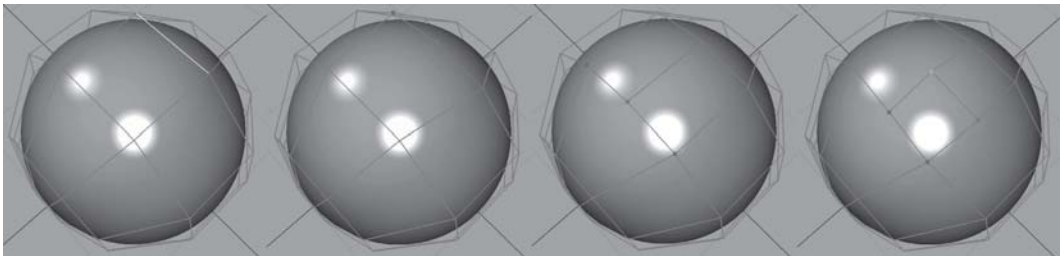


Figure 8 Local refinement for a T-spline. From left to right, the initial T-spline surface, insert one control point in the middle of the edge, insert one new control point in the second T-spline surface and insert one control point in the third T-spline surface with one additional insertion

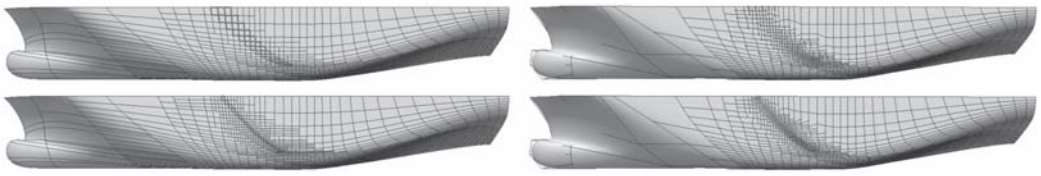


Figure 9 Analysis-suitable T-spline local refinement. The left ones are the T-spline surfaces before refinement, the dark regions are the elements which will be refined in the next iteration. The final T-spline surfaces are shown on the right

(b) If the blending function has missing knots, refine the blending function into two new blending functions and add the two blending functions into the list L.

(c) If the blending function has a knot that is not dictated, add an appropriate control point associated with the knot into the T-mesh.

3. Compute the new positions for the control points.

Figure 8 shows a simple local refinement example. Given a sphere represented in a T-spline in the first figure, if one inserts one control point in the middle of the marked edge, the algorithm will output the second figure. Similarly, one can insert one more control point into the new T-spline surface and create the third figure. The last one is the result of inserting one control point but with one more additional control point by the algorithm in order to keep the geometry unchanged.

Scott et al. [68] developed a highly localized refinement optimized algorithm to make the T-spline after knot insertion an analysis suitable T-spline, which meets the demands of both design and analysis. The basic steps of the algorithm are described as follows, for more details, please refer to [68].

1. Modify the topology for the T-mesh by insertion all the new control points.

2. Convert the new T-mesh into an analysis-suitable T-mesh by minimizing the number of new insertion control points using a greedy strategy.

3. Compute the positions for the control points.

The following real-world example of a ship hull design is used to show the results of refinement. Figure 9 shows two iterations of local refinements for the T-spline ship hull design. A set of elements are selected for refinement as shown on the left of Figure 9. These T-mesh elements are subdivided and the local refinement algorithm is applied to the resulting subdivided T-mesh. The control points added during local refinement are shown on the right of Figure 9. Notice that these control points remain localized to the region of selected elements.

4.6 Applications

T-splines have been applied into several important geometric modeling technologies, which have been released into a commercial software for Rhinoers [78]. In this section, we summary some applications of T-splines in geometric modeling and iso-geometric analysis.

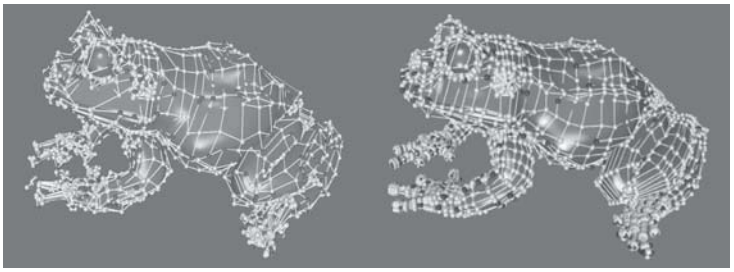


Figure 10 T-spline construction via NURBS simplification (reprinted with permission from [10])

4.6.1 *Simplification from NURBS*

Simplification is the process of modifying an existing geometric description of an object to an equivalent or approximately equivalent form that is less complex, i.e., that has fewer control points. Motivated by the new possibilities available in T-splines, there are two methods for simplifying T-spline surfaces [10,70]. The first approach is based on *iterative refinement*. The key initial step is to generate an over-simplified version of the model (for example, a single Bézier patch) to be simplified and then to refine it until a desired approximation to the original is achieved. The second approach is based on *iterative simplification*. The approach tries to remove the unnecessary control points under the given tolerances. Figure 10 illustrates a simplification example for the frog model, which has 11625 control points. The left figure is the result of iterative simplification and the right one is the result of iterative refinement for the tolerance 1.5%. Iterative simplification solution has 3,975 control points and iterative refinement solution has 5,035 control points.

4.6.2 *Merging Multiple NURBSes into a single T-spline surface*

Another way to produce a T-spline surface is to merge a set of NURBS surfaces. In geometric modeling, portions of objects are modeled separately. This creates a problem that result pieces do not always fit precisely. So, it is very important to combine them into one single surface. This process is called merging or stitching [36,72].

Merging a set of B-splines into a single B-spline surface requires that any two adjacent surfaces must have the same common knot vector along the common boundary lines. Hence knot insertion must first be performed before stitching. However, in a tensor-product spline surface, these knot insertions can significantly increase the number of control points. But merging with T-splines is different which only needs to modify a narrow band of the surfaces along their common boundary curves. The main steps of the merging algorithm are as follows. The first step is to determine the common parameter domains. Then the surface should be reparameterized such that the parametrization of the two surfaces coincides along the common boundary curve. And then some control points are inserted into T-meshes to combine the vertices and edges for common boundary curves. The last step is to merge the end conditions. Figure 11 illustrates the behavior of merging with T-splines. The left figure is the face model represented with 18 different NURBS surfaces. And the right one is a single T-spline model after merging.

4.6.3 *Convert trimming NURBS into a single T-spline surface*

The trimmed-NURBS modeling paradigm suffers from a serious fundamental flaw: parametric trimming curves are mathematically incapable of fulfilling their primary role, which is to represent the curve of intersection between two NURBS surfaces. Sederberg et al. [71] proposed a two-steps method to convert trimmed NURBS surfaces into a single watertight un-trimmed T-spline surface. Besides watertightness, the result T-spline model is totally editable, directly fillet, simple crease generation, see Figure 12 for an example.

4.6.4 *Fitting a mesh model with T-splines*

High-order and regularly sampled surface representations are more efficient and compact than general meshes and considerably simplify many geometric modeling and processing algorithms. A number of

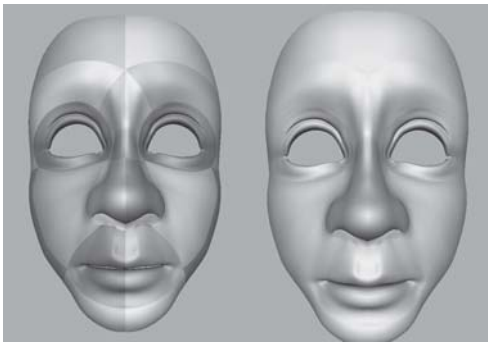


Figure 11 Merging a face model using T-splines (reprinted with permission from [36])

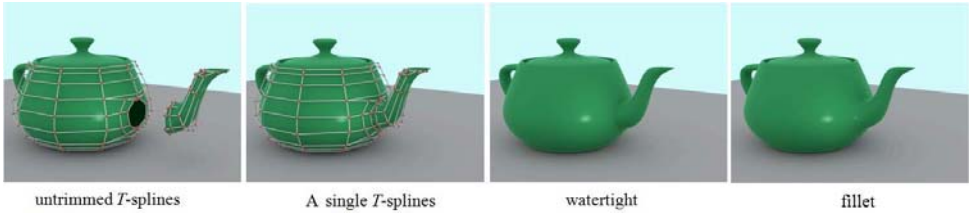


Figure 12 Convert trimmed NURBS into a single T-spline, which can be edited and filleted naturally

recent algorithms convert arbitrary meshes to regularly T-spline form. Zheng et al. [114] considered an adaptive T-spline fitting algorithm for function data. Later, they generalized the algorithm to handle triangle mesh of simple topology [94, 95, 113]. Nasri et al. [57], and Yang and Zheng [106] approximated a set of lofting curves with T-splines. Li et al. [45] demonstrated that how to use the periodic global parametrization (PGP) to fit T-spline surfaces from triangle meshes of arbitrary topology. An important feature of PGP is the ability that can introduce T-junctions during the parametrization process. However, the complexity of the resulting domain mesh is still determined by the topological structure of the field, with significant smoothing required to make it simpler. Thus, Myles et al. [55] proposed an approach to construct patch layouts consisting of small numbers of quadrilateral patches while maintaining good feature alignment. He et al. [31], and Wang et al. [88] presented a new and effective method to construct manifold T-splines of complicated topology/geometry. The fundamental idea of the approach is the geometry-aware object segmentation, by which an arbitrarily complicated surface model can be segmented into several simple charts and be fitted with T-spline surfaces. Zhao et al. [112] discussesed a coupling method of surface patch reconstruction based on T-splines and employed an iteration reconstruction method to construct each patch. A lot of articles focus on the T-spline level set for surface reconstruction and shape metamorphosis [24, 101–105]. Besides all these works, there are also several work focusing on part of the reconstruction problem, such as [32, 61, 74, 75, 77, 97–99, 115].

4.6.5 *Iso-geometric analysis with T-splines*

It was found that T-splines possess the same optimal convergence properties as NURBS with far fewer degrees-of-freedom [3,19,79], see Figure 13 for a simple example. T-spline-based isogeometric analysis has been applied in various contexts. Application areas include fracture and damage [6,83–85], fluid-structure interaction [4], and shells [18]. A design-through-analysis framework utilizing immersed boundary methods, hierarchical refinement, and T-splines is described in [62]. Scott et al. [69] coupled collocated isogeometric boundary element methods and unstructured T-spline surfaces for linear elastostatic problems. Later, Buffa et al. [9] gave an example of a T-spline with linearly dependent blending functions, which means that not all T-splines are analysis-suitable or suitable as a basis for iso-geometric analysis.

Figure 13 IGA with T-splines for advection skew to the mesh, $\theta = 45°$ and $p = 1$ (reprinted with permission from [3])

Thus an important development in the evolution of iso-geometric analysis is the advent of analysis-suitable T-splines which were formulated in [53]. Analysis-suitable T-splines are a canonical class of T-splines which possess the basic mathematical properties of NURBS (linear independence, partition of unity, etc.) while maintaining the local refinement property and design flexibility of general T-spline descriptions. Automatic conversion algorithms from unstructured quadrilateral and hexahedral meshes to T-splines surfaces and volumes is described in [92, 93, 110, 111]. Efficient and canonical finite element data structures for NURBS and T-splines based on Bezier extraction are described in [67]. Mathematical studies of the basic approximation properties of analysis-suitable T-spline spaces are studied in [52]. Hierarchical analysis-suitable T-splines, which are a superset of both analysis-suitable T-splines and hierarchical B-splines, is developed in [20].

5 Polynomial spline over T-mesh

5.1 Spline spaces over T-meshes

The notion of a spline space over a T-mesh was firstly put forward in [14]. The parametric mesh is a collection of axis-aligned rectangles F_i, where the distinct rectangles F_i and F_j can only intersect at points on their edges. The T-mesh requires to be regular, which means that for any vertex in the T-mesh, the set of all the rectangles that contain the vertex has a connected interior [64]. For example, in Figure 14, T-mesh b is a regular T-mesh but T-mesh c is not a regular T-mesh because vertex V does not satisfy the requirement for a regular T-mesh. The rectangles F_i are also called the *faces* or *cells* of the T-mesh. The vertices of the rectangles are called the *nodes* or *vertices* for the T-mesh. The line segment that contains two adjacent vertices on a grid line is called an *edge* of the T-mesh. T-meshes include tensor-product meshes as a special case. However, in contrast to tensor-product meshes, T-meshes are allowed to have *T-junctions*, or *T-nodes*, which are the vertices of one rectangle that lie in the interior of an edge of another rectangle. The domain Ω does not need to be rectangular, so it may have concave corners and holes (the grey region in Figure 14).

If a vertex is on the boundary grid line of the T-mesh, it is called a *boundary vertex*. Otherwise, it is called an *interior vertex*. If both of the vertices of an edge are boundary vertices, the edge is called a *boundary edge*; otherwise, it is called an *interior edge*. A *composite edge* (or c-edge) is a line segment that consists of one or more edges. It is the longest possible line segment where the interior vertices are all T-junctions. An *l-edge* is a line segment which consists of several interior edges. It is the longest possible line segment, where all the interior edges are connected, and the two end points are T-junctions or boundary vertices. If the two end vertices of an l-edge are interior vertices, the l-edge is called an *interior l-edge*. If two end vertices of an l-edge are both boundary vertices, the l-edge is called a *cross-cut*. Otherwise, if one end vertex is a boundary vertex and the other end vertex is an interior vertex, the l-edge is called a *ray*. A *mono-vertex* is the intersection between an interior l-edge and a cross-cut or a ray, and a *free-vertex* is the intersection between cross-cuts and rays. For example, in the T-mesh a in Figure 14, vertex V_{58}, is an interior vertex, V_{15}, V_{16} and V_{57} are boundary vertices. The l-edge $V_{16}V_{46}$ is a cross-cut, while $V_{57}V_{58}$ is a ray, $V_{33}V_{63}$ is an interior l-edges. $V_{33}V_{43}$, $V_{57}V_{58}$ are c-edges while $V_{57}V_{58}$, $V_{33}V_{63}$ are l-edges.

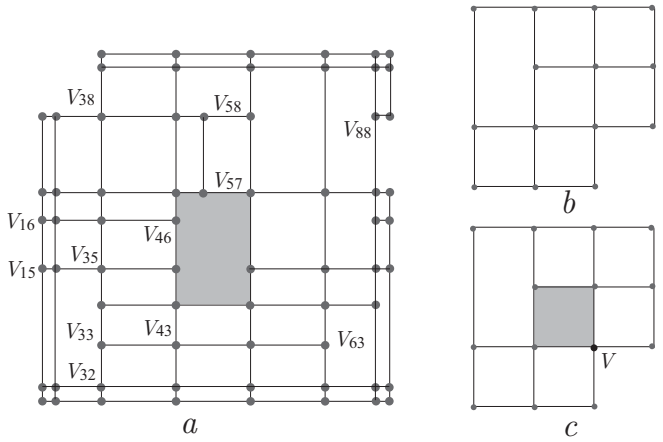


Figure 14 Example T-meshes

Definition 5.1. Suppose $T_1, \ldots, T_n$ is a collection of T-nodes in a T-mesh such that for each $i = 1, \ldots, n$, the vertex T_i lies in the interior of a c-edge with one endpoint at T_{i+1}, where we set $T_{n+1} = T_1$. Then we say that $T_1, \ldots, T_n$ form a cycle.

Definition 5.2. Given a T-mesh $\mathbb{T}$, $\mathcal{F}$ denotes all the cells in $\mathbb{T}$ and Ω the region occupied by all the cells in $\mathbb{T}$. Define

$$\mathcal{S}(m, n, \alpha, \beta, \mathbb{T}) := \{f(s,t) \in C^{\alpha,\beta}(\Omega) | f(s,t)|_\phi \in \mathbb{P}_{mn} \text{ for any } \phi \in \mathcal{F}\},$$

where $\mathbb{P}_{mn}$ is the space of all the polynomials with bi-degree (m, n), and $C^{\alpha,\beta}(\Omega)$ is the space consisting of all the bivariate functions which are continuous in Ω with order α along s direction and with order β along t direction. It is obvious that $\mathcal{S}(m, n, \alpha, \beta, \mathbb{T})$ is a linear space. It is called the spline space over the given T-mesh $\mathbb{T}$.

5.2 Dimension

In order to understand the spline space over a T-mesh, the foundation but non-trivial problem is to calculate the dimension of the space. Until now, many different methods were applied to tackle this issue, including the B-net [14], the minimal determining set method [64], the smoothing cofactor-conformality method [43, 48] and the homological technique [56].

Reduced regularity. In 2006, Deng et al. [14] studied the dimension of the spline space under the constraints that required the order of smoothness is less than half of the degree of the spline functions and provided the following main result.

Theorem 5.3. *If a T-mesh has no cycles (see [14] used a tree structure to define this condition), and $d_1 \geqslant 2\alpha + 1$ and $d_2 \geqslant 2\beta + 1$, then the dimension of spline space over the T-mesh is*

$$\begin{aligned}\dim \mathcal{S}(d_1, d_2, \alpha, \beta, \mathcal{T}) = {} & F(d_1+1)(d_2+1) - E_h(d_1+1)(\beta+1) \\ & - E_v(d_2+1)(\alpha+1) + V(\alpha+1)(\beta+1),\end{aligned}$$

where F, E_h, E_v, and V are the number of aces, horizonal interior edges, vertical interior edges and interior vertices, respectively.

Huang et al. [33], and Schumaker and Wang [63,64] also proved this result using the minimal determining set method and the smoothing cofactors method. Later, Buffa et al. [8] analyzed a special T-spline with reducing regularity using the dimension formula in [14].

Enough mono-vertices. In 2006, Li et al. [43] calculated the dimension of spline space over a T-mesh if each interior l-edges have enough mono-vertices.

Theorem 5.4. *If each horizontal l-edge has at least N^h-1 mono-vertices and each vertical l-edge has at least N^v-1 mono-vertices except two end vertices in a T-mesh, here N^h and N^v are the minimal integer larger or equal to $\frac{d_1+1}{d_1-\alpha}$ and $\frac{d_2+1}{d_2-\beta}$, then the dimension of spline space over the T-mesh is,*

$$\begin{aligned}\dim \mathcal{S}(d_1,d_2,\alpha,\beta,\mathcal{T}) = {} & (d_1+1)(d_2+1)+(C_h-T_h)(d_1+1)(d_2-\beta)\\ & +(C_v-T_v)(d_2+1)(d_1-\alpha)+V(d_1-\alpha)(d_2-\beta),\end{aligned}$$

here C_h, C_v, T_h, T_v and V are the number of horizontal cross-cuts, vertical cross-cuts, horizontal interior l-edges, vertical interior l-edges and interior vertices, respectively.

Diagonalizable *T*-mesh. Li and Deng [48] provided a general formula for the dimension of spline spaces over general planar T-meshes (having concave corners or holes) using smoothing cofactor-conformality method and introduced a new notion, the diagonalizable T-mesh, over which the dimension formula is only associated with the topological information of the T-mesh. Li and Scott [52] computed the dimension of the spline space $\mathcal{S}(d,d,d-1,d-1,\mathbb{T})$ if the T-mesh is an extended T-mesh of an analysis-suitable T-mesh [53] by using the similar idea.

Weighted *T*-mesh. Mourrain [56] gave a general formula for the spline spaces by homological techniques with a term in the dimension formula which is very hard to compute in practice. In the paper, Mourrain defined a class of T-meshes, called weighted T-meshes, over which the dimension can be computed in an explicit formula. The weighted T-meshes are associated with the order of the l-edges. So Mourrain studied the dimension for spline space when the T-mesh is a regular T-subdivision, where the T-mesh could be created by insertion edges into a tensor-product mesh.

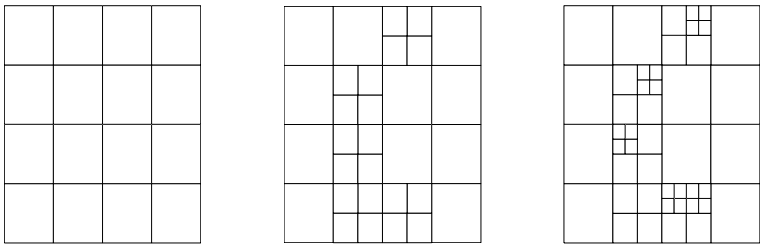
Special hierarchical *T*-mesh. Wu et al. [100] provided the dimension for the spline space $\mathcal{S}(d,d,d-1,d-1,\mathcal{T})$ over a special hierarchical T-mesh using homological algebra technique. Deng et al. [15] derived a dimension formula for C^1 biquadratic spline spaces over any hierarchical T-meshes. Giannelli and Jüttler [28] gave certain conditions such that the hierarchical spline basis spanned the entire space of all piecewise polynomial functions of the given degree and smoothness defined on the underlying grid.

Other generalization. There are several other articles generalizing spline space over T-meshes to 3D T-meshes and T-meshes with L-junctions. For example, Li et al. [49] and Wang [90] discussed the dimension of spline space over 3D hierarchical T-mesh using B-net and minimal determining set method. Huang et al. [34], Lang and Xu [42], Li [44], Zhang [107], and Zhang [109] generalized the dimension formula to more general type of T-meshes. Jin [37], and Villamizar and Mourrain [86] discussed the bound for the dimension of spline space. Schumaker and Wang [65] extended the results to spline space on triangulations with hanging vertices.

Besides all these positive results, Li and Chen [47] discovered that the dimension of the spline space $\mathcal{S}(d_1,d_2,d_1-1,d_2-1,\mathcal{T})$ is instability over certain T-meshes, i.e., the dimension is not only associated with the topological information of the T-mesh but also associated with the geometric information of the T-mesh. And later, Berdinskya et al. [5] provided two more example T-meshes for which dimensions were also instable for spline space $S(5,5,3,3,\mathscr{T})$ and $S(4,4,2,2,\mathscr{T})$. These results suggest us that we can only study some special spline spaces for practical use. The first class of spline spaces are spline spaces with reduced regularity.

5.3 Spline space with reduced regularity

Currently, the spline spaces $\mathcal{S}(m,n,\alpha,\beta,\mathbb{T})$ when the order of smoothness is less than half of the degree of the spline functions have been well understood. In the following, such spline space is called the spline space with reduced regularity, or PHT-splines. The biggest advantages of the PHT-splines are the perfect behavior of local refinement, which will never introduce any additional refinement. The PHT-spline basis functions fulfill all important properties in the context of numerical analysis, i.e., non-negativity, partition of unity, linear independence and local support. Moreover, the PHT-spline formulation facilitates adaptive refinement that is cumbersome for NURBS based finite element formulations. However, the main



(a) level 0 (b) level 1 (c) level 2

Figure 15 Generation of a hierarchical T-mesh

drawback of PHT-spline is the reduced regularity, i.e., it will introduce more degrees of freedoms than the other three approaches. Without lose of generality, we derive all the algorithms in term of bi-cubic spline space with C^1 continuity in the following, which can be generalized to higher degrees very easily.

5.3.1 *Basis functions*

Basis construction has been studied in [16, 46, 51]. This section only provides the detail construction for spline space over a special T-mesh, called hierarchical T-mesh because most of the applications are based on the spline spaces over the T-mesh. Also, for simplicity we only focus on the spline space for $m = n = 3$ and $\alpha = \beta = 1$ which can be very easily to generalize to other degrees.

A *hierarchical T-mesh* is a special type of T-mesh which has a natural level structure. It is defined in a recursive fashion. One generally starts from a tensor-product mesh (level 0). From level k to level $k + 1$, one subdivides a cell at level k into four subcells which are cells at level $k + 1$. For simplicity, the operator subdivides each cell by connecting the middle points of the opposite edges with two straight lines. Check Figure 15 for an example of hierarchical T-mesh.

The basis functions for a PHT-spline space can be constructed in a level-by-level strategy. For the initial level (level 0, denoted as $\mathbb{T}_0$), the standard bi-cubic tensor-product B-splines are used as basis functions. Suppose the initial knot vectors are

$$[u_1, u_2, u_3, \ldots, u_m] \times [v_1, v_2, v_3, \ldots, v_n],$$

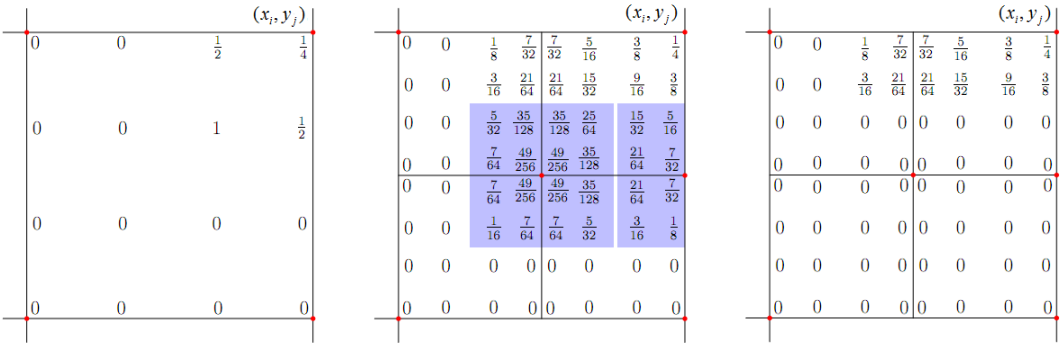
then there are four basis functions to be defined on any vertex (u_i, v_j), since all the vertices are either crossing vertices or boundary vertices (called *basis vertices*). These four basis functions are defined to be four B-spline basis functions with knot vectors

$$(u_{i-1}, u_{i-1}, u_i, u_i, u_{i+1}, u_{i+1}) \times (v_{j-1}, v_{j-1}, v_j, v_j, v_{j+1}, v_{j+1}).$$

Inductively, suppose the basis functions $\{b_j^k\}$, $j = 1, \ldots, d_k$, on $\mathbb{T}_k$ have been constructed, the basis functions on $\mathbb{T}_{k+1}$ can be constructed from two sources: some are from the modifications of the old basis functions on $\mathbb{T}_k$, and others are from the new basis functions associated with the new basis vertices of $\mathbb{T}_{k+1}$.

Notice the fact that a basis function can be represented by specifying its 16 Bézier ordinates (coefficients) in every cell within the support of the basis function. When a cell is refined by adding a cross vertex, the cell is subdivided into four subcells. Each subcell supports the original basis function, and there are 16 Bézier ordinates on it. By adding a cross, we get 5 new vertices, some of which are new basis vertices. Then we have to reset all the associated Bézier ordinates associated with the new basis vertices to zero. Figure 16 illustrates this process. Other than the modification of old basis functions, there are some new basis vertices. For these kind of basis vertices, the basis can be constructed as the B-spline basis functions over their supports. Example basis functions are shown in Figure 17.

Definition 5.5. Given a T-mesh $\mathbb{T}$, suppose the basis functions are $\{b_j^k(u, v)\}$, $j = 1, \ldots, N$, $k = 0, \ldots, 3$, here N is the number of basis vertices. Then a PHT-spline surface $\boldsymbol{S}(s, t)$ over $\mathcal{T}$ can be



(a) 16 Bézier ordinates of the function (b) The function is subdivided into four subcells, after adding a cross (c) Set the Bézier ordinates associated with new basis vertices to be zeros (shaded area in (b))

Figure 16 Modification of a basis function at (x_i, y_j), rectangle vertices are basis vertices

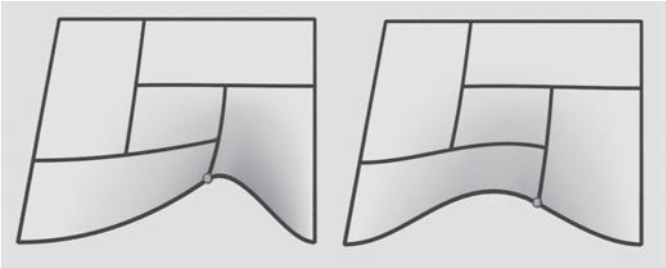


Figure 17 Two images of basis functions

defined as

$$\boldsymbol{S}(s,t) = \sum_{j=1}^{N} \sum_{k=0}^{3} \boldsymbol{C}_j^k b_j^k(s,t), \tag{5.1}$$

where $\boldsymbol{C}_j^k$ are the control points associated with the j-th basis vertex.

The control points $\boldsymbol{C}_j^k$ can be determined by a generalized hermite interpolation procedure [16, 66].

Definition 5.6. For any function $b(s,t)$, its function value $b(s,t)|_{(s_0,t_0)}$, two partial derivatives of first order and mixed partial derivative

$$b_s(s_0,t_0) = \frac{\partial}{\partial s} b(s,t)|_{(s_0,t_0)},$$
$$b_t(s_0,t_0) = \frac{\partial}{\partial t} b(s,t)|_{(s_0,t_0)},$$
$$b_{st}(s_0,t_0) = \frac{\partial^2}{\partial s \partial t} b(s,t)|_{(s_0,t_0)}$$

at some point (s_0,t_0) are called the geometric information of $b(s,t)$ at point (s_0,t_0).

Theorem 5.7. *A PHT-spline surface $\boldsymbol{S}(s,t)$ over $\mathcal{T}$ is unique determined by the geometric information at all basis vertices.*

Proof. Assume the geometric information for some unknown function at i-th basis vertex are f, f_s, f_t, f_{st} respectively. Then the control points of the i-th basis vertex can be computed as

$$(\boldsymbol{C}_i^0, \boldsymbol{C}_i^1, \boldsymbol{C}_i^2, \boldsymbol{C}_i^3) = (f, f_s, f_t, f_{st})(\boldsymbol{B}_i) \tag{5.2}$$

to interpolate the geometric information at the basis vertex. Here $(\boldsymbol{B}_i)$ is a 4×4 matrix for the i-th basis vertex,

$$(\boldsymbol{B}_i)=\begin{pmatrix} 1 & 1 & 1 & 1\\ -\frac{u_i}{\lambda} & \frac{1-u_i}{\lambda} & -\frac{u_i}{\lambda} & \frac{1-u_i}{\lambda}\\ -\frac{v_i}{\mu} & -\frac{v_i}{\mu} & \frac{1-v_i}{\mu} & \frac{1-v_i}{\mu}\\ \frac{u_iv_i}{\lambda\mu} & -\frac{(1-u_i)v_i}{\lambda\mu} & -\frac{u_i(1-v_i)}{\lambda\mu} & \frac{(1-u_i)(1-v_i)}{\lambda\mu} \end{pmatrix},$$

where

$$u_i=\frac{u_i^2}{u_i^1+u_i^2},\quad v_i=\frac{v_i^2}{v_i^1+v_i^2},\quad \lambda=\frac{1}{u_i^1+u_i^2}\quad\text{and}\quad \mu=\frac{1}{v_i^1+v_i^2}.$$

And u_i^1, u_i^2, v_i^1 and v_i^2 are the parameter length of four adjacent edges of the i-th basis vertex. □

5.3.2 *Edge insertion and edge deletion*

Local refinement is the most important operator for splines which is corresponding edge insertion operator for splines over T-meshes. Edge deletion is the inverse operator of the edge insertion. In this section, the general edge insertion and deletion algorithms for a PHT-spline surface in the form of (5.1) are discussed.

Suppose we insert an edge into a T-mesh $\mathcal{T}$ and get a new T-mesh $\mathcal{T}^1$. The first step of the operation is to construct the basis functions for the new T-mesh. Without lose of generality, suppose the edge insertion leads one new basis vertex v_{new} with four new basis functions $n_i(s,t), i=0,\ldots,3$ which can be constructed using the algorithm in [51]. Then for any existing basis function $b_j^k(s,t)$, it will replaced by

$$\hat{b}_j^k(s,t)=b_j^k(s,t)-\sum_{i=0}^{3}\lambda_i n_i(s,t).$$

Here λ_i's satisfy that $b_j^k(s,t)$ and

$$\sum_{i=0}^{3}\lambda_i n_i(s,t)$$

have the same geometric information at v_{new}. So $\hat{b}_j^k(u,v)$ will have vanish geometric information at v_{new}. The second step is to modify the new control points and keep the existing control points unchanged. First, one computes the geometric information of PHT-spline surface at the new basis vertices and then interpolate these geometric information to get the new control points according to (5.2).

Edge deletion is the inverse operator of the edge insertion. Suppose the T-mesh after an edge removal is $\mathcal{T}^2$ and the corresponding spline space is $\mathcal{S}^2$. Since edge deletion is not an exact operator, there are many opinions to define the surface after edge removed. The opinion that one usually uses is to keep the geometric information at all the existing basis vertices unchanged. According to interpolation Theorem 5.7, the key step of edge deletion is define the basis functions for spline space $\mathcal{S}^2$. For more details of the basis functions contruction, the reader is referred to [51].

5.4 Applications

PHT-splines have been applied in fitting [16], stitching [50], simplification [51], adaptive surface reconstruction based on implicit PHT-splines [89]. Also PHT-splines have been applied in solving elliptic equations [76], and iso-geometric analysis [58–60, 91].

5.4.1 *Fitting*

In [16], an efficient scheme is proposed to fit triangle meshes with spline surfaces over hierarchical T-meshes. According to 5.1 and the interpolation Theorem 5.7, the main step is to estimate the geometric information at all the basis vertices. Suppose the vertices positions for the triangle mesh vertices are $\{\boldsymbol{P}_i\}_{i=1}^{N_p}$. Then first, one estimates the geometric information at every vertex $\boldsymbol{P}_i$. For every vertex $\boldsymbol{P}_i$ in the given mesh, its topological neighborhood $\mathcal{N}(\boldsymbol{P}_i)$ is organized with enough points for information

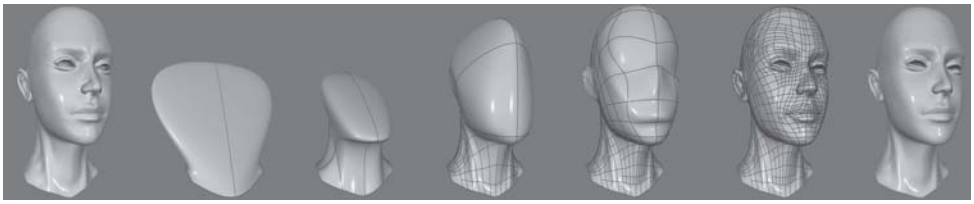


Figure 18 Surface fitting a female head mesh (courtesy of Open3D Project on www.Project-Human.com) with splines over hierarchical T-meshes. The curves on surfaces from b to f are the mapping images of the hierarchical T-meshes. From left to right, are Original mesh, Levels 1, 2, 4, 6 and 10, respectively

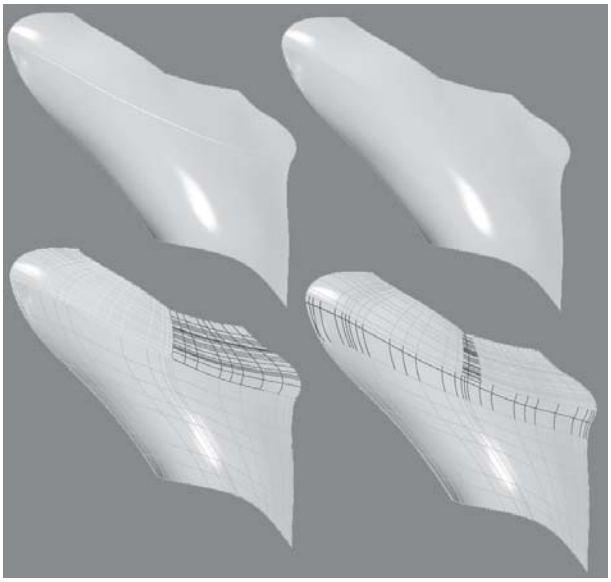


Figure 19 Stitching two pieces and three pieces of surface patches

estimation. Then fit a bi-cubic or bi-quadratic patch to the points in $\mathcal{N}(\boldsymbol{P}_i)$ (assuming a parametrization of the mesh model is obtained). The required geometric information at point $\boldsymbol{P}_i$ is obtained by evaluating the patch at the corresponding parameter value of $\boldsymbol{P}_i$. Second, for each basis vertex $\boldsymbol{Q}$, the geometric information at $\boldsymbol{Q}$ can be obtained by linearly interpolating the geometric information of three neighboring points $\boldsymbol{P}_i$, $\boldsymbol{P}_j$ and $\boldsymbol{P}_k$ of $\boldsymbol{Q}$. For the details for basis construction and fitting process, the reader is referred to [16].

Figure 18 illustrates the surface fitting procedure, where the given mesh is a female head with 19,231 points and 38,388 triangular faces, and a hole around her neck. With the provided parameterization into a square, the new spline surface can be constructed about two seconds with an ordinary personal computer.

5.4.2 *Stitching*

Computer graphics and computer aided design communities prefer piecewise spline patches to represent surfaces. The geometric models are composed with a set of B-spline patches, where the adjacent patches have many gaps when the knot vectors of the adjacent patches do not match. Thus how to keep the smoothness between the adjacent patches is a challenging task. In [50], Li et al. presented a method for stitching several surface patches with a PHT-spline. The method was simple and could be easily used in complex models. Figure 19 shows an result of the stitching algorithm. The left-top picture in Figure 19 shows the gap among three of the patches from a real B-spline model. The picture on the right-top depicts the result with the stitching algorithm with C^1 continuity. The bottom two pictures are the results of the C^0 and C^1 stitching algorithm with T-meshes.

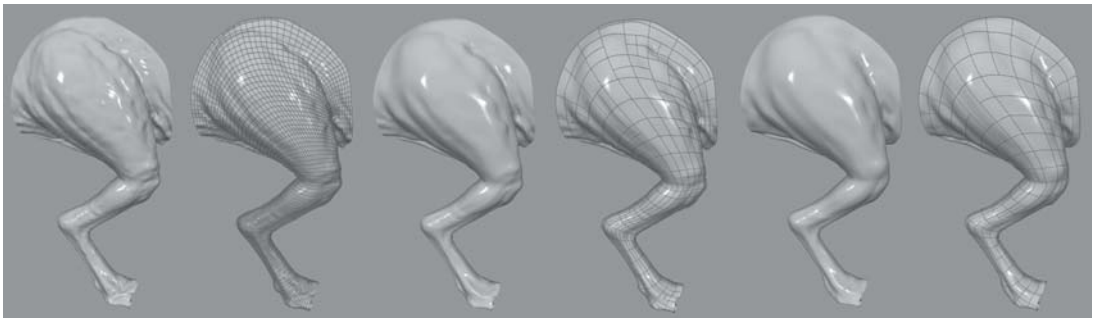


Figure 20 Simplify a B-spline surface into PHT-spline surface using iterative edge removal algorithm

5.4.3 *Simplification*

Surface fitting with tensor-product B-spline surfaces might have many superfluous control points or surface patches. Deng et al. [16], Li et al. [51] provided two algorithms for converting a tensor-product B-spline surface into a PHT-spline surface within a given tolerance. Deng et al. [16] discussed the B-spline surface simplification with PHT-spline using iterative refinement with above fitting algorithm. Li et al. [51] achieved the some purpose using iterative edge removal algorithm. The solution presented does not produce the result with fewest surface patches that falls bellow the given tolerance. Since in general, the models may have several thousands of surface patches, it is intractable to make an exhaustive search. However, the surface will exactly fall under the given tolerance for both algorithms and the second one can work on rational surface and arbitrary topology mesh also. A simple example is illustrated in Figure 20 to show the effect of the iterative edge removal algorithm. The example can be computed within twenty seconds on a personal computer with Pentium 4 CPU 3.20GHz and 1.0GB RAM. The origin model has 8280 bi-cubic patches, for the given error of 1.0%, the algorithm produces 713 patches and for the given error of 1.4%, the algorithm produces 387 patches.

5.4.4 *Implicit PHT-splines*

Wang et al. [89] proposed an adaptive surface reconstruction algorithm based on implicit PHT-splines. The implicit PHT-spline representation could be viewed as an adaptive signed distance fields with globally C^1 continuous.

Definition 5.8. Given a 3D hierarchical T-mesh $\mathbb{T}$, suppose the basis functions are $\{b_j^k(x,y,z)\}$,

$$j = 1, \ldots, N_c, \quad k = 0, \ldots, 7.$$

Here N_c is the number of basis vertices. Then a 3D PHT spline surface $f(x,y,z)$ over $\mathcal{T}$ can be defined as

$$f(x,y,z) = \sum_{j=1}^{N_c} \sum_{k=0}^{7} \boldsymbol{C}_j^k b_j^k(x,y,z), \tag{5.3}$$

where $\boldsymbol{C}_j^k$ are the control points associated with the j-th basis vertex.

Given a set of points $\{\boldsymbol{P}_i\}_{i=1}^{N_p}$ with oriented normals, the goal is to generate a 3D PHT-spline function $f(x,y,z)$ whose zero level set gives a good approximation to the underlying surface. Intuitively, the algorithm tries to use the PHT-spline function to approximate the signed distance field as accurate as possible in the vicinity of signed distance field while the approximation can be rough in regions away from the surface. The scheme is to recursively construct a hierarchical T-mesh with simple and error-guided local refinements that adapt to the target geometric details, and to determine the PHT-spline by estimating the Hermitian information at basis vertices. First, one approximates the target geometry of a point cloud with an implicit surface of polynomial splines over 3D hierarchical T-mesh, which is constructed adaptively by error-guided local refinements. In each progressive level, the PHT-spline function is determined by interpolating the Hermitian information at the basis vertices of the

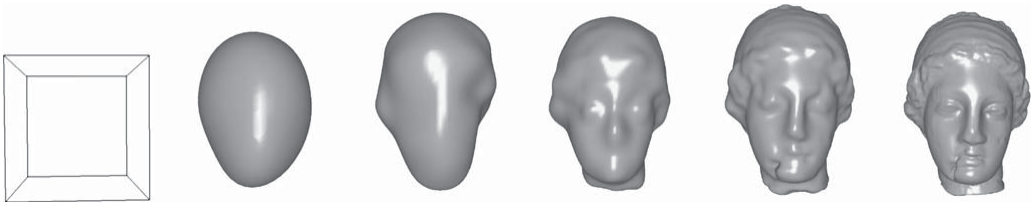


Figure 21 Adaptive reconstruction of Igea model, from left to right are the intermediate results at level 0 to level 5. The leftmost is the T-mesh at level 0 with which there is no surface generated

hierarchical T-mesh, and the Hermitian information at the basis vertices is obtained from the geometric quantities on the underlying surface of the point cloud. As a PHT-spline function in each cell is a tricubic polynomial, so it has strong capability to capture geometric details. And since we can adaptively produces a hierarchical T-mesh, thus the number of basis vertices is roughly one-third of the number of cells. Also in the reconstruction process, one only has to estimate the Hermitian information at the basis vertices instead of fitting local shape to data points in each cell. Figure 21 shows one simple example of adaptive reconstruction of Igea model.

5.4.5 *Iso-geometric analysis*

There are a set of articles trying to consider the flexibility of the T-meshes and simple local refinement algorithm for PHT-splines and applying PHT-splines to iso-geometric analysis [58–60, 76, 91]. In traditional finite element analysis, an adaptive procedure consists of successive loops of the form Solve $\rightarrow$ Estimate $\rightarrow$ Mark $\rightarrow$ Refine.

The essential part of the loops is the Estimate step. Error estimate methods with a posteriori error control have so far been well-developed in adaptive FEA. The posteriori error estimate started with the pioneering work in [2]. Wang et al. [91] followed up on the ideas presented by Verfürth [82] and Ainsworth [1] and derived a residual-based error estimator based on RPHT-splines (rational PHT-splines). The adaptive scheme of [91] is as follows:

1. Build a PHT-based geometry or Rational-PHT-based geometry, use the geometry basis function to represent the physical domain and construct the approximate solution space V^h.
2. Solve the system of equations of the iso-geometric approximation to calculate field variables.
3. Calculate the local error indicator η_K patch by patch. If the total error η is less than the prescribed tolerance, then end.
4. Mark the patches that contribute most to the total error with the marking strategy.
5. Refine the marked patches according to the refinement strategy, and go to Step 2.

The following several Figures 22–24 are PHT-based iso-geomtric analysis the infinite plate with circular hole under constant in-plane tension in the x-direction, which is a benchmark problem in IGA literature. The setup is illustrated in Figure 22. Here, T_x is the magnitude of the applied stress for the infinity plate case, R is the radius of the hole, L is the length of the finite quarter plate, E is Young's modulus, and υ is Poisson's ratio.

At the coarsest level, the geometry is exactly represented with two patches by RPHT-splines. Figure 23 shows the adaptive hierarchical T-meshes after 3, 5, 6, 13 level refinements. The results show that the geometry can be exactly represented with the RPHT-splines at every refinement level. Figure 24 gives the the convergence results with NURBS and with RPHT-splines.

5.5 Spline space with highest order of smoothness

There are also several articles focusing on the spline space

$$\mathcal{S}(m, n, m-1, n-1, \mathbb{T}),$$

which is called spline space with highest order of smoothness. As the dimension for the spline space is instable over some particular T-meshes [47], so the researchers focus on the spline spaces over some special

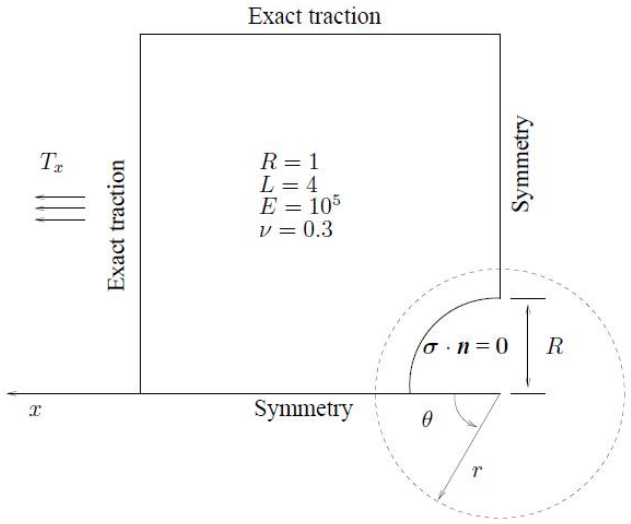


Figure 22 Setup of an infinite elastic plate with a circular hole

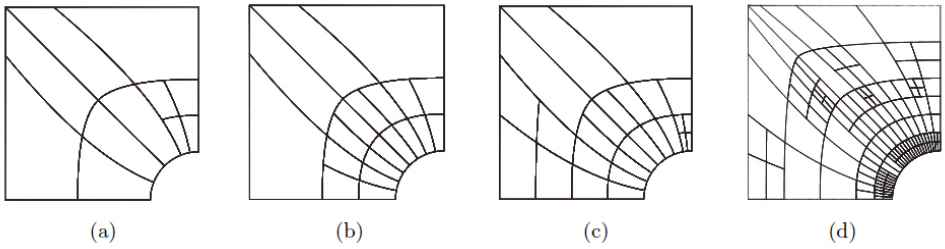


Figure 23 Resulting meshes after 3, 5, 6 and 13 refinements

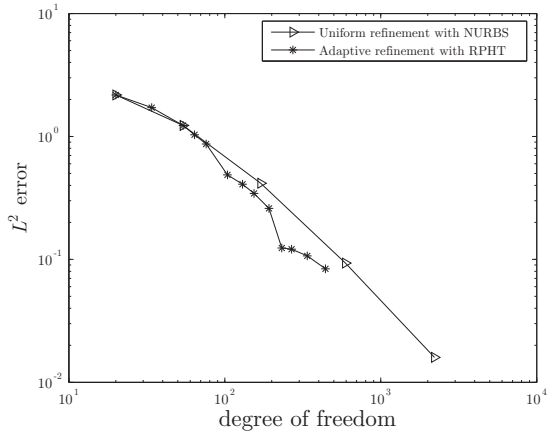


Figure 24 Comparison of the exact L^2 error of x-direction stress

T-meshes. Deng et al. [15] obtained the dimension formula of biquadratic spline spaces over hierarchical T-meshes. But the basis functions still need further construction. Wu et al. [100] presented a dimension formula for a spline space

$$\mathcal{S}(m, n, m-1, n-1, \mathbb{T})$$

when the T-mesh is a certain type of hierarchical T-mesh. They also proposed a method for constructing a set of basis in the paper. However, it is still unclear whether the restriction of the hierarchical T-mesh is flexible enough for application. All the issues associated with the highest order of smoothness are the main future directions of the polynomial splines over T-meshes.

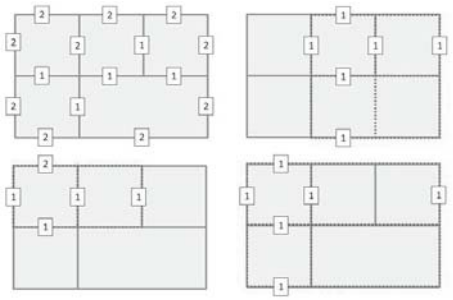


Figure 25 A μ-extended box-mesh M at the top left, and 3 examples of bilinear B-splines and their relation to M. The number on a edge of a mesh is the multiplicities of this edge

6 LR B-splines

Local-refined (LR) B-spline is introduced in [17] which is based on the dimension result in [56]. However, LR B-splines are not always linear independent although there exists an algorithm to check whether the spline is linear dependent and convert the spline to be linear independent by insertion more control points. And what is more, LR B-spline does not have a nature control grid which restricts the application in geometric modeling. Thus, it is still unclear whether LR B-spline will be a powerful tool for design or analysis.

An LR-spline is defined on a μ-extended LR-mesh M which is constructed by inserting line segments starting from a tensor product mesh according to certain rules (can be referred in [17]). For the construction of LR-splines, we use Figure 25 to show the procedure. In Figure 25, the B-spline indicated by the knot line multiplicities at the top to the right does not have support in M since a part of a knot line of the B-spline is not present in M, and the two examples below has support in M, but only the B-spline to the right has minimal support in M. In the B-spline to the left the internal vertical knot line has lower multiplicity than the corresponding mesh-rectangle in M.

In [17], Dokken et al. discussed splines defined over box-meshes (LR-mesh is a special box-mesh), and the dimension formulas of some special cases are also given. When the refinement follows certain rules, LR-splines can span the full spline space defined by μ-extended LR-mesh. Additionally, for the linear dependence of LR-splines, a so-called peeling algorithm is proposed to select the maximum set of linearly independent LR-splines function. In [7], different properties of the LR-splines are analyzed: In particular the coefficients for polynomial representations and their relation with other properties such as linear independence and the number of B-splines covering each element.

7 Conclusion and a few open questions

As described in Section 2, the tensor-product B-spline space can be thought as the linear space spanned by a set of B-spline basis functions or a linear piecewise polynomial space with specified smoothness continuity. Thus, in the beginning development of generalization tensor-produce B-spline space to spline space with local refinement property, all the approaches try to focus on one of the two ways. For example, hierarchical B-splines [26, 87], T-splines [53, 70, 72], and LR B-splines [17] focus on how to define a set of functions in terms of the knot information and control grid. All these approaches are easy to construct the set of functions but it is difficult to derive the mathematical properties for the spline space, which is one of the key issues for IGA. On the other hand, polynomial spline spaces over T-meshes [14, 16, 96] try to generalize the piecewise polynomial linear space to the parametric domain with T-junctions. The approaches are easy to have the mathematical properties but it is difficult to derive the dimension and construct the basis functions for the space in general. But recently, the new development of all these approaches try to combine both ways, such as the completeness of spline space for hierarchical B-splines [28], Analysis-suitable T-splines [53] and LR B-spline [17].

In the end of the paper, we provide some future directions that we think should be pursued associated

with local refinable splines and their application in geometric modeling and iso-geometric analysis.

Analysis-suitable models generation. The key to eliminating the CAD/CAE bottleneck is to create parameterized geometries in the design phase. The recent development of T-spline surfaces illustrates the possibilities to convert trimming NURBS BRep models to untrimmed T-splines which can be directly used as iso-geometric analysis. The most significant challenge facing iso-geometric analysis is developing three-dimensional spline parameterizations from surfaces. This is a problem of geometry generation. The most promising starting points seem to be based on the assumption of untrimmed T-spline or NURBS surfaces containing a volume. This would be applicable to solid parts and also internal flow geometries. The case of external flow geometries appears to be easier. Ideally, it would be desirable to retain the surface parameterization in the process. However, this probably could be relaxed for many practical applications.

Structure and shape optimization. A key advantage of isogeometric analysis is the potential ability of integrating CAD, FEA and shape optimization. The control variables of the geometry provide a concise parameterization that can be used as design variables. Once an optimal design has been obtained, the design can be returned to the CAD system directly because it will already be in the language of the system, namely, NURBS, T-splines, etc.

Efficient computation. Isogeometric analysis has been shown to be more accurate than traditional finite element analysis per degree-of-freedom. So far, sufficiently accurate Gaussian quadrature has been utilized on knot spans, which engenders considerable overhead compared with higher order C^0 elements. It should be very important to generalize the Gaussian quadrature rules for any locally refinable splines.

Application in computer engineering. Local refinable spline should have a big impact in iso-geometric analysis in many important engineering problems, such as dynamic structural applications, contact problems with friction, thin shell elements, curved beam element and so on.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 11031007 and 60903148), the Chinese Universities Scientific Fund, Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, the Chinese Academy of Sciences Startup Scientific Research Foundation, and the State Key Development Program for Basic Research of China (973 Program) (Grant No. 2011CB302400).

References

1 Ainsworth M, Oden J T. A posteriori error estimation in finite element analysis. Comput Methods Appl Engrg, 1997, 142: 1–88
2 Babuška I, Vogelius M. Feedback and adaptive finite element solution of one-dimensional boundary value problems. Numer Math, 1984, 44: 75–102
3 Bazilevs Y, Calo V M, Cottrell J A, et al. Isogeometric analysis using T-splines. Comput Methods Appl Mech Engrg, 2010, 199: 229–263
4 Bazilevs Y, Hsu M C, Scott M A. Isogeometric fluid-structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. Comput Methods Appl Mech Engrg, 2012, 249: 28–41
5 Berdinskya D, Oh M, Kim T, et al. On the problem of instability in the dimension of a spline space over a T-mesh. Comput Graph, 2012, 36: 507–513
6 Borden M J, Scott M A, Verhoosel C V, et al. A phase-field description of dynamic brittle fracture. Comput Methods Appl Mech Engrg, 2012, 217: 77–95
7 Bressan A. Some properties of LR-splines. Comput Aided Geom Design, 2013, 30: 778–794
8 Buffa A, Cho D, Kumar M. Characterization of T-splines with reduced continuity order on T-meshes. Comput Methods Appl Mech Engrg, 2012, 201: 112–126
9 Buffa A, Cho D, Sangalli G. Linear independence of the T-spline blending functions associated with some particular T-meshes. Comput Methods Appl Mech Engrg, 2010, 199: 1437–1445
10 Cardon D L. T-spline simplification. Master's thesis. Provo: Brigham Young University, 2007
11 Cottrell J A, Hughes T J R, Reali A. Studies of refinement and continuity in isogeometric analysis. Comput Methods Appl Mech Engrg, 2007, 196: 4160–4183

12 Cottrell J A, Hughes T J R, Bazilevs Y. Isogeometric Analysis: Toward Integration of CAD and FEA. Chichester: Wiley, 2009

13 Cottrell J A, Reali A, Bazilevs Y, et al. Isogeometric analysis of structural vibrations. Comput Methods Appl Mech Engrg, 2006, 195: 5257–5296

14 Deng J, Chen F, Feng Y. Dimensions of spline spaces over T-meshes. J Comput Appl Math, 2006, 194: 267–283

15 Deng J, Chen F, Jin L. Dimensions of biquadratic spline spaces over T-meshes. J Comput Appl Math, 2013, 238: 68–94

16 Deng J, Chen F, Li X, et al. Polynomial splines over hierarchical T-meshes. Graph Models, 2008, 74: 76–86

17 Dokken T, Lyche T, Pettersen K F. Polynomial splines over locally refined box-partitions. Comput Aided Geom Design, 2013, 30: 331–356

18 Dörfel M, Jüttler B, Simeon B. T-spline finite element method for the analysis of shell structures. Internat J Numer Methods Engrg, 2009, 80: 507–536

19 Dörfel M, Jüttler B, Simeon B. Adaptive isogeometric analysis by local h-refinement with T-splines. Comput Methods Appl Mech Engrg, 2009, 199: 264–275

20 Evans E J, Scott M A, Li X, et al. Hierarchical analysis-suitable T-splines: Formulation, Bézier extraction, and application as an adaptive basis for isogeometric analysis. Comput Methods Appl Mech Engrg, 2015, 284: 1–20

21 Evans J A, Bazilevs Y, Babuška I, et al. n-widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. Comput Methods Appl Mech Engrg, 2009, 198: 1726–1741

22 Farin G E. Curves and Surfaces for CAGD, A Practical Guide, 5th ed. San Francisco: Morgan Kaufmann Publishers, 1999

23 Farin G E. NURBS Curves and Surfaces: From Projective Geometry to Practical Use, 4th ed. Natick: AK Peters Ltd, 2002

24 Feichtinger R, Fuchs M, Jüttler B, et al. Dual evolution of planar parametric spline curves and T-spline level sets. Comput Aided Design, 2008, 40: 13–24

25 Finnigan G T. Arbitrary degree T-splines. Master's thesis. Provo: Brigham Young University, 2008

26 Forsey D, Bartels R. Hierarchical B-spline refinement. ACM SIGGRAPH Comput Graph, 1988, 22: 205–212

27 Forsey D, Bartels R. Surface fitting with hierarchical splines. ACM Trans Graph, 1995, 14: 134–161

28 Giannelli C, Jüttler B. Bases and dimensions of bivariate hierarchical tensor-product splines. J Comput Appl Math, 2013, 239: 162–178

29 Giannelli C, Jüttler B, Speleers H. THB-splines: The truncated basis for hierarchical splines. Comput Aided Geom Design, 2012, 29: 485–498

30 Greiner G, Hormann K. Interpolating and approximating scattered 3D-data with hierarchical tensor product B-splines. In: Méhauté A L, Rabut C, Schumaker L L, eds. Surface Fitting and Multiresolution Methods. Innovations in Applied Mathematics. Nashville: Vanderbilt University Press, 1997, 163–172

31 He Y, Wang K, Wang H, et al. Manifold T-Spline. In: Geometric Modeling and Processing. Lecture Notes in Computer Science, vol. 4077. Berlin-Heidelberg: Springer, 2006, 409–422

32 Huang H. Heterogeneous object scalar modeling and visualization using trivariate T-splines. PhD thesis. Stony Brook: Stony Brook University, 2010

33 Huang Z, Deng J, Feng Y, et al. New proof of dimension formula of spline spaces over T-meshes via smoothing cofactors. J Comput Math, 2006, 24: 501–514

34 Huang Z, Deng J, Li X. Dimensions of spline spaces over general T-meshes. J Univ Sci Technol China, 2006, 36: 573–581

35 Hughes T J R, Cottrell J A, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. Comput Methods Appl Mech Engrg, 2005, 194: 4135–4195

36 Ipson H. T-Spline Merging. Master's thesis. Provo: Brigham Young University, 2006

37 Jin L. A Lower Bound on the Dimension of Bicubic Spline Spaces over T-meshes. In: 2010 International Forum on Information Technology and Applications, vol. 1. New York: IEEE, 2010, 109–112

38 Kang H, Chen F, Deng J. Modified T-splines. Comput Aided Geom Design, 2013, 30: 827–843

39 Kang H, Chen F, Deng J. Hierarchical B-splines on regular triangular partitions. Graph Models, 2014, 76: 289–300

40 Kiss G, Giannelli C, Jüttler B. Algorithms and data structures for truncated hierarchical B-splines. In: Mathematical Methods for Curves and Surfaces. Berlin-Heidelberg: Springer, 2014, 304–323

41 Kraft R. Adaptive and linearly independent multilevel B-splines. In: Méhauté A L, Rabut C, Schumaker L L, eds. Surface Fitting and Multiresolution Methods. Innovations in Applied Mathematics. Nashville: Vanderbilt University Press, 1997, 209–218

42 Lang F-G, Xu X-P. Dimension formulae for tensor-product spline spaces with homogeneous boundary conditions over regular T-meshes. Thai J Math, 2012, 10: 693–701

43 Li C J, Wang R H, Zhang F. Improvement on the dimensions of spline spaces on T-mesh. J Inf Comput Sci, 2006, 3: 235–244

44 Li T. On the dimenson of spline space over T-meshes. Master's thesis. Dalian: Dalian University of Technology, 2007
45 Li W-C, Ray N, Lévy B. Automatic and interactive mesh to T-spline conversion. In: Proceedings of the fourth Eurographics symposium on Geometry processing. Switzerland: Eurographics Association, 2006, 191–200
46 Li X. T-splines and spline over T-meshes. PhD thesis. Hefei: University of Science and Technology of China, 2008
47 Li X, Chen F. On the instability in the dimension of spline space over particular T-meshes. Comput Aided Geom Design, 2011, 28: 420–426
48 Li X, Deng J. On the dimension of splines spaces over T-meshes with smoothing cofactor-conformality method. Comput Aided Geom Design, 2015, doi:10.1016/j.cagd.2015.12.002
49 Li X, Deng J, Chen F. Dimensions of spline spaces over 3D hierarchical T-meshes. J Inf Comput Sci, 2006, 3: 487–501
50 Li X, Deng J, Chen F. Surface modeling with polynomial splines over hierarchical T-meshes. Vis Comput, 2007, 23: 1027–1033
51 Li X, Deng J, Chen F. Polynomial splines over general T-meshes. Vis Comput, 2010, 26: 277–286
52 Li X, Scott M A. Analysis-suitable T-splines: Characterization, refinablility and approximation. Math Models Methods Appl Sci, 2014, 24: 1141–1164
53 Li X, Zheng J, Sederberg T W, et al. On the linear independence of T-splines blending functions. Comput Aided Geom Design, 2012, 29: 63–76
54 Lipton S, Evans J A, Bazilevs Y, et al. Robustness of isogeometric structural discretizations under severe mesh distortion. Comput Methods Appl Mech Engrg, 2010, 199: 357–373
55 Myles A, Pietroni N, Kovacs D, et al. Feature-aligned T-meshes. ACM Trans Graph, 2010, 29: 117
56 Mourrain B. On the dimension of spline spaces on planar T-meshes. Math Comp, 2014, 83: 847–871
57 Nasri A, Sinno K, Zheng J. Local T-spline surface skinning. Vis Comput, 2012, 28: 787–797
58 Nguyen-Thanh N, Kiendl J, Nguyen-Xuan H, et al. Rotation free isogeometric thin shell analysis using PHT-splines. Comput Methods Appl Mech Engrg, 2011, 200: 3410–3424
59 Nguyen-Thanh N, Muthu J, Zhuang X, et al. An adaptive three-dimensional RHT-splines formulation in linear elasto-statics and elasto-dynamics. Comput Mech, 2013, 53: 369–385
60 Nguyen-Thanh N, Nguyen-Xuan H, Bordas S P A, et al. Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids. Comput Methods Appl Mech Engrg, 2011, 200: 1892–1908
61 Peng X, Tang Y. Automatic reconstruction of T-spline surfaces. J Image Graph, 2010, 15: 1818–1825
62 Schillinger D, Dedé L, Scott M A, et al. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline cad surfaces. Comput Methods Appl Mech Engrg, 2012, 249: 116–150
63 Schumaker L L, Wang L. Spline spaces on tr-meshes with hanging vertices. Numer Math, 2011, 118: 531–548
64 Schumaker L L, Wang L. Approximation power of polynomial splines on T-meshes. Comput Aided Geom Design, 2012, 29: 599–612
65 Schumaker L L, Wang L. Splines on triangulations with hanging vertices. Constr Approx, 2012, 36: 487–511
66 Schumaker L L, Wang L. On hermite interpolation with polynomial splines on T-meshes. J Comput Appl Math, 2013, 240: 42–50
67 Scott M A, Borden M J, Verhoosel C V, et al. Isogeometric finite element data structures based on Bézier extraction of T-splines. Internat J Numer Methods Engrg, 2011, 88: 126–156
68 Scott M A, Li X, Sederberg T W, et al. Local refinement of analysis-suitable T-splines. Comput Methods Appl Mech Engrg, 2012, 213: 206–222
69 Scott M, Simpson R, Evans J, et al. Isogeometric boundary element analysis using unstructured T-splines. Comput Methods Appl Mech Engrg, 2013, 254: 197–221
70 Sederberg T W, Cardon D L, Finnigan G T, et al. T-spline simplification and local refinement. ACM Trans Graph, 2004, 23: 276–283
71 Sederberg T W, Finnigan G T, Li X, et al. Watertight trimmed NURBS. ACM Trans Graph, 2008, 27: 79
72 Sederberg T W, Zheng J, Bakenov A, et al. T-splines and T-NURCCSs. ACM Trans Graph, 2003, 22: 477–484
73 Sederberg T W, Zheng J, Sewell D, et al. Non-uniform recursive subdivision surfaces. Proce Siggraph, 1999, 43: 387–394
74 Song W, Yang X. Freeform deformation with weighted T-splin. Vis Comput, 2005, 21: 139–151
75 Tang Y, Li Y, Chen Z, et al. Implicit T-spline surface reconstruction to achieve closure. J Comput Aided Design Comput Graph, 2011, 23: 270–275
76 Tian L, Chen F, Du Q. Adaptive finite element methods for elliptic equations over hierarchical T-meshes. J Comput Appl Math, 2011, 236: 878–891
77 Tong W, Feng Y, Chen F. A surface reconstruction based on implicit T-spline surfaces. J Comput Aided Design Comput Graph, 2006, 18: 358–365
78 T-Splines Inc. http://www.tsplines.com/rhino, 2011
79 Veiga L B, Buffa A, Sangalli D C G. Isogeometric analysis using T-splines on two-patch geometries. Comput Methods

Appl Mech Engrg, 2011, 200: 1787–1803
80 Veiga L B, Buffa A, Sangalli D C G. Analysis-suitable T-splines are dual-compatible. Comput Methods Appl Mech Engrg, 2012, 249: 42–51
81 Veiga L B, Buffa A, Sangalli D C G, et al. Analysis-suitable T-splines of arbitrary degree: Definition and properties. Math Models Methods Appl Sci, 2013, 23: 1979–2003
82 Verfürth R. A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques. New York: John Wiley & Sons Inc, 1996
83 Verhoosel C V, Scott M A, Borden M J, et al. Discretization of higher-order gradient damage models using isogeometric finite elements. In: ICES report 11-12. The Institute for Computational Engineering and Sciences. Texas: The University of Texas at Austin, 2011, 89–120
84 Verhoosel C V, Scott M A, de Borst R, et al. An isogeometric approach to cohesive zone modeling. Internat J Numer Methods Engrg, 2011, 87: 336–360
85 Verhoosel C V, Scott M A, Hughes T J R, et al. An isogeometric analysis approach to gradient damage models. Internat J Numer Methods Engrg, 2011, 86: 115–134
86 Villamizar N, Mourrain B. Bounds on the dimension of splines spaces. Effective Methods Algebra Geom, http://www.math.kth.se/mega2011/MEGA/Nelly.pdf, 2011
87 Vuong A-V, Giannelli C, Jüttler B, et al. A hierarchical approach to adaptive local refinement in isogeometric analysis. Comput Methods Appl Mech Engrg, 2011, 200: 3554–3567
88 Wang H, He Y, Li X, et al. Geometry-aware domain decomposition for T-spline based manifold modeling. Comput Graph, 2009, 33: 359–368
89 Wang J, Yang Z, Jin L, et al. Adaptive surface reconstruction based on implicit PHT-splines. In: SPM '10 Proceedings of the 14th ACM. Symposium on Solid and Physical Modeling. New York: ACM, 2010, 101–110
90 Wang L. Trivariate polynomial splines on 3D T-meshes. PhD thesis. Nashville: The Vanderbilt University, 2012
91 Wang P, Xu J, Deng J, et al. Adaptive isogeometrican alysis using rational PHT-splines. Comput Aided Design, 2011, 43: 1438–1448
92 Wang W, Zhang Y, Lie L, et al. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. Comput Aided Design, 2013, 45: 351–360
93 Wang W, Zhang Y, Xu G, et al. Converting an unstructured quadrilateral/hexahedral mesh to a rational T-spline. Comput Mech, 2012, 50: 65–84
94 Wang Y, Zheng J. Curvature-guided adaptive T-spline surface fitting. Compu Aided Geom Design, 2013, 45: 1095–1107
95 Wang Y, Zheng J, Seah H S. Conversion between T-splines and hierarchical B-splines. In: Proceedings of the 8th IASTED International Conference Computer Graphics and Imaging. Calgary: Acta Press, 2005, 8–13
96 Weller F, Hagen H. Tensor product spline spaces with knot segments. In: Daehlen M, Lyche T, Schumaker L, eds. Mathematical Methods for Curves and Surfaces. Nashville: Vanderbilt University Press, 1995, 563–572
97 Weng B, Pan R J, Yao Z Q, et al. Watermarking T-spline surfaces. In: 11th IEEE International Conference on Communication Technology. New York: IEEE, 2008: 773–776
98 Weng B, Pan R J, Yao Z Q, et al. A fragile watermarking algorithm for T-spline surfaces. In: IEEE Youth Conference on Information. Computing and Telecommunication. New York: IEEE, 2009, 546–549
99 Weng B, Pan R J, Yao Z Q, et al. Robust watermarking for T-spline surfaces based on nonuniform B-spline wavelets transformation. In: IEEE Youth Conference on Information Computing and Telecommunications. New York: IEEE, 2010, 335–338
100 Wu M, Deng J, Chen F. The dimension of spline spaces with highest order smoothness over hierarchical T-meshes. Comput Aided Geom Design, 2013, 30: 20–34
101 Yang H, Fuchs M, Juettler B, et al. Evolution of T-spline level-sets for 2D/3D shape reconstruction. Tech Rep, http://www.industrial-geometry.at/events/strobl2006/2006-06huaiping.pdf, 2006
102 Yang H, Fuchs M, Jüttler B, et al. Evolution of T-spline level sets with distance field constraints for geometry reconstruction and image segmentation. In: Proceedings of the IEEE International Conference on Shape Modeling and Applications. Linz: Johannes Kepler University, 2006, 247–252
103 Yang H, Jüttler B. 3D shape metamorphosis based on T-spline level sets. Vis Comput, 2007, 23: 1015–1025
104 Yang H, Jüttler B. Evolution of T-spline level sets for meshing non-uniformly sampled and incomplete data. Vis Comput, 2008, 24: 435–448
105 Yang H, Kepler J, Juttler B. Meshing non-uniformly sampled and incomplete data based on displaced T-spline level sets. In: IEEE International Conference on Shape Modeling and Applications. Linz: Johannes Kepler University, 2007, 251–260
106 Yang X, Zheng J. Approximate T-spline surface skinning. Comput Aided Design, 2012, 44: 1269–1276
107 Zhang F. On the dimension of spline space over several special type of T-meshes. Master's thesis. Dalian: Dalian Technologic School, 2005

108 Zhang J, Li X. On the linear independency and partition of unity of arbitrary degree analysis-suitable T-splines. Comm Math Stat, 2015, 2: 353–364
109 Zhang M. On the properties of T-splines. Master's thesis. Jinlin: Jinlin University, 2005
110 Zhang Y, Wang W, Hughes T J R. Solid T-spline construction from boundary representations for genus-zero geometry. Comput Methods Appl Mech Engrg, 2012, 249: 185–197
111 Zhang Y, Wang W, Hughes T J R. Conformal solid T-spline construction from boundary T-spline representations. Comput Mech, 2013, 51: 1051–1059
112 Zhao X J, Lu M, Zhang H X. Patch reconstruction with T-splines iterated fitting for mesh surface. Appl Mech Mater, 2011, 88: 491–496
113 Zheng J, Wang Y. Periodic T-splines and tubular surface fitting: Curves and surfaces. In: 7th International Conference, Avignon. Lecture Notes in Computer Science, vol. 6920. Berlin-Heidelberg: Springer, 2012, 731–746
114 Zheng J, Wang Y, Seah H S. Adaptive T-spline surface fitting to z-map models. In: Proceeding GRAPHITE '05 Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia. Providence, RI: Amer Math Soc, 2005, 405–411
115 Zhuang X. Improvement on the local refinement algorithm for T-splines. Master's thesis. Dalian: Dalian University of Technology, 2008