Appl. Math. J. Chinese Univ.
2014, 29(4): 410-421

NUAT T-splines of odd bi-degree and local refinement

DUAN Xiao-juan WANG Guo-zhao

Abstract. This paper presents a new kind of spline surfaces, named non-uniform algebraic-trigonometric T-spline surfaces (NUAT T-splines for short) of odd bi-degree. The NUAT T-spline surfaces are defined by applying the T-spline framework to the non-uniform algebraic-trigonometric B-spline surfaces (NUAT B-spline surfaces). Based on the knot insertion algorithm of the NUAT B-splines, a local refinement algorithm for the NUAT T-splines is given. This algorithm guarantees that the resulting control grid is a T-mesh as the original one. Finally, we prove that, for any NUAT T-spline of odd bi-degree, the linear independence of its blending functions can be determined by computing the rank of the NUAT T-spline-to-NUAT B-spline transformation matrix.

§1 Introduction

Nowadays NURBS is widely used for generating and representing curves and surfaces in industry. T-spline, introduced by Sederberg, et al. [1-2], is a kind of generalization of non-uniform B-spline surfaces and has been proved to overcome many limitations inherent in NURBS. One advantage of T-splines is local refinement. NURBS control points must lie topologically in a rectangular grid. Therefore NURBS refinement requires the insertion of an entire row or column of control points, which means that a large number of NURBS control points serve no purpose other than to satisfy topological constraints. However, the T-spline refinement allows partial rows or columns of control points, which means that T-splines need much less control points than NURBS. Another advantage is that T-spline models are watertight, whereas NURBS models comprised of distinct patches generally fit together with unwanted gaps. These undesirable gaps place a heavy burden on model creators, who must repair a widened gap whenever the model is deformed. Merging two B-spline surfaces with different knot vectors into a single T-spline is a possible way to solve this problem [1-2, 6]. Besides, the T-spline can be used in isogemetric analysis for its nice local refinement properties [9-11].

A T-spline is easy to be converted into the equivalent B-spline surface based on the local refinement of T-splines. Meanwhile, an existing NURBS model can also be converted into a T-spline through simplification within a certain error tolerance. This can eliminate a large number of superfluous control points in geometric modeling and simplify the representation of the models [2].

Received: 2013-10-15.
MR Subject Classification: 65D07, 68U05.
Keywords: odd bi-degree, non-uniform algebraic-trigonometric T-spline, local refinement, blending function, linear independence.
Digital Object Identifier(DOI): 10.1007/s11766-014-3235-x.
Supported by the National Natural Science Foundation of China (60933008 and 61272300).

Many transcendental curves which play an important role in engineering such as the helix, the cycloid and the catenary cannot be represented exactly by NURBS. In order to avoid these inconvenience of NURBS, many new splines are constructed. C-curves, generated over the space spanned by $\{\sin t, \cos t, t, 1 | 0 \le t \le \alpha\}$, are introduced by Zhang [3-4]. This kind of curves can exactly represent cycloids and arcs of circles. Non-uniform algebraic-trigonometric B-splines (NUAT B-splines) generated over the space spanned by $\{1, t, \cdots, t^{k-3}, \sin t, \cos t\} (k \ge 3)$ are introduced by Wang, et al. [5]. The NUAT B-splines which can represent the cycloid and the helix exactly can be seen as an extension of C-curves. Li, et al. [8] and Lü, et al. [9] propose the H-Bézier curves and the uniform hyperbolic polynomial B-spline curves in the space spanned by $\{1, t, \cdots, t^{k-3}, \sinh t, \cosh t\}$. This kind of curves can exactly represent the catenary.

T-spline is a generalization of NURBS. Therefore, applying the T-spline framework to the NUAT B-spline surface is significant. In this paper, our main contributions are:

1. The definition of the NUAT T-splines of odd bi-degree is given. It is a generalization of NUAT B-spline surfaces.

2. We propose a local refinement algorithm for NUAT T-splines. Local refinement of splines is an important issue in isogeometric analysis [10-12]. This algorithm guarantees that the resulting control grid is a T-mesh, which is better than the local refinement algorithm of T-splines proposed by Sederberg, et al. [2].

3. A necessary and sufficient condition for any NUAT T-spline's blending functions to be linearly independent is given. It can be determined by computing the rank of the NUAT T-spline-to-NUAT B-spline transformation matrix.

The paper is organized as follows. Section 2 introduces the T-mesh for the NUAT T-splines. In Section 3, we give the definition of the NUAT T-splines of odd bi-degree. A local refinement algorithm for the NUAT T-splines of odd bi-degree is presented in Section 4. In Section 5, we discuss the linear independence of the blending functions for any given NUAT T-spline of odd bi-degree. We conclude the paper in Section 6.

§2 T-mesh

A NUAT B-spine surface is a grid-based spline surface whose control mesh is simply a rectangular grid. A NUAT T-spline is defined as a point-based spline surface whose control mesh allows T-junctions. Hence, the NUAT T-splines generalize NUAT B-spline surfaces to allow partial rows or columns of control points. Each control point of a NUAT T-spline corresponds to one blending function according to the control mesh.

The control grid of a NUAT T-spline is basically a rectangular gird that permits T-junctions. It is called a T-mesh for the spline surface, which is similar to the T-mesh of a T-spline. To define a basis, the information of knot vectors must be assigned to the T-mesh. Thus, the T-mesh can also be regarded as the knot vector space.

Let $\mathbf{u} = \{u_{1-\lceil p/2 \rceil}, \cdots, u_1, \cdots, u_s, \cdots, u_{s+\lceil p/2 \rceil}\}$ be the horizontal global knot vector and $\mathbf{v} = \{v_{1-\lceil q/2 \rceil}, \cdots, v_1, \cdots, v_t, \cdots, v_{t+\lceil q/2 \rceil}\}$ be the vertical global knot vector. A T-mesh is a rectangular partition of the domain $[u_{1-\lceil p/2 \rceil}, u_{s+\lceil p/2 \rceil}] \times [v_{1-\lceil q/2 \rceil}, v_{t+\lceil q/2 \rceil}]$, where the vertices of the rectangles have coordinates belonging to $\mathbf{u}$ and $\mathbf{v}$. A NUAT T-spline of odd bi-degree (p, q) is defined in terms of the above T-mesh and global knot vectors $\mathbf{u}$ and $\mathbf{v}$. The region $[u_1, u_s] \times [v_1, v_t]$ is the active region of this T-mesh. All the control points are located in this area. Each control point $\mathbf{P}$ in the active region corresponds to a unique pair of knots (u_i, v_j), where u_i is the horizontal coordinate of $\mathbf{P}$ and v_j is the vertical coordinate of $\mathbf{P}$ in the knot vector space. The T-mesh permits T-junctions which means that vertices may have valence 3

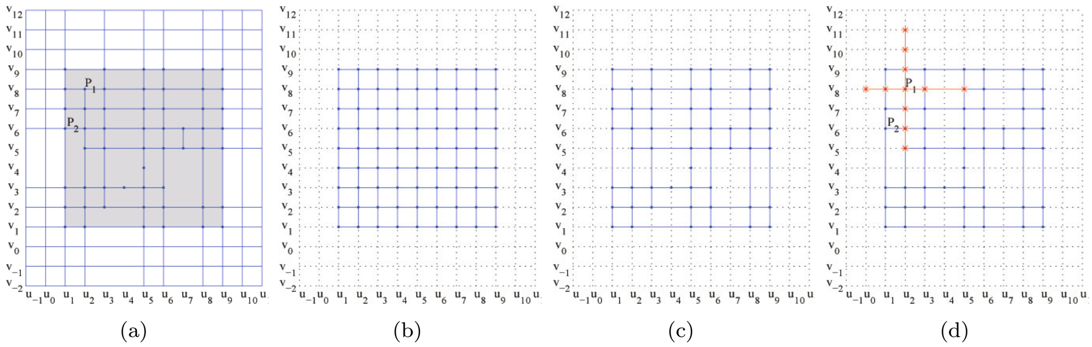


Figure 1: A NUAT T-spline example of bi-degree $(3, 5)$. (a) T-mesh; (b) Corresponding B-mesh; (c) TB-mesh; (d) The knot vectors of $\mathbf{P}_1$ in TB-mesh.

or 2. If the valence of the vertex is 3, the vertex is a T-node. If the valence of the vertex is 2, the vertex should be shared by two u-edges or two v-edges. L-nodes which are shared by one u-edge and one v-edge are not allowed. The following NUAT T-splines of bi-degree (p, q) are of odd bi-degree, which means that p and q are positive odd integers.

The control mesh of the NUAT B-spline surface with bi-degree (p, q) defined in terms of the global knot vectors $\mathbf{u}$ and $\mathbf{v}$ is called the corresponding B-mesh of the NUAT T-spline. Each vertex in the B-mesh has valence 4. Therefore, when a T-mesh forms a rectangular grid with no T-junctions, the NUAT T-spline degenerates to a NUAT B-spline surface.

For brevity, the T-mesh and the corresponding B-mesh can be combined in a single mesh called TB-mesh which contains the information of the two meshes. The TB-mesh can be obtained by changing the corresponding region of the B-mesh using the active region of the T-mesh. By observing the TB-mesh, it is easy to find the control points needed to be added when a NUAT T-spline is converted into an equivalent NUAT B-spline surface.

Figure 1.a shows the T-mesh of a NUAT T-spline of bi-degree $(3, 5)$. $\mathbf{u} = \{u_{-1}, u_0, \cdots, u_{11}\}$ is the horizontal global knot vector and $\mathbf{v} = \{v_{-2}, v_{-1}, \cdots, v_{12}\}$ is the vertical global knot vector. $[u_1, u_9] \times [v_1, v_9]$ is the active region in which all of the control points locate. Both $\mathbf{P}_1$ and $\mathbf{P}_2$ are control points in the T-mesh. $\mathbf{P}_1$ corresponds to a unique pair of knots (u_2, v_8) and (u_1, v_6) are the knot coordinates of $\mathbf{P}_2$. $\mathbf{P}_1$ is a T-node. Figure 1.b presents the corresponding B-mesh of Figure 1.a while Figure 1.c shows the corresponding TB-mesh of Figure 1.a.

The T-mesh not only contains the information of the control points, but also contains the information of the knot vectors. More importantly, the information of the blending functions is included in the T-mesh. The T-mesh must satisfy three conditions as below [1].

- **Condition 1.** A T-mesh is basically a rectangular grid that allows T-junctions.

- **Condition 2.** The sum of knot intervals on opposing edges of any face must be equal.

- **Condition 3.** If a T-junction on one edge of a face can be connected to a T-junction on an opposing edge of the face (thereby splitting the face into two faces) without violating Condition 2, that edge must be included in the T-mesh.

§3 Non-uniform algebraic-trigonometric T-splines of odd bi-degree

A NUAT T-spline of odd bi-degree is a point-based spline surface whose control points have no topological relationship with each other. Each control point in the T-mesh corresponds to a blending function. The blending functions are associated with the basis functions of the NUAT B-spline surface.

3.1 The basis functions of NUAT B-splines

Let T be a given knot sequence $\{t_i\}_{-\infty}^{+\infty}$ with $0 \leq t_{i+1} - t_i < \pi$. A NUAT B-spline of order k is generated over the space spanned by $\{1, t, \cdots, t^{k-3}, \sin t, \cos t\}$ in which k is an arbitrary integer greater than or equal to 3. A set of basis functions $N_{i,k}(t)$ of the space is defined as follows [5].

$$N_{i,2}(t) = \begin{cases} \frac{\sin(t-t_i)}{\sin(t_{i+1}-t_i)}, & t_i < t \leq t_{i+1}, \\ \frac{\sin(t_{i+2}-t)}{\sin(t_{i+2}-t_{i+1})}, & t_{i+1} < t \leq t_{i+2}, \\ 0, & \text{otherwise}. \end{cases} \tag{1}$$

For $k \geq 3$, $N_{i,k}(t)$ is defined recursively by

$$N_{i,k}(t) = \int_{-\infty}^{t} (\delta_{i,k-1} N_{i,k-1}(s) - \delta_{i+1,k-1} N_{i+1,k-1}(s)) ds, k \geq 3, i = 0, \pm 1, \cdots, \tag{2}$$

where

$$\delta_{i,k} = [\int_{-\infty}^{+\infty} N_{i,k}(s) ds]^{-1}. \tag{3}$$

It is a NUAT B-spline basis with simple knot sequence. If there are multiple knots in the knot sequence, we set $\delta_{i,k} N_{i,k} = 0$ when $N_{i,k} = 0$ (set $0/0 = 0$). However, $\delta_{i,k} N_{i,k}$ has to satisfy

$$\int_{-\infty}^{t} \delta_{i,k} N_{i,k}(t) dt = \begin{cases} 1, & t \geq t_{i+k}, \\ 0, & t < t_{i+k}. \end{cases} \tag{4}$$

The NUAT B-spline of order k is $k - r_i - 1$ times continuously differential at a knot t_i if t_i has multiplicity r_i.

The basis $N_{i,k}(t)$ has many important properties: positivity, local support, partition of unity and linear independence.

3.2 Non-uniform algebraic-trigonometric T-splines of odd bi-degree

A NUAT T-spline of odd bi-degree (p, q) is defined in terms of a T-mesh and global knot vectors $\mathbf{u} = \{u_{1-\lceil p/2 \rceil}, \cdots, u_1, \cdots, u_s, \cdots, u_{s+\lceil p/2 \rceil}\}$ and $\mathbf{v} = \{v_{1-\lceil q/2 \rceil}, \cdots, v_1, \cdots, v_t, \cdots, v_{t+\lceil q/2 \rceil}\}$. The equation of the NUAT T-spline surface in homogeneous form is

$$\mathbf{P}(u, v) = (x(u, v), y(u, v), z(u, v), \omega(u, v)) = \sum_{i=1}^{n} B_i(u, v) \mathbf{P}_i \tag{5}$$

where $\mathbf{P}_i = (x_i, y_i, z_i, \omega_i) \in \mathbf{R^4}$ are homogeneous control points whose weights are ω_i, and whose Cartesian coordinates are $\frac{1}{\omega_i}(x_i, y_i, z_i)$. Likewise, the Cartesian coordinates of points on the surface are given by

$$\frac{\sum_{i=1}^{n} B_i(u, v)(x_i, y_i, z_i)}{\sum_{i=1}^{n} \omega_i B_i(u, v)}$$

$B_i(u, v)$ in (5) are the blending functions and are given by $B_i(u, v) = N[\mathbf{u}^i](u) N[\mathbf{v}^i](v)$, where $N[\mathbf{u}^i](u)$ is the NUAT B-spline basis function of degree p associated with the knot vector

$\mathbf{u}^i = [u^i_{j-\lceil p/2\rceil}, \cdots, u^i_j, \cdots, u^i_{j+\lceil p/2\rceil}]$ and $N[\mathbf{v}^i](v)$ is the NUAT B-spline basis function of degree q associated with the knot vector $\mathbf{v}^i = [v^i_{k-\lceil q/2\rceil}, \cdots, v^i_k, \cdots, v^i_{k+\lceil q/2\rceil}]$. The knot vectors $\mathbf{u}^i$ and $\mathbf{v}^i$ for each blending function $B_i(u,v)$ corresponding to $\mathbf{P}_i$ are deduced from the T-mesh according to the Knot Vector Determining Rule (KVD Rule) as follows.

KVD Rule. (u^i_j, v^i_k) are the knot coordinates of $\mathbf{P}_i$. Consider a ray $R(\alpha) = (u^i_j + \alpha, v^i_k)$ $(\alpha > 0)$ in the parameter space. Then u^i_{j+m} is the $\mathbf{u}$ coordinate of the m-th u-edge intersected by the ray (not including the initial (u^i_j, v^i_k)). u-edge means a vertical line segment of constant $\mathbf{u}$. Similarly, consider a ray $R(\beta) = (u^i_j - \beta, v^i_k)$ $(\beta > 0)$ in the parameter space. Then u^i_{j-m} is the $\mathbf{u}$ coordinate of the m-th u-edge intersected by the ray (not including the initial (u^i_j, v^i_k)). The knot vector $\mathbf{v}^i$ can be determined in a similar way (where $m = 1, 2, \cdots, \lceil p/2\rceil$) [1].

In the T-mesh shown in Figure 1.a, the horizontal knot vector of $\mathbf{P}_1$ is $[u_0, u_1, u_2, u_3, u_5]$ and the vertical knot vector is $[v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}]$. The horizontal knot vector of $\mathbf{P}_2$ is $[u_{-1}, u_0, u_1, u_2, u_3]$ with the vertical knot vector $[v_1, v_2, v_3, v_6, v_7, v_8, v_9]$. The knot vectors of $\mathbf{P}_1$ are shown in Figure 1.d labeled with '*'.

§4 Local refinement algorithm

Local refinement is an interesting property of T-splines in geometric modeling since it can make the solid more attractive. More over, it plays an important role in isogeometric analysis using T-splines [10-12].

The local refinement of NUAT T-spline means to insert one or more control points into a NUAT T-spline mesh without changing the shape of the surface. This procedure can also be called local knot insertion, since the addition of control points into a T-mesh must be accompanied with knots inserted into neighboring blending functions. With the knot insertion, the blending function is split into two new scaled blending functions.

4.1 Knot insertion of the NUAT B-splines

Let $T := \{t_i\}^{+\infty}_{-\infty}$ be a knot sequence, and let $T^1 := \{t^1_i\}^{+\infty}_{-\infty}$ be a new knot sequence obtained by inserting a new knot t' into T with $t_i \le t' < t_{i+1}$. $N_{j,k}(t)$ and $N^1_{j,k}(t)$ are defined as in (2) on the knot sequence T and T^1 respectively. For all $j, k \ge 2$ [5],

$$N_{j,k}(t) = \lambda_{j,k} N^1_{j,k}(t) + \mu_{j+1,k} N^1_{j+1,k}(t), \tag{6}$$

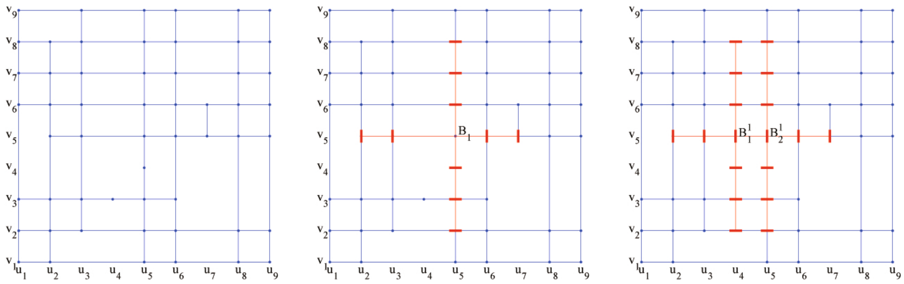
where for $0 \le r < k$

$$\lambda_{j,k} = \begin{cases} 1, & j \le i-k, \\ \lambda_{j,k-1}\frac{\delta_{j,k-1}}{\delta^1_{j,k-1}}, & i-k < j < i-r+1, \\ 0, & j \ge i-r+1, \end{cases}$$

$$\mu_{j,k} = \begin{cases} 0, & j \le i-k+1, \\ \mu_{j+1,k-1}\frac{\delta_{j,k-1}}{\delta^1_{j+1,k-1}}, & i-k+1 < j < i-r+2, \\ 1, & j \ge i-r+2, \end{cases}$$

and for $r \ge k$

$$\lambda_{j,k} = \begin{cases} 1, & j \le i-k, \\ 0, & j > i-k, \end{cases} \qquad \mu_{j,k} = \begin{cases} 0, & j \le i-k+1, \\ 1, & j > i-k+1, \end{cases}$$



(a) (b) (c)

Figure 2: Sample Refinement of $B_1(u,v)$. (a) T-mesh of the NUAT T-spline; (b) The blending function $B_1(u,v)$ in T-mesh; (c) After inserting (u_4,v_5), $B_1(u,v)$ is split into the linear combination of two scaled basis functions $B_1^1(u,v)$ and $B_2^1(u,v)$.

and

$$\lambda_{j,2}=\begin{cases}1, & j\le i-1,\\ \frac{\sin(t'-t_i)}{\sin(t_{i+1}-t_i)}, & j=i,\\ 0, & j\ge i+1,\end{cases}\qquad \mu_{j,2}=\begin{cases}0, & j\le i-1,\\ \frac{\sin(t_{i+1}-t')}{\sin(t_{i+1}-t_i)}, & j=i,\\ 1, & j\ge i+1,\end{cases}$$

where r is the multiplicity of the knot t' in T. $\lambda_{j,k}+\mu_{j,k}=1$ for all $j,k\ge 3$.

4.2 Blending function refinement

For a NUAT B-spline of order k, if $\mathbf{t}=[t_0,t_1,t_2,\cdots,t_k]$ is a knot vector and $\mathbf{t}^*$ is obtained by inserting a new knot t' into $\mathbf{t}$. The blending function corresponding to $\mathbf{t}$ is

$$N[\mathbf{t}](t)=\begin{cases}cN[\mathbf{t_1}](t)+dN[\mathbf{t_2}](t), & \mathbf{t}^*=[t_0,\cdots,t',\cdots,t_k],\\ N[\mathbf{t_2}](t), & \mathbf{t}^*=[t',t_0,t_1,\cdots,t_k],\\ N[\mathbf{t_1}](t), & \mathbf{t}^*=[t_0,t_1,\cdots,t_k,t'].\end{cases}$$

where $\mathbf{t_1}$ is the knot vector with the first $k+1$ knots of the knot vector $\mathbf{t}^*$ and $\mathbf{t_2}$ is the knot vector with the last $k+1$ knots of the knot vector $\mathbf{t}^*$. For example, if $\mathbf{t}^*=[t_0,\cdots,t',\cdots,t_k]$, then $\mathbf{t_1}=[t_0,\cdots,t',\cdots,t_{k-1}]$ and $\mathbf{t_2}=[t_1,\cdots,t',\cdots,t_k]$. The coefficients c and d can be determined by (6).

A NUAT T-spline blending function $B_i(u,v)$ of bi-degree (p,q) can undergo knot insertion operation in either $\mathbf{u}$ or $\mathbf{v}$, thereby splitting it into two scaled blending functions that sum to the initial one [2]. For example, Figure 2.a presents the original T-mesh of a NUAT T-spline of bi-degree $(3,5)$. It is shown in Figure 2.b that the blending function at (u_5,v_5) is $B_1(u,v)=N[u_2,u_3,u_5,u_6,u_7](u)N[v_2,v_3,v_4,v_5,v_6,v_7,v_8](v)$ in the original T-mesh. After inserting one knot (u_4,v_5), the blending function $B_1(u,v)$ is split into two scaled basis functions:

$$B_1(u,v)=c_1^1B_1^1(u,v)+c_2^1B_2^1(u,v)$$

where $B_1^1(u,v)=N[u_2,u_3,u_4,u_5,u_6](u)N[v_2,v_3,v_4,v_5,v_6,v_7,v_8](v)$ and we can get $B_2^1(u,v)=N[u_3,u_4,u_5,u_6,u_7](u)N[v_2,v_3,v_4,v_5,v_6,v_7,v_8](v)$. The above coefficients c_1^1 and c_2^1 can be determined according to (6) (Figure 2.c.).

4.3 Local refinement algorithm

The local refinement of NUAT T-splines means to insert control points into the T-mesh. Meanwhile, necessary knots should be inserted into neighboring blending functions. The refinement algorithm presented here is better than the T-spline local refinement algorithm because after the local refinement the new control grid is still a T-mesh as the initial one.

In a NUAT T-spline, the blending functions and the T-mesh are tightly coupled. Each control point corresponds to one blending function whose knot vectors are defined by KVD Rule. Blending functions will be refined during local refinement and new blending functions will be produced. This may cause violations, which means that during the process of the algorithm, there may be blending functions that violate KVD Rule. In fact, there are three possible violations that could occur during the course of the refinement algorithm.

- **Violation 1.** A blending function is missing a knot dictated by KVD Rule for the current T-mesh.
- **Violation 2.** Several blending functions correspond to one control point in the current T-mesh.
- **Violation 3.** L-node shared by one u-edge and one v -edge is in the T-mesh.

These possible violations must be eliminated. If Violation 1 occurs, the necessary knot insertion should be performed into that blending function and an appropriate control point should be added into the T-mesh. If Violation 2 occurs, the relevant blending functions should be refined according to the current T-mesh. When Violation 3 occurs, an appropriate control point should be added into the T-mesh accompanied by the corresponding knots to eliminate the L-node.

The local refinement algorithm for a NUAT T-spline of bi-degree (p, q) defined in (5) consists of the following steps:

1. Insert all desired control points and their corresponding knots into the T-mesh.
2. The blending functions of the original T-mesh are refined. Then check the new blending functions:
   (a) If any blending function is guilty of Violation 1, insert the missing knot and its corresponding control point into the T-mesh.
   (b) If Violation 2 occurs, those blending functions corresponding to one control point should be refined according to the current T-mesh.
3. If Violation 3 occurs, there is an L-node whose knot vectors are $\mathbf{u}^i$ and $\mathbf{v}^i$. Suppose $\mathbf{u}^i = [u^i_{j-\lceil p/2\rceil}, \cdots, u^i_j, \cdots, u^i_{j+\lceil p/2\rceil}]$ and $\mathbf{v}^i = [v^i_{k-\lceil q/2\rceil}, \cdots, v^i_k, \cdots, v^i_{k+\lceil q/2\rceil}]$. If (u^i_j, v^i_{k-1}) is not in the T-mesh, insert it into the T-mesh. Otherwise, insert (u^i_j, v^i_{k+1}) into the T-mesh. The knot pairs represent both control points and knots in the knot vector space.
4. Repeat Step 2 and Step 3 whenever a control point is added, until there are no more violations.

During the local refinement procedure, one important issue should be taken attention to. That is after the insertion of control points, some necessary edges according to Condition 3 for the T-mesh should be added into the control grid.

In fact, the local refinement occurs only in the active region of the T-mesh. The algorithm is illustrated with the following example. The active region of a NUAT T-spline of bi-degree $(3,5)$ is shown in Figure 3.a.

The NUAT T-spline is defined with the global knot vectors $\mathbf{u} = [u_{-1}, u_0, \cdots, u_8, u_9]$ and $\mathbf{v} = [v_{-2}, v_{-1}, \cdots, v_9, v_{10}]$. $[u_1, u_7] \times [v_1, v_7]$ is the active region of the T-mesh. In Figure 3.b, a control point $\mathbf{P}$ and its corresponding knot are inserted into the T-mesh. Since $\mathbf{P}$ has knot coordinate (u, v_4), the blending functions which centered at (u_3, v_4), (u_4, v_4), (u_5, v_4) and (u_6, v_4) are refined (Figure 3.c). The blending function centered at (u_4, v_4) is $N[u_2, u_3, u_4, u_5, u_6](u)N[v_1, v_2, v_3, v_4, v_5, v_6, v_7](v)$ in the initial T-mesh (Figure 3.d). Inserting a knot u into the $\mathbf{u}$ knot vector of this blending function, it is split into two scaled blending functions $N[u_2, u_3, u_4, u, u_5](u)N[v_1, v_2, v_3, v_4, v_5, v_6, v_7](v)$ and $N[u_3, u_4, u, u_5, u_6](u)N[v_1, v_2, v_3, v_4, v_5, v_6, v_7](v)$.

The blending function $N[u_2, u_3, u_4, u, u_5](u)N[v_1, v_2, v_3, v_4, v_5, v_6, v_7](v)$ in Figure 3.e satisfies KVD Rule. Likewise, the refinements of the blending functions centered at (u_3, v_4), (u_5, v_4) and (u_6, v_4) all satisfy KVD Rule. However, $N[u_3, u_4, u, u_5, u_6](u)N[v_1, v_2, v_3, v_4, v_5, v_6, v_7](v)$ is guilty of Violation 1 since it is missing a knot v_3 in its $\mathbf{v}$ knot vector dictated by KVD Rule for the current T-mesh in Figure 3.f. Therefore, we should insert the control point $\mathbf{P}_1$ and its corresponding knot (u, v_3) into the T-mesh. Then the blending functions which centered at (u_3, v_3) and (u_4, v_3) are refined. The refinements of the blending functions at (u_3, v_3) and (u_4, v_3) all satisfy KVD Rule. However, Violation 2 occurs since there are several blending functions corresponding to (u, v_4) in the current T-mesh, so the blending functions $N[u_3, u_4, u, u_5, u_6](u)N[v_{-1}, v_1, v_2, v_4, v_5, v_6, v_7](v)$ (Figure 3.h) should be refined according to the T-mesh.

There is an L-node $\mathbf{P}_1$ in the control grid, whose knot vectors are $\mathbf{u}^1 = [u_3, u_4, u, u_5, u_6]$ and $\mathbf{v}^1 = [v_0, v_1, v_2, v_3, v_4, v_5, v_6]$. (u, v_2) is not in the T-mesh, then insert it into the T-mesh. There are no violations for the T-mesh. The local refinement is finished meanwhile a new T-mesh is produced as shown in Figure 3.i. For this example, we can insert (u, v_2) in $\mathbf{v}$ direction or (u_5, v_3) in $\mathbf{u}$ direction to eliminate the L-node. In practice, we choose (u, v_2).

This algorithm is always guaranteed to terminate, because the blending function refinements and control point insertions must involve knot values that initially exist in the T-mesh, or that were added in Step 1. In the worst case, the algorithm would extend all partial rows of control points until the T-mesh turns to be a B-mesh [2]. In practice, the algorithm requires few additional new control points beyond the ones the user wants to insert. This algorithm is better than the local refinement algorithm of T-splines proposed by Sederberg et al. [2], whose resulting control grid after the local refinement may not be a T-mesh.

§5 The linear independence condition

A NUAT T-spline of odd bi-degree (p, q) is in homogeneous form:

$$\mathbf{T}(u, v) = \sum_{i=1}^{n_T} \mathbf{P}_i T_i(u, v). \tag{7}$$

All the control points are located in $[u_1, u_s] \times [v_1, v_t]$, and n_T is the number of the control points. $T_i(u, v) = N(\mathbf{u}^i)N(\mathbf{v}^i)$ are the blending functions of $\mathbf{P}_i$ which are homogeneous control points.

Through local refinement, the NUAT T-spline can be converted into its corresponding NUAT

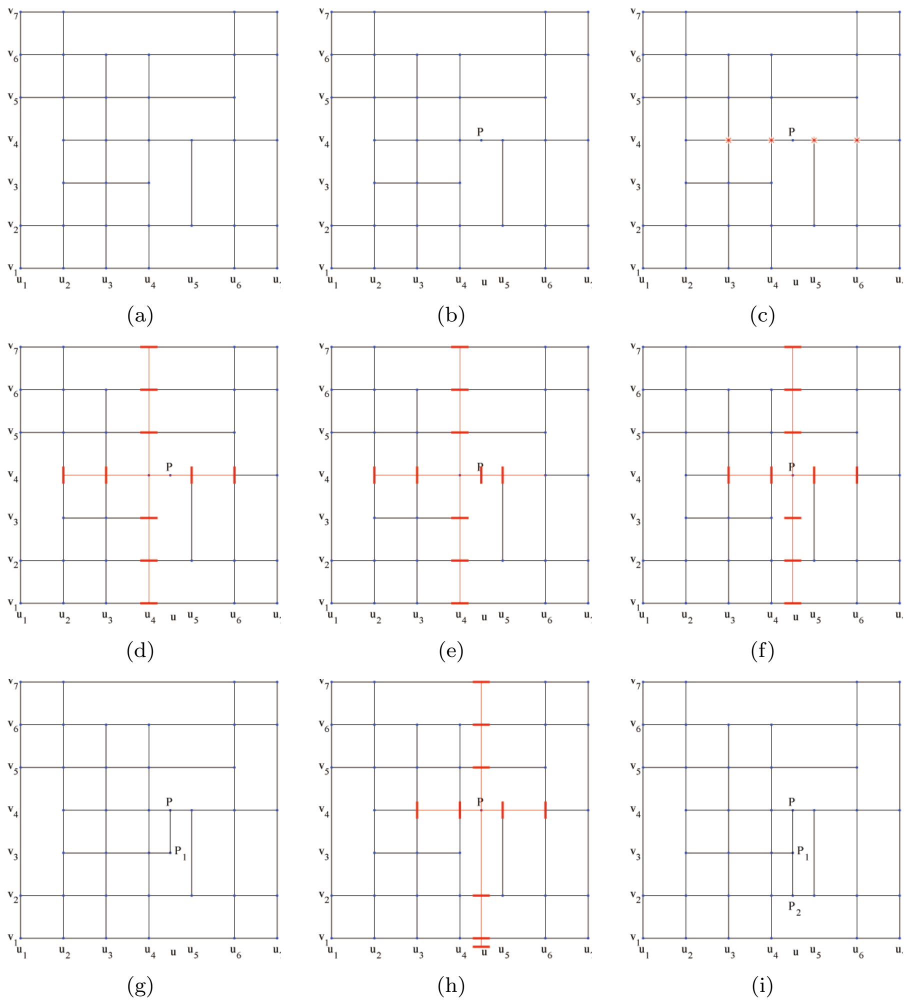


Figure 3: A local refinement example of NUAT T-spline with bi-order $(3, 5)$. (a) Original T-mesh of the NUAT T-spline; (b) Insert **P**; (c) After inserting **P**, the blending functions which centered at (u_3, v_4), (u_4, v_4), (u_5, v_4) and (u_6, v_4) are refined; (d) The blending function centered at (u_4, v_4) in original T-mesh is refined; (e) The new blending function centered at (u_4, v_4) after local refinement; (f) The new blending function centered at (u, v_4) after local refinement in current T-mesh is missing a knot; (g) Insert a missing knot (u, v_3) and necessary edges; (h) One blending function corresponding to (u, v_4) does not satisfy the current T-mesh; (i) The resulting T-mesh after local refinement.

B-spline surface. The NUAT B-spline surface can be described in homogeneous form:

$$\mathbf{B}(u,v)=\sum_{i=1}^{s}\sum_{j=1}^{t}\hat{\mathbf{Q}}_{ij}\hat{B}_{ij}(u,v) \tag{8}$$

where $\hat{B}_{ij}(u,v)=N_{i,p+1}(u)N_{j,q+1}(v)(i=1,2,\cdots,s;\ j=1,2,\cdots,t)$ are the basis functions of $\hat{\mathbf{Q}}_{ij}$, and $\hat{\mathbf{Q}}_{ij}$ are the control points in the B-mesh. The NUAT B-spline surface is a NUAT T-spline, so the surface can be described as:

$$\mathbf{B}(u,v)=\sum_{j=1}^{n_B}\mathbf{Q}_jB_j(u,v), n_B=st \tag{9}$$

where $\mathbf{Q}_j$ are the control points of the surface, and $B_j(u,v)$ are the blending functions of $\mathbf{Q}_j$. $n_B=st$ is the number of the control points in the B-mesh.

Each NUAT T-spline blending function can be written as a linear combination of its corresponding NUAT B-spline basis functions:

$$T_i(u,v)=\sum_{j=1}^{n_B}c_j^iB_j(u,v), i=1,2,\cdots,n_T. \tag{10}$$

Denote $\begin{pmatrix} c_1^1 & c_2^1 & \cdots & c_{n_B}^1 \\ c_1^2 & c_2^2 & \cdots & c_{n_B}^2 \\ \vdots & \vdots & & \vdots \\ c_1^{n_T} & c_2^{n_T} & \cdots & c_{n_B}^{n_T} \end{pmatrix}=C$, where C is called the NUAT T-spline-to-NUAT B-spline transformation matrix. Then (10) can be written in matrix form:

$$(T_1(u,v),T_2(u,v),\cdots,T_{n_T}(u,v))^{\mathrm{T}}=C(B_1(u,v),B_2(u,v),\cdots,B_{n_B}(u,v))^{\mathrm{T}}.$$

The knot insertion does not change the geometry of the NUAT T-spline which means $\mathbf{T}(u,v)=\mathbf{B}(u,v)$. So the relationship between the NUAT T-spline control points and the NUAT B-spline control points can be also obtained by

$$C^{\mathrm{T}}\begin{pmatrix}\mathbf{P}_1\\ \mathbf{P}_2\\ \vdots\\ \mathbf{P}_{n_T}\end{pmatrix}=\begin{pmatrix}\mathbf{Q}_1\\ \mathbf{Q}_2\\ \vdots\\ \mathbf{Q}_{n_B}\end{pmatrix}.$$

The above formula can be written as $C^{\mathrm{T}}\mathbb{P}=\mathbb{Q}$, where $\mathbb{P}=(\mathbf{P}_1,\mathbf{P}_2,\cdots,\mathbf{P}_{n_T})^{\mathrm{T}}$ and $\mathbb{Q}=(\mathbf{Q}_1,\mathbf{Q}_2,\cdots,\mathbf{Q}_{n_B})^{\mathrm{T}}$.

Theorem 5.1. *The NUAT T-spline-to-NUAT B-spline transformation matrix C for any NUAT T-spline of odd bi-degree surface is unique. Furthermore, the necessary and sufficient condition for the blending functions of any NUAT T-spline with odd bi-degree to be linearly independent is that C is of full rank.*

Proof. The NUAT T-spline is defined in formula (7). Referring to (10), since the $B_j(u,v)$ are NUAT B-spline basis functions, each column of C is unique, so C is also unique.

By definition, the blending functions of the NUAT T-spline are linearly independent if and only if there do not exist constants k_i, not all zero, such that

$$k_1T_1(u,v)+k_2T_2(u,v)+\cdots+k_{n_T}T_{n_T}(u,v)=(k_1,k_2,\cdots,k_{n_T})\begin{pmatrix}T_1(u,v)\\ T_2(u,v)\\ \vdots\\ T_{n_T}(u,v)\end{pmatrix}=0. \tag{11}$$

The NUAT T-spline blending functions are the linear combinations of the NUAT B-spline blending functions, then the linear independence requires

$$(k_1, k_2, \cdots, k_{n_T}) \begin{pmatrix} T_1(u,v) \\ T_2(u,v) \\ \vdots \\ T_{n_T}(u,v) \end{pmatrix} = (k_1, k_2, \cdots, k_{n_T}) C \begin{pmatrix} B_1(u,v) \\ B_2(u,v) \\ \vdots \\ B_{n_B}(u,v) \end{pmatrix} = 0. \quad (12)$$

Since $B_j(u,v)$ is a basis, the necessary and sufficient condition for linear dependence of the NUAT T-spline blending functions becomes

$$C^{\mathrm{T}} \begin{pmatrix} k_1 \\ k_2 \\ \vdots \\ k_{n_T} \end{pmatrix} = 0 \quad (13)$$

for k_i, which are not all zero. This will happen if and only if C satisfies $Rank(C) = n_{n_T}$. It means that when C is full rank the blending functions of the NUAT T-spline are linearly independent. Then the blending functions are basis functions. □

§6 Conclusion

In this paper, our main contributions can be stated as follows: First, we present the definition of the NUAT T-splines of odd bi-degree. Second, a local refinement algorithm for the NUAT T-splines is proposed. This algorithm guarantees that the resulting control grid is a T-mesh. It is better than the local refinement algorithm of T-splines proposed by Sederberg et al. [4]. At last but not at least, a necessary and sufficient condition for the blending functions of the NUAT T-spline to be linearly independent is given. In fact, NUAT T-splines of arbitrary degree can be defined. But the definition of even bi-degree and mixed degree NUAT T-splines are quite different from the odd bi-degree ones. Besides, the knot insertions and the control points adding are not simultaneous in the local refinement for these types. These types of NUAT T-splines remain to be investigated.

There are some other transcendental curves which cannot be represented exactly by NUAT T-splines. In order to give exact representations of these curves, NUAH B-splines and UE-splines can be used [8-9]. Applying the T-spline framework to these splines is of interest. On the other hand, the application of NUAT T-splines in isogeometric analysis [11-12] is also one part of our future work.

Acknowledgements. The authors would like to thank the referees and Dr. Xu Gang for many valuable comments and suggestions which greatly improved the presentation of this paper.

References

[1] T W Sederberg, J Zheng, A Bakenov, A Nasri. *T-splines and T-NURCCs*, ACM Trans Graph, 2003, 22(3): 477-484.

[2] T W Sederberg, D L Cardon, G T Finnigan, N S North, J Zheng, T Lyche. *T-spline simplification and local refinement*, ACM Trans Graph, 2004, 23: 276-283.

[3] J W Zhang. *C-curves: an extension of cubic curves*, Comput Aided Geom Design, 1996, 13: 199-217.

[4] J W Zhang. *Two different forms of C-B-splines*, Comput Aided Geom Design, 1997, 14: 31-41.

[5] G Z Wang, Q Y Chen, M H Zhou. *NUAT B-spline curves*, Comput Aided Geom Design, 2004, 21: 193-205.

[6] X Li, J Zheng, T W Sederberg, T G R Hughes, M A Scott. *On linear independence of T-spline blending functions*, Comput Aided Geom Design, 2012, 29: 63-76.

[7] G T Finnigan. *Arbitrary degree T-splines*, Master's Thesis, Brigham Young University, 2008.

[8] Y Li, G Wang. *Two kinds of B-basis of the algebraic hyperbolic space*, J Zhejiang University, 2005, 6(A): 750-759.

[9] Y Lü, G Wang, X Yang. *Uniform hyperbolic polynomial B-spline curves*, Comput Aided Geom Design, 2002, 19(6): 379-393.

[10] Y Bazilevs, V M Calo, J A Cottrell, J Evans, T J R Hughes, S Lipton, M A Scott, T W Sederberg. *Isogeometric analysis using T-splines*, Comput Methods Appl Mech Engrg, 2010, 199: 229-263.

[11] G Xu, B Mourrain, R Duvigneau, A Galligo. *Parameterization of computational domain in isogeometric analysis: methods and comparison*, Comput Methods Appl Mech Engrg, 2011, 200: 2021-2031.

[12] G Xu, B Mourrain, R Duvigneau, A Galligo. *Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications*, Comput Aided Design, 2013, 45: 395-404.

Department of Mathematics, Zhejiang University, Hangzhou 310027, China.
Email: heng-juan0721@163.com