SCIENCE CHINA
Information Sciences

• Supplementary File •

Supporting Information on the Article "Trivariate B-spline Solid Construction by Pillow Operation and Geometric Iterative Fitting"

Hongwei LIN[1*], Hao HUANG[1] & Chuanfeng HU[1]

[1]*School of Mathematical Science, State Key Lab. of CAD&CG, Zhejiang University, Hangzhou* 310058, *China*

1 Introduction

While the traditional geometric design community focuses on the design of curves and surfaces, the advent of Isogeometric Analysis (IGA) [1] has made the development of methods for designing Trivariate B-spline Solids (TBSs) imperative. In IGA, a *valid* TBS should have a positive Jacobian value at every point in its domain. A negative Jacobian value at any point in the domain of the TBS can render the IGA invalid.

As is widely recognized, the generation of a valid TBS is difficult, owing to two key factors:

- The geometric condition for ensuring that a TBS is valid, i.e., the Jacobian value at every point is positive, is highly nonlinear. Hence, it is theoretically very difficult to guarantee the *validity* of a TBS. The state-of-the-art methods for generating TBS usually transform the validity problem into an constrained optimization problem. However, the optimization is prone to fail owing to the high nonlinearity of the objective function and high complexity of constraints. Therefore, to generate valid TBSs, the high nonlinearity of the objective function and high complexity of constraints should be reduced and an efficient method should be developed for solving the constrained optimization problem.
- The regions with negative Jacobian are usually concentrated near the boundary curves where adjacent surfaces are smoothly stitched (refer to Fig. 6). To eliminate negative Jacobian values, the input model should be segmented along the boundary curves.

In this paper, we developed a method that can generate a TBS with a guaranteed positive Jacobian, if the initial TBS is valid. Using a tetrahedral (tet) mesh with six surfaces segmented on its boundary mesh as the input, we first partition the tet mesh model into seven sub-volumes using the *pillow* operation [2], a method originally developed for improving the quality of hexahedral meshes. After each of them is parameterized into a cubic parameter domain, seven initial valid TBSs are constructed. Moreover, starting with the initial valid TBSs, the boundary curves, boundary surfaces and the TBSs are fitted by a geometric iterative fitting algorithm, known as the Geometric Feasible Direction (GFD) algorithm. In each iteration of the GFD algorithm, the movements of the control points are restricted inside a feasible region to ensure the validity. Finally, the smoothness between two adjacent TBSs is improved by the GFD algorithm. In this way, the validity of the generated TBSs is guaranteed with desirable smoothness between adjacent TBSs.

[*] Corresponding author (email: hwlin@zju.edu.cn)

The structure of this paper is as follows: Section 1.1 provides a survey on related work. Section 2 presents an overview of the proposed method. In Section 3, we introduce the validity conditions for B-spline curves, surfaces, and TBSs, the geometric continuity between TBSs, and the pillow operation. In Section 4, the Geometric Feasible Direction (GFD) algorithm is developed. After elucidating the details of the developed method in Section 5, the experimental results are illustrated in Section 6. Lastly, Section 7 concludes the paper.

1.1 Related work

In this section, work related to TBS modeling and geometric iterative fitting is briefly reviewed.

TBS modeling: To analyze the arterial blood flow by IGA, a trivariate NURBS-solid modeling the artery was constructed using a skeleton-based method [3]. Following volume parameterization by a harmonic function, a cylinder-like trivariate B-spline solid was generated with a singular centric curve [4]. Moreover, trivariate B-spline solids with positive Jacobian values were produced from boundary representations using optimization-based approaches [5,6]. However, the optimization method may fail due to the highly nonlinear objective function and highly complicated constraints. Based on the given boundary conditions and guiding curves, a NURBS solid was constructed to model a swept volume by the variational approach [7]. After an automatic and robust pants-to-cuboids decomposition, surface and volume parameterizations are generated in order to fit the trivariate splines, based on a discrete harmonic mapping between the cuboids' boundary and a polycube's boundary [8]. In addition, using the volumetric parametrization provided by a hierarchical basis of C^1 cubic polynomials, a fast and automated method is proposed to get a spline-based model from voxel data in [9]. In [10], a C^2-continuous seven-directional box spline based framework was developed for the parametrization of volumetric data with high accuracy.

Moreover, trivariate T-spline solids are employed in IGA owing to their adaptive refinement capability [11]. To fill a genus-zero triangular mesh, a T-spline solid was constructed after mesh untangling and smoothing [12]. Starting from a boundary surface triangulation with genus-zero, Zhang et al. developed a mapping-based method to generate rational trivariate solid T-splines [13]. Furthermore, to fill a boundary triangulation with arbitrary genus, a polycube mapping-based algorithm for constructing T-spline solids was proposed in [14]. In [15], a volumetric T-spline was constructed for filling a B-rep model using Boolean operations, polycube mapping, and octree subdivision. The inputs for the aforementioned methods are triangular meshes. When the input was a genus-zero T-spline surface model, Zhang et al. constructed a solid T-spline whose boundary exactly conformed to the given T-spline surface model [16].

On the other hand, Catmull-Clark subdivision solids were developed to model the computational domain in IGA [17]. Additionally, other representations for spline solids include simplex splines [18] and polycube splines [19,20].

The methods described above usually generate a trivariate solid to fill a given B-rep model. However, the generation of a TBS by fitting a tet mesh model is much easier than by filling a B-rep model. Because it is very easy to produce a tet mesh using popular software, such as, TetGen [21], and NetGen [22], it is feasible to generate a TBS by fitting a tet mesh model. In Ref. [23], a tet mesh model is fitted by the geometric iterative method to generate a TBS. However, there are some regions with negative Jacobian values close to the boundary. In this paper, a tet mesh is first segmented into seven sub-volumes, each of which is fitted with a TBS by a geometric iterative fitting method. In this way, the generated TBSs are ensured to be valid, i.e. the Jacobian value at any point of the TBSs is positive.

In conclusion, the representations of a trivariate solid include B-spline solid, NURBS solid, T-spline solid, subdivision solid, etc. The input for a solid generation method is either a boundary representation, such as a B-spline surface, a triangular mesh, or a volume mesh, such as a tet mesh. In our method, the input model is segmented to generate the valid TBS. While tet meshes are easy to be partitioned into sub-volumes, the boundary representations are hard to be segmented owing to the loss of inner information. Therefore, tet meshes are taken as inputs for our method.

Geometric iterative fitting: Geometric Iterative Fitting (GIF), also called Progressive-Iterative Approximation (PIA), was first developed in [24,25]. The GIF method endows iterative methods with geometric meanings, facilitating the handling of geometric problems appearing in the field of geometric design. It was proved that the GIF method is convergent for B-spline fitting [26,27], NURBS fitting [28], T-spline fitting [29], subdivision surface fitting [30–32], as well as curve and surface fitting with totally positive basis [25]. Moreover, the GIF methods have been employed in some applications such as reverse engineering [33,34], curve design [35], surface-surface intersection [36], etc. For more information on the GIF methods, please refer to the survey paper [37]. In this paper, the GIF method is used to fit several sub-volumes with valid TBSs.

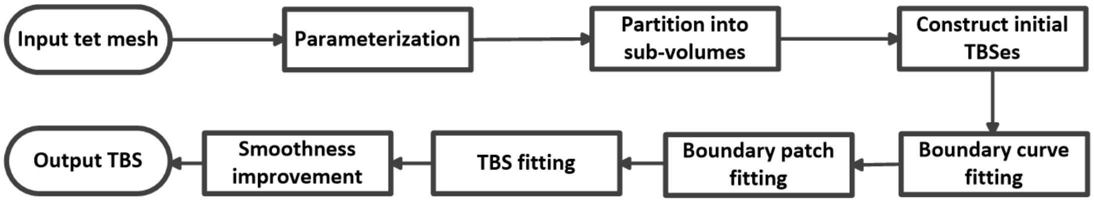


Figure 1 Overview of the TBS generation algorithm developed in this paper.

2 Algorithm overview

The TBS generation algorithm is demonstrated in Fig. 1. The input to this algorithm is a tet mesh with six boundary surfaces (Fig. 5(a)). The tet mesh is first parameterized into a cubic parametric domain. Then, the cubic domain is segmented into seven sub-domains, as shown in Fig. 5(b). By mapping the sub-domains into the input tet mesh, it is divided into seven sub-volumes. After the initial TBSs are constructed, the boundary curves, boundary surfaces, and the seven TBSs are respectively fitted using the GFD algorithm. Finally, the smoothness between adjacent TBSs is improved. In subsequent sections, the details of the TBS generation algorithm will be elucidated.

3 Validity conditions, geometric continuity between TBSs, and pillow operation

In this paper, what we want to generate is a composition of valid TBSs with as desirable as possible smoothness between adjacent TBSs. So, the validity conditions and geometric continuity definition between TBSs should be clarified. Moreover, the pillow operation is also introduced in this section.

3.1 Validity conditions

In this section, the validity conditions for a B-spline curve, surface, and TBS are presented, respectively.

Definition 1. 1) A B-spline curve is valid, if its derivative vector is nonzero at any parameter in its domain;

2) A B-spline surface is valid, if its normal vector is nonzero at any point in its parametric domain;

3) A TBS $\boldsymbol{H}(u,v,w)$ is valid, if $\boldsymbol{H}_u \cdot (\boldsymbol{H}_v \times \boldsymbol{H}_w) \neq 0$ at any point in its parametric domain.

Given a B-spline curve of degree d with control points $\boldsymbol{P}_i, i = 0,1,\cdots,m$,

$$\boldsymbol{C}(u) = \sum_{i=0}^{m} \boldsymbol{P}_i B_i^d(u), \tag{1}$$

where $B_i^d(u),\ i = 0,1,\cdots,m$ are B-spline basis functions of degree d defined on some knot vector, and denoting the difference vectors as,

$$\boldsymbol{T}_i = \frac{\boldsymbol{P}_{i+1} - \boldsymbol{P}_i}{\|\boldsymbol{P}_{i+1} - \boldsymbol{P}_i\|},\ i = 0,1,\cdots,m-1, \tag{2}$$

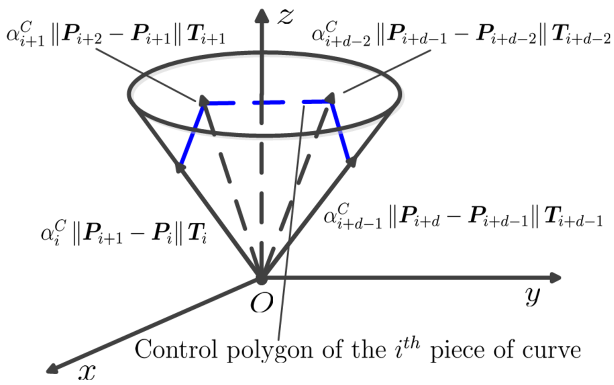

$\alpha_{i+1}^C \|\boldsymbol{P}_{i+2}-\boldsymbol{P}_{i+1}\| \boldsymbol{T}_{i+1}$
z
$\alpha_{i+d-2}^C \|\boldsymbol{P}_{i+d-1}-\boldsymbol{P}_{i+d-2}\| \boldsymbol{T}_{i+d-2}$
$\alpha_{i+d-1}^C \|\boldsymbol{P}_{i+d}-\boldsymbol{P}_{i+d-1}\| \boldsymbol{T}_{i+d-1}$
$\alpha_i^C \|\boldsymbol{P}_{i+1}-\boldsymbol{P}_i\| \boldsymbol{T}_i$
O
y
Control polygon of the i^{th} piece of curve
x

Figure 2 If the apertures of the minimum circular cones enclosing the difference vectors (2) $\{\boldsymbol{T}_i, \boldsymbol{T}_{i+1}, \cdots, \boldsymbol{T}_{i+d-1}\}$, $i = 0, 1, \cdots, m-d$, respectively, are all less than π, the B-spline curve of degree d is valid.

the validity condition for the B-spline curve (1) is:

Proposition 1 (Validity condition for B-spline curves)**.** A B-spline curve of degree d is valid if the apertures of the minimum circular cones enclosing the difference vectors (2)

$$\{\boldsymbol{T}_i, \boldsymbol{T}_{i+1}, \cdots, \boldsymbol{T}_{i+d-1}\}, \ i = 0, 1, \cdots, m-d,$$

respectively, are all less than π.

Proof: The derivative of the B-spline curve (1) is,

$$\boldsymbol{C}'(u) = \sum_{i=0}^{m-1} \alpha_i^C (\boldsymbol{P}_{i+1} - \boldsymbol{P}_i) B_i^{d-1}(u) = \sum_{i=0}^{m-1} \alpha_i^C \|\boldsymbol{P}_{i+1} - \boldsymbol{P}_i\| \boldsymbol{T}_i B_i^{d-1}(u),$$

where $\alpha_i^C > 0$ is related to the knot vector of $\boldsymbol{C}(u)$ (1).

Because $\boldsymbol{C}'(u)$ is a B-spline curve of degree $d-1$, the d control points,

$$\left\{\alpha_i^C \|\boldsymbol{P}_{i+1} - \boldsymbol{P}_i\| \boldsymbol{T}_i, \ \alpha_{i+1}^C \|\boldsymbol{P}_{i+2} - \boldsymbol{P}_{i+1}\| \boldsymbol{T}_{i+1}, \ \cdots, \ \alpha_{i+d-1}^C \|\boldsymbol{P}_{i+d} - \boldsymbol{P}_{i+d-1}\| \boldsymbol{T}_{i+d-1}\right\} \tag{3}$$

determine the i^{th} piece of curve, $i = 0, 1, \cdots, m-d$. In Fig. 2, if the aperture of the minimum circular cone enclosing the d control points is less than π, the i^{th} piece of curve, $i = 0, 1, \cdots, m-d$, of $\boldsymbol{C}'(u)$ will not pass the origin, due to the convex hull property of B-spline curve. Therefore, the derivative vector of $\boldsymbol{C}(u)$ (1) is nonzero at any parameter in its domain. According to Definition 1, the B-spline curve $\boldsymbol{C}(u)$ (1) is valid.

Furthermore, because

$$\alpha_i^C \|\boldsymbol{P}_{i+1} - \boldsymbol{P}_i\| > 0, i = 0, 1, \cdots, m-d,$$

the minimum circular cones enclosing the the d control points (3) is the same as that enclosing the difference vectors (refer to Eq. (2) and Fig. 2)

$$\{\boldsymbol{T}_i, \boldsymbol{T}_{i+1}, \cdots, \boldsymbol{T}_{i+d-1}\}, \ i = 0, 1, \cdots, m-d.$$

Proposition 1 is proved. □

Moreover, suppose we are given a B-spline surface of degree $d_u \times d_v$ with control points

$$\boldsymbol{S}_{ij}, i = 0, 1, \cdots, m, \ j = 0, 1, \cdots, n,$$

and denote the difference vectors as

$$\boldsymbol{T}_{ij}^u = \frac{\boldsymbol{S}_{i+1,j} - \boldsymbol{S}_{ij}}{\|\boldsymbol{S}_{i+1,j} - \boldsymbol{S}_{ij}\|}, \ i = 0, 1, \cdots, m-1, j = 0, 1, \cdots, n, \tag{4}$$

and,

$$\boldsymbol{T}^v_{kl} = \frac{\boldsymbol{S}_{k,l+1} - \boldsymbol{S}_{kl}}{\|\boldsymbol{S}_{k,l+1} - \boldsymbol{S}_{kl}\|},\ k = 0, 1, \cdots, m, l = 0, 1, \cdots, n-1. \tag{5}$$

Let $\mathcal{M}_{IJ}, I = 0, 1, \cdots, m - d_u, J = 0, 1, \cdots, n - d_v$ be the sub-control-polygon constituted by the control points,

$$\begin{matrix} \boldsymbol{S}_{IJ} & \boldsymbol{S}_{I,J+1} & \cdots & \boldsymbol{S}_{I,J+d_v} \\ \boldsymbol{S}_{I+1,J} & \boldsymbol{S}_{I+1,J+1} & \cdots & \boldsymbol{S}_{I+1,J+d_v} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{S}_{I+d_u,J} & \boldsymbol{S}_{I+d_u,J+1} & \cdots & \boldsymbol{S}_{I+d_u,J+d_v} \end{matrix} \tag{6}$$

Moreover, suppose $\boldsymbol{U}_{IJ}$ and $\boldsymbol{V}_{IJ}$ are the unit axis vectors of the minimum circular cones $\mathcal{C}^u_{IJ}$ and $\mathcal{C}^v_{IJ}$ enclosing the difference vectors $\boldsymbol{T}^u_{ij}$ (4) and $\boldsymbol{T}^v_{kl}$ (5) of the sub-control-polygon $\mathcal{M}_{IJ}$, starting from the apexes of cones, respectively. Then, a sufficient condition for the validity of a B-spline surface is presented as follows:

Proposition 2 (Validity condition for B-spline surfaces)**.** If $\boldsymbol{T}^u_{ij} \cdot \boldsymbol{U}_{IJ} > \boldsymbol{T}^u_{ij} \cdot \boldsymbol{V}_{IJ} \geqslant 0$, and, $\boldsymbol{T}^v_{kl} \cdot \boldsymbol{V}_{IJ} > \boldsymbol{T}^v_{kl} \cdot \boldsymbol{U}_{IJ} \geqslant 0$, where $\boldsymbol{T}^u_{ij}$ and $\boldsymbol{T}^v_{kl}$ are defined on each sub-control-polygon $\mathcal{M}_{IJ}$, $I = 0, 1, \cdots, m - d_u, J = 0, 1, \cdots, n - d_v$, the B-spline surface is valid.

Proof: Because,

$$\boldsymbol{T}^u_{ij} \cdot \boldsymbol{U}_{IJ} > \boldsymbol{T}^u_{ij} \cdot \boldsymbol{V}_{IJ} \geqslant 0, \text{ and, } \boldsymbol{T}^v_{kl} \cdot \boldsymbol{V}_{IJ} > \boldsymbol{T}^v_{kl} \cdot \boldsymbol{U}_{IJ} \geqslant 0,$$

we have,

$$(\boldsymbol{T}^u_{ij} \times \boldsymbol{T}^v_{kl}) \cdot (\boldsymbol{U}_{IJ} \times \boldsymbol{V}_{IJ}) = (\boldsymbol{T}^u_{ij} \cdot \boldsymbol{U}_{IJ})(\boldsymbol{T}^v_{kl} \cdot \boldsymbol{V}_{IJ}) - (\boldsymbol{T}^u_{ij} \cdot \boldsymbol{V}_{IJ})(\boldsymbol{T}^v_{kl} \cdot \boldsymbol{U}_{IJ}) > 0.$$

It means that, all of the vectors $\boldsymbol{T}^u_{ij} \times \boldsymbol{T}^v_{kl}$ of the sub-control-polygon $\mathcal{M}_{IJ}$ are on the same side of the plane with $\boldsymbol{U}_{IJ} \times \boldsymbol{V}_{IJ}$ as the normal vector. It makes the normal vector of the given B-spline surface nonzero at any point in its domain. So, the given B-spline surface is valid. □

Remark 1. Note that, the conditions $\boldsymbol{T}^u_{ij} \cdot \boldsymbol{U}_{IJ} > \boldsymbol{T}^u_{ij} \cdot \boldsymbol{V}_{IJ} \geqslant 0$, and, $\boldsymbol{T}^v_{kl} \cdot \boldsymbol{V}_{IJ} > \boldsymbol{T}^v_{kl} \cdot \boldsymbol{U}_{IJ} \geqslant 0$, mean that the two minimum circular cones $\mathcal{C}^u_{IJ}$ and $\mathcal{C}^v_{IJ}$ are separate. According to the definition of normal vector, the closer to orthogonality the two unit axis vectors $\boldsymbol{U}_{IJ}$ and $\boldsymbol{V}_{IJ}$ of the circular cones $\mathcal{C}^u_{IJ}$ and $\mathcal{C}^v_{IJ}$, the better the validity of the B-spline surface.

Similarly, we can develop a sufficient condition for determining the validity of a TBS $\boldsymbol{H}(u, v, w)$ of degree $d_u \times d_v \times d_w$,

$$\boldsymbol{H}(u, v, w) = \sum_i \sum_j \sum_k \boldsymbol{H}_{ijk} B^{d_u}_i(u) B^{d_v}_j(v) B^{d_w}_k(w),$$

where $B^{d_u}_i(u), B^{d_v}_j(v)$, and $B^{d_w}_k(w)$ are B-spline basis functions of degrees d_u, d_v, and d_w, respectively, and,

$$\boldsymbol{H}_{ijk}, i = 0, 1, \cdots, m, j = 0, 1, \cdots, n, k = 0, 1, \cdots, l,$$

are control points. Denote the difference vectors as

$$\boldsymbol{T}^u_{ijk} = \frac{\boldsymbol{H}_{i+1,j,k} - \boldsymbol{H}_{ijk}}{\|\boldsymbol{H}_{i+1,j,k} - \boldsymbol{H}_{ijk}\|},\ \boldsymbol{T}^v_{ijk} = \frac{\boldsymbol{H}_{i,j+1,k} - \boldsymbol{H}_{ijk}}{\|\boldsymbol{H}_{i,j+1,k} - \boldsymbol{H}_{ijk}\|},\ \boldsymbol{T}^w_{ijk} = \frac{\boldsymbol{H}_{i,j,k+1} - \boldsymbol{H}_{ijk}}{\|\boldsymbol{H}_{i,j,k+1} - \boldsymbol{H}_{ijk}\|}. \tag{7}$$

Moreover, letting

$$\mathcal{G}_{IJK},\ I = 0, 1, \cdots, m - d_u,\ J = 0, 1, \cdots, n - d_v,\ K = 0, 1, \cdots, l - d_w, \tag{8}$$

be the sub-grid constituted by the control points

$$\boldsymbol{H}_{ijk},\ i = I, I+1, \cdots, I + d_u,\ j = J, J+1, \cdots, J + d_v,\ k = K, K+1, \cdots, K + d_w,$$

we have,

Proposition 3 (Validity condition for TBSs). If

$$\boldsymbol{T}^u_{i_u j_u k_u} \cdot (\boldsymbol{T}^v_{i_v j_v k_v} \times \boldsymbol{T}^w_{i_w j_w k_w}) > 0,$$

where $\boldsymbol{T}^u_{i_u j_u k_u}, \boldsymbol{T}^v_{i_v j_v k_v}$ and $\boldsymbol{T}^w_{i_w j_w k_w}$ are defined on each sub-grid $\mathcal{G}_{IJK}$, the TBS $\boldsymbol{H}(u,v,w)$ is valid.

Proof: The Jacobian value of the TBS $\boldsymbol{H}(u,v,w)$ at (u,v,w) is,

$$\begin{aligned} J(u,v,w) &= \boldsymbol{H}_u(u,v,w) \cdot (\boldsymbol{H}_v(u,v,w) \times \boldsymbol{H}_w(u,v,w)) \\ &= \sum_{\mathcal{I}_u} \sum_{\mathcal{I}_v} \sum_{\mathcal{I}_w} \alpha^H_{\mathcal{I}_u \mathcal{I}_v \mathcal{I}_w} \left[\boldsymbol{T}^u_{i_u j_u k_u} \cdot (\boldsymbol{T}^v_{i_v j_v k_v} \times \boldsymbol{T}^w_{i_w j_w k_w})\right] B_{\mathcal{I}_u}(u) B_{\mathcal{I}_v}(v) B_{\mathcal{I}_w}(w), \end{aligned}$$

where $\mathcal{I}_u = (i_u, i_v, i_w), \mathcal{I}_v = (j_u, j_v, j_w), \mathcal{I}_w = (k_u, k_v, k_w)$ are index sets, $\alpha_{\mathcal{I}_u \mathcal{I}_v \mathcal{I}_w} > 0$, and,

$$\begin{aligned} &B_{\mathcal{I}_u}(u) = B^{d_u-1}_{i_u}(u) B^{d_u}_{i_v}(u) B^{d_u}_{i_w}(u), B_{\mathcal{I}_v}(v) = B^{d_v}_{j_u}(v) B^{d_v-1}_{j_v}(v) B^{d_v}_{j_w}(w), \\ &B_{\mathcal{I}_w}(w) = B^{d_w}_{k_u}(w) B^{d_w}_{k_v}(v) B^{d_w-1}_{k_w}(w). \end{aligned}$$

Moreover, because the local support property of TBS, the Jacobian value $J(u,v,w)$ is determined by one of the sub-grid $\mathcal{G}_{IJK}$ (8). Therefore, if $\boldsymbol{T}^u_{i_u j_u k_u} \cdot (\boldsymbol{T}^v_{i_v j_v k_v} \times \boldsymbol{T}^w_{i_w j_w k_w}) > 0$, where $\boldsymbol{T}^u_{i_u j_u k_u}, \boldsymbol{T}^v_{i_v j_v k_v}$ and $\boldsymbol{T}^w_{i_w j_w k_w}$ are defined on the sub-grid $\mathcal{G}_{IJK}$, it follows that

$$\boldsymbol{H}_u(u,v,w) \cdot (\boldsymbol{H}_v(u,v,w) \times \boldsymbol{H}_w(u,v,w)) > 0,$$

meaning that, the TBS $\boldsymbol{H}(u,v,w)$ is valid. □

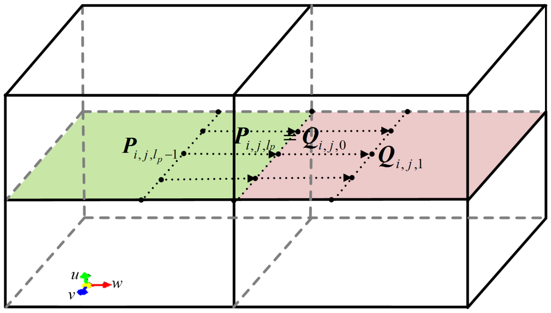

$\boldsymbol{P}_{i,j,l_p-1}$
$\boldsymbol{P}_{i,j,l_p} = \boldsymbol{Q}_{i,j,0}$
$\boldsymbol{Q}_{i,j,1}$
u
v
w

Figure 3 Geometric continuous condition between adjacent TBSs.

3.2 Geometric continuity between TBSs

Given two adjacent TBSs (Fig. 3),

$$\boldsymbol{P}(u,v,w) = \sum_{i_p=0}^{m_p} \sum_{j_p=0}^{n_p} \sum_{k_p=0}^{l_p} \boldsymbol{P}_{i_p,j_p,k_p} B_{i_p}(u) B_{j_p}(v) B_{k_p}(w),\ (u,v,w) \in [0,1] \times [0,1] \times [0,1], \tag{9}$$

and

$$\boldsymbol{Q}(\mu,\nu,\omega) = \sum_{i_q=0}^{m_q} \sum_{j_q=0}^{n_q} \sum_{k_q=0}^{l_q} \boldsymbol{Q}_{i_q,j_q,k_q} B_{i_q}(\mu) B_{j_q}(\nu) B_{k_q}(\omega),\ (\mu,\nu,\omega) \in [0,1] \times [0,1] \times [0,1], \tag{10}$$

where, $\boldsymbol{P}_{i_p,j_p,k_p}$ and $\boldsymbol{Q}_{i_q,j_q,k_q}$ are control points, $B_{i_p}(u)$, $B_{j_p}(v)$, $B_{k_p}(w)$, $B_{i_q}(\mu)$, $B_{j_q}(\nu)$, and $B_{k_q}(\omega)$ are B-spline basis, the geometric continuity between the two TBSs $\boldsymbol{P}(u,v,w)$ and $\boldsymbol{Q}(\mu,\nu,\omega)$ is defined as,

Definition 2 (Geometric Continuity). The two TBSs $\boldsymbol{P}(u,v,w)$ and $\boldsymbol{Q}(\mu,\nu,\omega)$ are G^n geometric continuous along the common boundary surface $\boldsymbol{P}(u,v,1) = \boldsymbol{Q}(\mu,\nu,0)$, if,

1) the two B-spline surfaces $\boldsymbol{P}(c,v,w)$ and $\boldsymbol{Q}(c,\nu,\omega)$ are G^n geometric continuous along the common boundary curve $\boldsymbol{P}(c,v,1) = \boldsymbol{Q}(c,\nu,0)$, and,

2) the two B-spline surfaces $\boldsymbol{P}(u,d,w)$ and $\boldsymbol{Q}(\mu,d,\omega)$ are G^n geometric continuous along the common boundary curve $\boldsymbol{P}(u,d,1)=\boldsymbol{Q}(\mu,d,0)$,

where c and d are arbitrary constants in their domains (refer to Fig. 3).

The following proposition presents a sufficient condition for the G^1 geometric continuity between two TBSs $\boldsymbol{P}(u,v,w)$ and $\boldsymbol{Q}(\mu,\nu,\omega)$ along their common boundary surface $\boldsymbol{P}(u,v,1)=\boldsymbol{Q}(\mu,\nu,0)$.

Proposition 4. Suppose the two TBSs $\boldsymbol{P}(u,v,w)$ (9) and $\boldsymbol{Q}(\mu,\nu,\omega)$ (10) have uniform knot vectors with Bézier end condition [38], respectively (Fig. 3). If

$$\boldsymbol{P}_{i,j,l_p}=\boldsymbol{Q}_{i,j,0}, \text{ and, } \boldsymbol{P}_{i,j,l_p}-\boldsymbol{P}_{i,j,l_p-1}=\alpha(\boldsymbol{Q}_{i,j,1}-\boldsymbol{Q}_{i,j,0}),$$

where $\alpha>0$ is a positive constant, the two TBSs $\boldsymbol{P}(u,v,w)$ and $\boldsymbol{Q}(\mu,\nu,\omega)$ are G^1 geometric continuous along their common boundary surface $\boldsymbol{P}(u,v,1)=\boldsymbol{Q}(\mu,\nu,0)$.

Proof: Referring to Fig. 3, if

$$\boldsymbol{P}_{i,j,l_p}=\boldsymbol{Q}_{i,j,0}, \text{ and, } \boldsymbol{P}_{i,j,l_p}-\boldsymbol{P}_{i,j,l_p-1}=\alpha(\boldsymbol{Q}_{i,j,1}-\boldsymbol{Q}_{i,j,0}),$$

it follows that $\boldsymbol{P}'_w(c,v,1)\parallel\boldsymbol{Q}'_w(c,\nu,0)$, and $\boldsymbol{P}'_w(u,d,1)\parallel\boldsymbol{Q}'_w(\mu,d,0)$, where c and d are arbitrary constants in the domain of $\boldsymbol{P}(u,v,w)$ and $\boldsymbol{Q}(\mu,\nu,\omega)$. That is, the two TBSs $\boldsymbol{P}(u,v,w)$ and $\boldsymbol{Q}(\mu,\nu,\omega)$ are G^1 geometric continuous along their common boundary surface $\boldsymbol{P}(u,v,1)=\boldsymbol{Q}(\mu,\nu,0)$. □

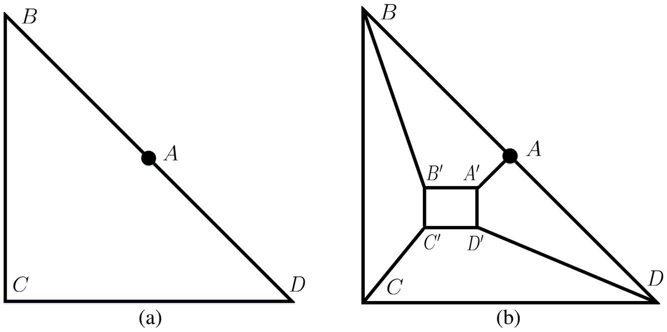


Figure 4 Improve the mesh quality using the pillow operation. (a) In the quadrilateral $ABCD$, the Jacobian value at vertex A is 0, because vertex D, A, and B are co-linear. (b) By the pillow operation, the Jacobian values at all of the vertices are greater than 0.

3.3 Pillow operation

The pillow operation was originally invented in Ref. [2] to improve the quality of hexahedral meshes and quadrilateral meshes. Take the quadrilateral $ABCD$ in Fig. 4(a) as an example, where the Jacobian value at vertex A is 0. To perform the pillow operation, the quadrilateral $ABCD$ shrinks inwards to the rectangle $A'B'C'D'$ (Fig. 4(b)), and then, the four corresponding edges AA', BB', CC', and DD' are linked, forming a quadrilateral mesh. After performing the pillow operation (Fig. 4(b)), the Jacobian values at all of the vertices are greater than 0, thus improving the mesh quality. Moreover, to perform the pillow operation on a hexahedral mesh, it first shrinks inwards, and then, the corresponding edges between the original and shrunk hexahedral meshes are linked to form a new hexahedral mesh with better quality. In this paper, pillow operation is employed to improve the quality of TBSs.

4 Geometric Feasible Direction algorithm

As pointed out in Section 2, the composition of TBSs is a step-by-step process, in the order of boundary curve fitting, boundary surface fitting, TBS fitting and smoothness improvement. The fitting problems

```
Input: An initial B-spline curve, B-spline surface, or TBS with control points arranged in a
       one-dimensional sequence $\{\boldsymbol{P}_i^{(0)}, i = 0, 1, \cdots, m\}$; an objective function $E$ and
       constraints (11)
Output: A B-spline curve, B-spline surface, or TBS meeting the termination condition
$k = 0$ ;
while the termination condition is not reached do
    Calculate the gradient vector $\nabla E^{(k)}$ (12) of the objective function $E$ (11) ;
    for $i = 0$ to $m$ do
        Determine the directional vector $\boldsymbol{D}_i^{(k)}$ (13) for the control point $\boldsymbol{P}_i^{(k)}$ ;
        Calculate the largest possible weight, $\tau_i$, ensuring that requirements 1) and 2) satisfied ;
    end
    Calculate the Armijo step $\alpha_k$ [39] ;
    $\boldsymbol{P}^{(k+1)} = \boldsymbol{P}^{(k)} + \alpha_k \boldsymbol{D}^{(k)}$ ;
    $k = k + 1$ ;
end
```

Algorithm 1: Geometric Feasible Direction algorithm (GFD)

and smoothness improvement in these steps are all modeled as constraint minimization problems,

$$\begin{aligned} min_{\boldsymbol{P}_i} \quad & E(\boldsymbol{P}_0, \boldsymbol{P}_1, \cdots, \boldsymbol{P}_m) \\ s.t. \quad & \text{Constraints} \end{aligned} \tag{11}$$

where $\boldsymbol{P}_i = (x_i, y_i, z_i),\ i = 0, 1, \cdots, m$ are the unknown control points, and E is an objective function. Moreover, the conditions for ensuring the validity of curves, surfaces, and TBSs are taken as the hard constraints (11). The specific minimization problems will be elucidated in subsequent sections.

To solve the constrained minimization problem (11), we developed the *geometric feasible direction* (GFD) algorithm (Algorithm 1). GFD has clear geometric meanings, so it is a kind of *geometric iterative fitting method*. As listed in Algorithm 1, the inputs to GFD algorithm are an initial B-spline curve, a B-spline surface, or a TBS with control points arranged in a one-dimensional sequence $\{\boldsymbol{P}_i^{(0)}, i = 0, 1, \cdots, m\}$, and a constrained minimization problem (11), including an objective function and the constraints.

Specifically, in the k^{th} iteration, the gradient vector of the objective function E (11) is,

$$\nabla E^{(k)} = \left(\frac{\partial E}{\partial x_0}, \frac{\partial E}{\partial y_0}, \frac{\partial E}{\partial z_0}, \cdots, \frac{\partial E}{\partial x_i}, \frac{\partial E}{\partial y_i}, \frac{\partial E}{\partial z_i}, \cdots, \frac{\partial E}{\partial x_m}, \frac{\partial E}{\partial y_m}, \frac{\partial E}{\partial z_m} \right)_{\boldsymbol{X}^{(k)}}, \tag{12}$$

where $\boldsymbol{X}^{(k)} = (x_0^{(k)}, y_0^{(k)}, z_0^{(k)}, \cdots, x_i^{(k)}, y_i^{(k)}, z_i^{(k)}, \cdots, x_m^{(k)}, y_m^{(k)}, z_m^{(k)})$. In the iteration, each control point $\boldsymbol{P}_i^{(k)}$ moves along the directional vector,

$$\boldsymbol{D}_i^{(k)} = -\left(\frac{\partial E}{\partial x_i}, \frac{\partial E}{\partial y_i}, \frac{\partial E}{\partial z_i} \right)_{\boldsymbol{X}^{(k)}}, \tag{13}$$

to produce the new control point $\boldsymbol{P}_i^{(k+1)}$, i.e.,

$$\boldsymbol{P}_i^{(k+1)} = \boldsymbol{P}_i^{(k)} + \tau_i \boldsymbol{D}_i^{(k)},$$

where $\tau_i \in [0, 1]$ is a weight. Initially, we set $\tau_i = 1, i = 0, 1, \cdots, m$, and the direction $\boldsymbol{D}^{(k)}$ in the k^{th} iteration is composed of,

$$\boldsymbol{D}^{(k)} = (\tau_0 \boldsymbol{D}_0^{(k)}, \tau_1 \boldsymbol{D}_1^{(k)}, \cdots, \tau_m \boldsymbol{D}_m^{(k)}). \tag{14}$$

Then, the weights $\tau_i, i = 0, 1, \cdots, m$ are adjusted one by one to make the direction $\boldsymbol{D}^{(k)}$ (14) a feasible one, which satisfies the following two requirements:

1) The new B-spline curve, surface, or TBS, generated by replacing $\boldsymbol{P}_i^{(k)}$ with $\boldsymbol{P}_i^{(k+1)}$, satisfies constraints of the minimization problem (11),

2) the condition, $-\frac{\boldsymbol{D}^{(k)}\cdot\nabla E^{(k)}}{\left\|\boldsymbol{D}^{(k)}\right\|_2\left\|\nabla E^{(k)}\right\|_2} > \delta_d$, holds as well.

To this end, we discretize $\tau_i \in [0,1]$ to $\{0, \frac{1}{n}, \frac{2}{n}, \cdots, \frac{n}{n}\}$, and select an as large as possible τ_i, which satisfies the two aforementioned requirements 1) and 2). In our implementation, we take $n = 20$, and $\delta_d = 0.3$.

Moreover, let,

$$\boldsymbol{P}^{(k)} = (\boldsymbol{P}_0^{(k)}, \boldsymbol{P}_1^{(k)}, \cdots, \boldsymbol{P}_m^{(k)}).$$

After the feasible direction $\boldsymbol{D}^{(k)}$ is figured out, we calculate the Armijo step α_k (please refer to Ref. [39] for the calculation of the Armijo step). Then, the new control points are generated as,

$$\boldsymbol{P}^{(k+1)} = \boldsymbol{P}^{(k)} + \alpha_k \boldsymbol{D}^{(k)}.$$

The GFD algorithm is presented in Algorithm 1, where the termination condition is taken as,

$$\left|\frac{E(\boldsymbol{P}^{(k+1)}) - E(\boldsymbol{P}^{(k)})}{E(\boldsymbol{P}^{(k)})}\right| < \varepsilon_e. \tag{15}$$

The value of ε_e will be specified in solving specific problems. The convergence analysis of the GFD algorithm is demonstrated in the Appendix.

5 Valid TBS generation by the geometric iterative method

In Section 2, the overview of the method for generating valid TBSs was presented. In this section, we will elucidate the details of the method.

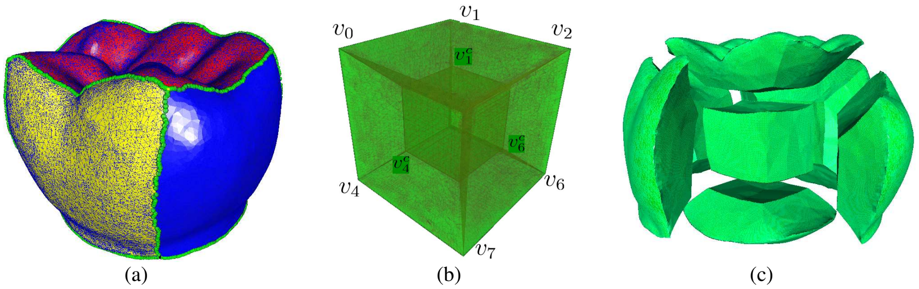


Figure 5 Partition of the input tet mesh model. (a) The input to the developed algorithm is a tet mesh with six surfaces segmented on its boundary mesh. (b) The tet mesh is parameterized into the cubic domain $[0,1] \times [0,1] \times [0,1]$, which is partitioned into seven sub-domains. (c) Mapping the seven sub-domains into the tet mesh model leads to the seven partitioned sub-volume meshes.

5.1 Partition of the tet mesh model by pillow operation

As stated above, the input to our method is a tet mesh model with six surfaces segmented on its boundary mesh (Fig. 5(a)). The scaled Jacobian value of the generated TBS is heavily influenced by the segmentation of the boundary mesh. If two adjacent surfaces are $C^n (n \geqslant 1)$ continuous across their common boundary, or even concave across the boundary, the Jacobian values at the points near the boundaries of the fitting TBS will be small or even negative (Figs. 6, 4(a)). This makes the generated TBS invalid. Fig. 6 illustrates the distribution of the scaled Jacobian values of the TBS fitting the tet mesh model in Fig. 5(a) by the method developed in Ref. [23]. It can be seen that, the small or negative Jacobian values mainly concentrate in the region (in blue) close to the boundaries of the surfaces, along which two adjacent surfaces are smoothly stitched.

In order to perform the pillow operation on the input tet mesh, the tet mesh model (Fig. 5(a)) is first parameterized into a cubic parameter domain $\Omega = [0,1]\times[0,1]\times[0,1]$ by the volume parameterization method [23] (Fig. 5(b)). Then, the parameter domain Ω shrinks to the subdomain,

$$\Omega_c = [\frac{1}{3}, \frac{2}{3}] \times [\frac{1}{3}, \frac{2}{3}] \times [\frac{1}{3}, \frac{2}{3}].$$

As illustrated in Fig. 5(b), the vertices of the cubes Ω and Ω_c are denoted as $v_0, v_1, \cdots, v_7$, and $v_0^c, v_1^c, \cdots, v_7^c$, respectively. Connecting v_i to $v_i^c, i = 0, 1, \cdots, 7$ generates six sub-domains $\Omega_u, \Omega_d, \Omega_l, \Omega_r, \Omega_f$, and Ω_b. For example, the sub-domain Ω_u is enclosed by the six faces

$$v_0v_1v_2v_3,\ v_0^cv_1^cv_2^cv_3^c,\ v_0v_0^cv_1^cv_1,\ v_1v_1^cv_2^cv_2,\ v_2v_2^cv_3^cv_3,\ \text{and},\ v_3v_3^cv_0^cv_0.$$

Mapping the seven sub-domains into the original tet mesh model produces seven partitioned sub-volumes (Fig. 5(c)).

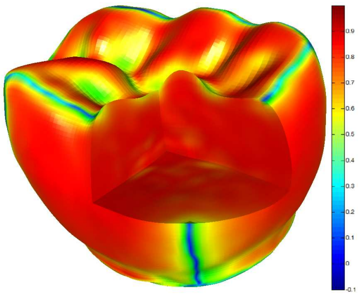


Figure 6 The small or negative scaled Jacobian values (in blue) concentrate near the boundaries where the adjacent surfaces are smoothly stitched.

5.2 Construction of the initial TBSs

In Section 5.1, the input tet mesh model is partitioned into seven sub-volumes. Each of them are parameterized into the cubic parameter domain by the volume parameterization method developed in [23]. Note that the parameterization on the common boundary curves and common boundary surfaces of adjacent sub-volumes should conform with each other. Moreover, each cubic parameter domain is sampled into a $(M+1)\times(N+1)\times(K+1)$ grid. Similar to the parameterization, the grid on the common boundary curves and common boundary surfaces of adjacent sub-volumes should be the same. Mapping the grids into the corresponding sub-volumes leads to the control grids of the initial TBSs, whose knot vectors are uniformly distributed in $[0,1]\times[0,1]\times[0,1]$ with Bézier end conditions. Owing to the conformity of the parameterization and control grids on the common boundary curves and boundary surfaces, there is the unique control grid on each boundary curve or boundary surface.

It should be pointed out that, the geometric iterative fitting method, which is employed to fit the sub-volumes (refer to Section 5.3), requires that the initial TBSs should be valid. If the input tet mesh has no extreme concave or even folder over dihedral angle (0 or even negative degree), the proposed method can generate valid initial TBSs. In fact, in all of the examples we handled, the initial TBSs generated by the proposed method are all valid. However, the proposed method does not guarantee the validity of the constructed initial TBSs in arbitrary case.

5.3 Geometric iterative fitting

The result generated by the developed method is a composition of seven valid TBSs. As stated above, the objective function for guaranteeing the validity of a TBS is highly nonlinear with a large number of unknowns, so the optimization is prone to fail. Even if we can find a solution, the computation for solving the optimization problem is complicated, owing to a significant number of unknowns. To reduce the difficulty in guaranteeing the validity of the TBSs, we solve this problem step by step, in the order of,

(1) boundary curve fitting (Section 5.3.1),

(2) boundary surface fitting (Section 5.3.2), and,

(3) TBS fitting (Section 5.3.3).

As mentioned above, the first objective we want to reach is that the generated TBSs should be valid, that is, the Jacobian value at any point of each TBS should be greater than 0. This means that the boundary curves and boundary surfaces of TBSs should also be valid (refer to Section 3.1 for validity condition).

5.3.1 *Boundary curve fitting*

Suppose the data point sequence to be fitted is,

$$\boldsymbol{V}_0, \boldsymbol{V}_1, \cdots, \boldsymbol{V}_{M_c}. \tag{16}$$

By volume parameterization [23], each data point $\boldsymbol{V}_j$ has been assigned a parameter $t_j \in [0,1], j = 0, 1, \cdots, M_c$. The data point sequence (16) will be fitted by a B-spline curve,

$$\boldsymbol{P}(u) = \sum_{i=0}^{m} \boldsymbol{P}_i B_i^k(u),\ u \in [0,1], \tag{17}$$

where, $\boldsymbol{P}_i$ are the control points, $B_i^k(u)$ are the B-spline basis functions with degree d, and the knot vector of the B-spline curve is uniformly defined in $[0,1]$ with Bézier end conditions.

The curve fitting should concern two aspects. One is the fitting error, and the other is the validity of the fitting curve. On one hand, the fitting error can be modeled as,

$$E_{fit}^c = \sum_{j=0}^{M_c} \|\boldsymbol{P}(u_j) - \boldsymbol{V}_j\|^2 .$$

On the other hand, denote α as the aperture of the minimum circular cone $\mathcal{C}$ enclosing the difference vectors,

$$\left\{\boldsymbol{T}_i = \frac{\boldsymbol{P}_{i+1} - \boldsymbol{P}_i}{\|\boldsymbol{P}_{i+1} - \boldsymbol{P}_i\|},\ i = 0, 1, \cdots, m-1\right\}, \tag{18}$$

with unit axis vector,

$$\boldsymbol{T} = \frac{\sum_{i=0}^{m-1} \boldsymbol{T}_i}{\left\|\sum_{i=0}^{m-1} \boldsymbol{T}_i\right\|}.$$

Evidently, if $\boldsymbol{P}'(u) = 0$, the B-spline curve (17) is not valid. On the contrary, the larger the norm $\|\boldsymbol{P}'(u)\|$, the better is the validity of the B-spline curve (17). With fixed lengths of the vectors $\boldsymbol{T}_i$ in (18), the smaller the aperture α, the larger is the norm $\|\boldsymbol{P}'(u)\|$. Therefore, the improvement of validity of the fitting curve can be achieved by minimizing the following objective function,

$$E_{val}^c = \frac{1}{m} \sum_{i=0}^{m-1} (1 - \boldsymbol{T}_i \cdot \boldsymbol{T}).$$

In conclusion, the fitting of the data point sequence (16) with a valid B-spline curve (17) can be formulated as the following minimization problem,

$$\begin{aligned} min_{\boldsymbol{P}_i}\ & E^c = (1-\lambda_c) E_{fit}^c + \lambda_c E_{val}^c \\ s.t.\ & 1) \text{ The control polygon of } \boldsymbol{P}(u) \text{ satisfies Proposition 1}, \\ & 2) \text{ Compatibility condition}, \\ & 3) \ \boldsymbol{P}_0 = \boldsymbol{V}_0,\ \boldsymbol{P}_m = \boldsymbol{V}_{M_c}, \end{aligned} \tag{19}$$

where $\lambda_c \in [0,1]$ is a weight. Constraint 1) in the minimization problem (19) ensures that the fitted curve satisfies the validity condition for B-spline curves (Proposition 1). The values of λ_c taken in our implementation are listed in Table 3.

Specifically, the B-spline curves are constructed piece by piece. After one piece of B-spline curve is constructed, it is fixed. When constructing a new B-spline curve $\boldsymbol{P}(u)$, it should satisfy the compatibility condition 2) with the adjacent constructed B-spline curves, which belong to the same B-spline surface as $\boldsymbol{P}(u)$. That is, they should satisfy Proposition 2. Specifically, refer to Fig. 7(a), when constructing the new B-spline curve $\boldsymbol{P}(u)$ with control points $\boldsymbol{P}_{00}, \boldsymbol{P}_{10}, \cdots, \boldsymbol{P}_{d_u,0}$ (red points), if there exists the piece of B-spline curve adjacent to $\boldsymbol{P}_{00}$, with control points $\boldsymbol{P}_{00}, \boldsymbol{P}_{01}, \cdots, \boldsymbol{P}_{0,d_v}$ (blue points), (the two

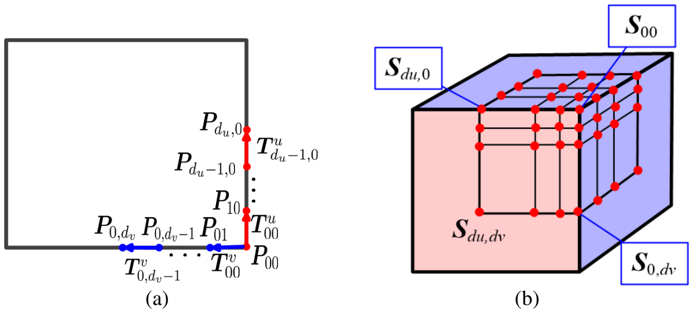


Figure 7 Compatibility conditions for curve fitting (a) and surface fitting (b).

pieces of B-spline curves belong to the same surface,) the difference vectors (refer to Eqs. (4) and (5)) $\boldsymbol{T}^u_{00}, \boldsymbol{T}^u_{10}, \cdots, \boldsymbol{T}^u_{d_u-1,0}$ and $\boldsymbol{T}^v_{00}, \boldsymbol{T}^v_{01}, \cdots, \boldsymbol{T}^v_{0,d_v-1}$ should satisfy Proposition 2.

The constrained minimization problem (19) is solved by the GFD algorithm (Algorithm 1), where the inputs are the initial B-spline curves constructed in Section 5.2, and the objective function and constraints of the minimization problem (19). In our implementation, the threshold ε_e in the termination condition (15) is taken as 10^{-4}.

5.3.2 *Boundary surface fitting*

After the boundaries of the seven sub-volumes are fitted by B-spline curves, they are fixed. The next task is to fit the boundary surfaces of the sub-volumes. Each of the boundary surfaces is a triangular mesh with four boundary curves. Suppose the *inner vertices* of such a triangular mesh are $\boldsymbol{V}_i, i = 0, 1, \cdots, M_s$, each of which is assigned a pair of parameter values (u_i, v_i) by the volume parameterization [23]. Moreover, suppose the B-spline surface fitted to the triangular mesh is,

$$\boldsymbol{S}(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} \boldsymbol{S}_{ij} B_i^{d_u}(u) B_j^{d_v}(v), \ (u,v) \in [0,1] \times [0,1], \tag{20}$$

where, $\boldsymbol{S}_{ij}$ are the control points, $B_i^{d_u}(u)$ and $B_j^{d_v}(v)$ are B-spline basis functions with degrees d_u and d_v, respectively, both defined in $[0,1]$ with Bézier end conditions.

Similar to the curve fitting, the surface fitting also involves two aspects: the fitting precision, and the validity. The fitting precision is measured by the following energy function,

$$E^s_{fit} = \sum_{k=0}^{M_s} \left\| \boldsymbol{S}(u_k, v_k) - \boldsymbol{V}_k \right\|^2. \tag{21}$$

Moreover, the validity of the B-spline surface should be improved as per the requirement. Suppose $\mathcal{C}_u$ and $\mathcal{C}_v$ are the minimum circular cones enclosing the difference vectors $\boldsymbol{T}^u_{ij}$ (4) and $\boldsymbol{T}^v_{ij}$ (5), respectively, with unit axis vectors $\boldsymbol{T}^u$ and $\boldsymbol{T}^v$, taken as,

$$\boldsymbol{T}^u = \frac{\sum_{i=0}^{m-1}\sum_{j=0}^{n} \boldsymbol{T}^u_{ij}}{\left\|\sum_{i=0}^{m-1}\sum_{j=0}^{n} \boldsymbol{T}^u_{ij}\right\|}, \ \text{and}, \ \boldsymbol{T}^v = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n-1} \boldsymbol{T}^v_{ij}}{\left\|\sum_{i=0}^{m}\sum_{j=0}^{n-1} \boldsymbol{T}^v_{ij}\right\|}. \tag{22}$$

The validity of a B-spline surface can be improved by two strategies. Firstly, as analyzed in Section 5.3.1, the apertures of the two cones should be as small as possible, which is achieved by minimizing the following two energy functions,

$$E^s_u = \frac{1}{m(n+1)} \sum_{i=0}^{m-1}\sum_{j=0}^{n} \left(1 - \boldsymbol{T}^u_{ij} \cdot \boldsymbol{T}^u\right), \ \text{and},$$

$$E^s_v = \frac{1}{(m+1)n} \sum_{i=0}^{m}\sum_{j=0}^{n-1} \left(1 - \boldsymbol{T}^v_{ij} \cdot \boldsymbol{T}^v\right).$$

Secondly, the two cones should be as close to perpendicularity as possible, which is formulated as the minimization of the following energy function (refer to Eq. (22)),

$$E^s_{uv} = (\boldsymbol{T}^u \cdot \boldsymbol{T}^v)^2.$$

Therefore, the B-spline surface fitting the triangular mesh with validity guarantee can be generated by the following minimization problem,

$$\begin{aligned} min_{\boldsymbol{S}_{ij}} \quad & E^s = (1-\lambda_s-\mu_s)E^s_{fit} + \lambda_s(E^s_u + E^s_v) + \mu_s E^s_{uv} \\ s.t.\ & \text{1) The control polygon of } \boldsymbol{S}(u,v) \text{ satisfies Proposition 2,} \\ & \text{2) Compatibility condition, and,} \\ & \text{3) The boundary control points of } \boldsymbol{S}(u,v) \text{ (20) are fixed,} \end{aligned} \tag{23}$$

where $\lambda_s, \mu_s \in [0,1]$ are weights, whose values can be found in Table 3, and Constraint 1) in Eq. (23) ensures that the B-spline surface $\boldsymbol{S}(u,v)$ (20) satisfies the validity condition for B-spline surfaces (Proposition 2).

Similar to the curve case, the B-spline surfaces are constructed piece by piece. Hence, the construction of a new B-spline surface $\boldsymbol{S}(u,v)$ (the surface in light red in Fig. 7(b)), should satisfy the *compatibility condition* 2) (Eq. (23)) with existing B-spline surfaces (the two surfaces in light blue in Fig. 7(b)), which are adjacent to one of the corners of the new B-spline surface $\boldsymbol{S}(u,v)$, and belong to the same TBS with $\boldsymbol{S}(u,v)$. In other words, the corresponding sub-control-polygons of these surfaces adjacent to one corner of $\boldsymbol{S}(u,v)$ should satisfy Proposition 3. Specifically, refer to Fig. 7(b), supposing the degree of the TBS is $d_u \times d_v \times d_w$, the three sub-control-polygons with size $d_u \times d_v$, $d_v \times d_w$, and $d_u \times d_w$, respectively (refer to Eq. (6)), should satisfy the validity condition for TBSs, i.e., Proposition 3.

The constrained minimization problem (23) is solved by the GFD algorithm (Algorithm 1). The inputs to the GFD algorithm are the initial B-spline surfaces constructed in Section 5.2, the objective function and constraints of the minimization problem (23). According to Eq. (20), the control points of the B-spline surface are arranged in a one-dimensional sequence with lexicographic order, i.e., $\{\boldsymbol{S}_{00}, \boldsymbol{S}_{01}, \cdots, \boldsymbol{S}_{mn}\}$. It should be pointed out that the threshold ε_e in the termination condition (15) is set as 10^{-5}.

5.3.3 *TBS fitting*

Now, it is the time to fit seven sub-volumes by TBSs. Similar to the boundary curve fitting and boundary surface fitting, the seven sub-volumes are also fitted one by one. Each sub-volume is a tetrahedral (tet) mesh, whose boundary triangular meshes were fitted with B-spline surfaces (refer to Section 5.3.2), and thus fixed in the TBS fitting. Suppose $\boldsymbol{V}_n, n = 0, 1, \cdots, M_h$ are the inner vertices of a sub-volume mesh with parameter values (u_n, v_n, w_n), and the TBS fitting the sub-volume mesh vertices is,

$$\boldsymbol{H}(u,v,w) = \sum_{i=0}^{m_u}\sum_{j=0}^{m_v}\sum_{k=0}^{m_w} \boldsymbol{H}_{ijk} B_i^{d_u}(u) B_j^{d_v}(v) B_k^{d_w}(w), \tag{24}$$

where $\boldsymbol{H}_{ijk}$ are the control points, $B_i^{d_u}(u), B_j^{d_v}(v)$, and $B_k^{d_w}(w)$ are the B-spline basis functions of degrees d_u, d_v, and d_w, defined on the interval $[0,1]$, with Bézier end conditions, respectively.

The generation of a TBS should involve the following factors:

1) The fitting precision to the tet mesh vertices, and,

2) the improvement of validity of each TBS.

On one hand, the fitting precision is modeled by the following formula,

$$E^h_{fit} = \sum_{n=0}^{M_h} \left\|\boldsymbol{H}(u_n, v_n, w_n) - \boldsymbol{V}_n\right\|^2. \tag{25}$$

On the other hand, to improve the validity of a TBS, we define the difference vectors along the $u-, v-$, and $w-$directions, respectively, as,

$$\boldsymbol{T}^u_{ijk} = \frac{\boldsymbol{H}_{i+1,j,k} - \boldsymbol{H}_{ijk}}{\|\boldsymbol{H}_{i+1,j,k} - \boldsymbol{H}_{ijk}\|}, \quad \boldsymbol{T}^v_{ijk} = \frac{\boldsymbol{H}_{i,j+1,k} - \boldsymbol{H}_{ijk}}{\|\boldsymbol{H}_{i,j+1,k} - \boldsymbol{H}_{ijk}\|}, \quad \boldsymbol{T}^w_{ijk} = \frac{\boldsymbol{H}_{i,j,k+1} - \boldsymbol{H}_{ijk}}{\|\boldsymbol{H}_{i,j,k+1} - \boldsymbol{H}_{ijk}\|}.$$

Suppose $\mathcal{C}_u, \mathcal{C}_v$, and $\mathcal{C}_w$ are the minimum circular cones enclosing $\boldsymbol{T}^u_{ijk}$, $\boldsymbol{T}^v_{ijk}$, and $\boldsymbol{T}^w_{ijk}$, with unit axis vectors $\boldsymbol{T}^u, \boldsymbol{T}^v$, and $\boldsymbol{T}^w$, taken as, respectively,

$$\boldsymbol{T}^u = \frac{\sum_{ijk} \boldsymbol{T}^u_{ijk}}{\left\|\sum_{ijk} \boldsymbol{T}^u_{ijk}\right\|}, \quad \boldsymbol{T}^v = \frac{\sum_{ijk} \boldsymbol{T}^v_{ijk}}{\left\|\sum_{ijk} \boldsymbol{T}^v_{ijk}\right\|}, \quad \boldsymbol{T}^w = \frac{\sum_{ijk} \boldsymbol{T}^w_{ijk}}{\left\|\sum_{ijk} \boldsymbol{T}^w_{ijk}\right\|}. \tag{26}$$

Similar as the analysis in Section 5.3.2, the smaller the apertures of the cones, the better the validity of the TBS, which is modeled as the minimization of the following three energy functions, i.e.,

$$E^h_u = \frac{1}{N_u} \sum_{i,j,k} (1 - \boldsymbol{T}^u_{ijk} \cdot \boldsymbol{T}^u), \quad E^h_v = \frac{1}{N_v} \sum_{i,j,k} (1 - \boldsymbol{T}^v_{ijk} \cdot \boldsymbol{T}^v) \quad E^h_w = \frac{1}{N_w} \sum_{i,j,k} (1 - \boldsymbol{T}^w_{ijk} \cdot \boldsymbol{T}^w),$$

where, N_u, N_v, and N_w are the number of the vectors $\boldsymbol{T}^u_{ijk}, \boldsymbol{T}^v_{ijk}$, and $\boldsymbol{T}^w_{ijk}$, respectively. In addition, the closer the perpendicularities of the three cones are to each other, the better is the validity of the TBS, which can be formulated as the minimization of the following energy functions:

$$E^h_{uv} = (\boldsymbol{T}^u \cdot \boldsymbol{T}^v)^2, \quad E^h_{vw} = (\boldsymbol{T}^v \cdot \boldsymbol{T}^w)^2, \quad E^h_{uw} = (\boldsymbol{T}^u \cdot \boldsymbol{T}^w)^2.$$

Essentially, the minimization problem for generating the valid TBSs can be modeled as,

$$\begin{aligned} & min_{\boldsymbol{H}_{ijk}} \quad E^h = (1 - \lambda_h - \mu_h) E^h_{fit} + \lambda_h (E^h_u + E^h_v + E^h_w) + \mu_h (E^h_{uv} + E^h_{uw} + E^h_{vw}) \\ & s.t. \quad \text{1) The control grid of } \boldsymbol{H}(u,v,w) \text{ satisfies Proposition 3,} \\ & \qquad\;\; \text{2) The boundary control points of } \boldsymbol{H}(u,v,w) \text{ (24) are fixed,} \end{aligned} \tag{27}$$

where $\lambda_h, \mu_h \in [0,1]$ are weights, and Constraint 1) makes the TBS $\boldsymbol{H}(u,v,w)$ satisfy the validity condition for TBSs (Proposition 3). The values of weights $\lambda_h,\ \mu_h \in [0,1]$ are presented in Table 3.

The constrained minimization problem (27) is solved by the GFD algorithm (Algorithm 1), with the initial TBSs constructed in Section 5.2 as input. The control points of an input TBS (refer to Eq. (24)) are arranged in a one-dimensional sequence with lexicographic order, i.e., $\{\boldsymbol{H}_{000}, \boldsymbol{H}_{001}, \cdots, \boldsymbol{H}_{m_u,m_v,m_w}\}$. Moreover, we set $\varepsilon_e = 10^{-6}$ in the termination condition (15).

5.3.4 *Smoothness and fairness improvement*

In this section, we will improve the smoothness between two adjacent TBSs, and the fairness of each TBS. Suppose $\boldsymbol{H}_l(u,v,w), l = 0, 1, \cdots, 6$ is a TBS with control points $\boldsymbol{H}_{l;ijk}$. Since the boundary patches are fixed before the TBS fitting, the two adjacent TBSs have reached G^0 continuity. According to Proposition 4 and Fig. 3, the two adjacent TBSs $H_{l_1}(u,v,w)$ and $H_{l_2}(u,v,w)$ are G^1 continuous, if the two unit vectors $\boldsymbol{T}_{l_1;ij}$ and $\boldsymbol{T}_{l_2;ij}$, i.e.,

$$\boldsymbol{T}_{l_1;ij} = \frac{\boldsymbol{P}_{i,j,l_p} - \boldsymbol{P}_{i,j,l_p-1}}{\left\|\boldsymbol{P}_{i,j,l_p} - \boldsymbol{P}_{i,j,l_p-1}\right\|}, \text{ and, } \boldsymbol{T}_{l_2;ij} = \frac{\boldsymbol{Q}_{i,j,1} - \boldsymbol{Q}_{i,j,0}}{\|\boldsymbol{Q}_{i,j,1} - \boldsymbol{Q}_{i,j,0}\|},$$

have the same direction. Then, the smoothness between a pair of adjacent TBSs can be improved by the minimization of the energy function,

$$\sum_{ij} (1 - \boldsymbol{T}_{l_1;ij} \cdot \boldsymbol{T}_{l_2;ij}).$$

The smoothness between all of the adjacent TBSs can be improved by minimizing the energy,

$$E^h_{smooth} = \sum_{\substack{\text{any pair of adjacent TBSs} \\ H_{l_1} \text{ and } H_{l_2}}} \sum_{i,j} (1 - \boldsymbol{T}_{l_1;ij} \cdot \boldsymbol{T}_{l_2;ij}).$$

Meanwhile, the fairness of the TBSs is improved by minimizing the fairness energy,

$$E^h_{fair} = \sum_l \int_0^1 \int_0^1 \int_0^1 \left(\left(\frac{\partial^2 \boldsymbol{H}_l}{\partial u^2} \right)^2 + \left(\frac{\partial^2 \boldsymbol{H}_l}{\partial v^2} \right)^2 + \left(\frac{\partial^2 \boldsymbol{H}_l}{\partial w^2} \right)^2 + \right.$$
$$\left. 2 \left(\frac{\partial^2 \boldsymbol{H}_l}{\partial u \partial v} \right)^2 + 2 \left(\frac{\partial^2 \boldsymbol{H}_l}{\partial u \partial w} \right)^2 + 2 \left(\frac{\partial^2 \boldsymbol{H}_l}{\partial v \partial w} \right)^2 \right) dudvdw.$$

The fairness energy E^h_{fair} is a quadratic function of control points of the TBS H_l. In our implementation, the composite trapezoidal rule [40] is utilized to calculate the integrals in the fairness energy E^h_{fair}.

Therefore, the improvement of smoothness and fairness with validity guarantee can be modeled as the constrained minimization problem,

$$\begin{aligned} &min_{\boldsymbol{H}_{l;ijk}} \quad (1-\lambda_f)E^h_{smooth} + \lambda_f E^h_{fair} \\ s.t.\ &1) \text{ The control grid of each TBS satisfies Proposition 3,} \\ &2) \text{ The boundary control points of each TBS are fixed,} \end{aligned} \tag{28}$$

where, $\lambda_f \in [0,1]$ is a weight, listed in Table 3. Constraint 1) in (28) guarantees that each TBS satisfies the validity condition for TBSs (Proposition 3).

The constrained minimization problem (28) is solved by the GFD algorithm (Algorithm 1). The input to the GFD algorithm is the composition of the seven TBSs, constructed in Section 5.3.3, with control points arranged in a one-dimensional sequence. The threshold ε_e in the termination condition (15) is taken as 10^{-5}.

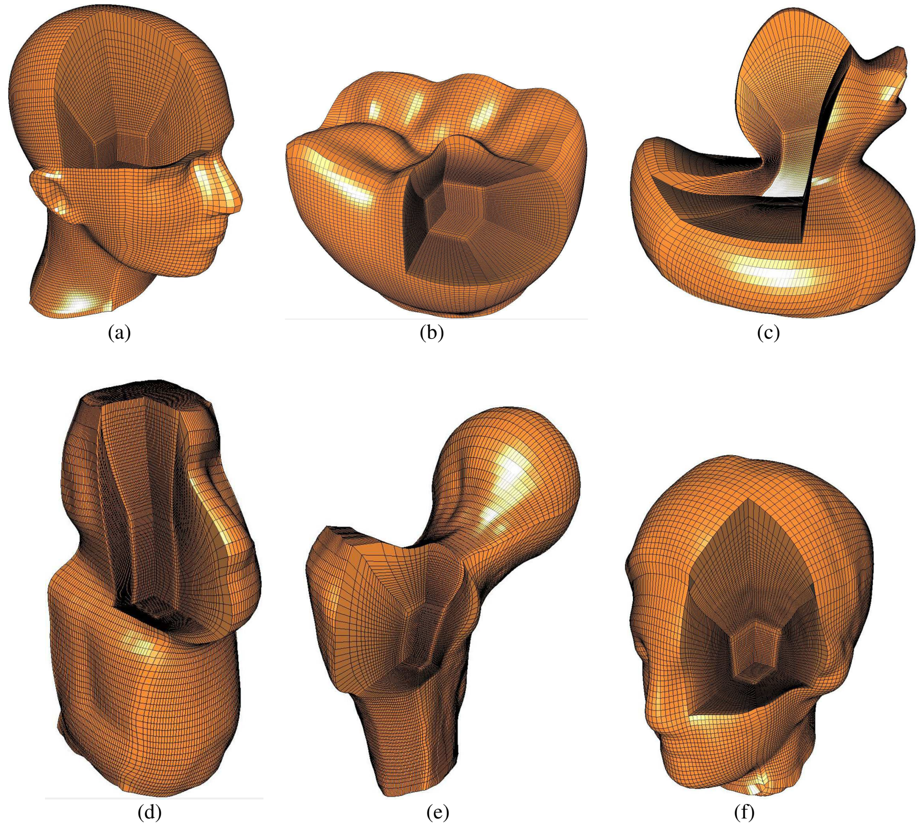
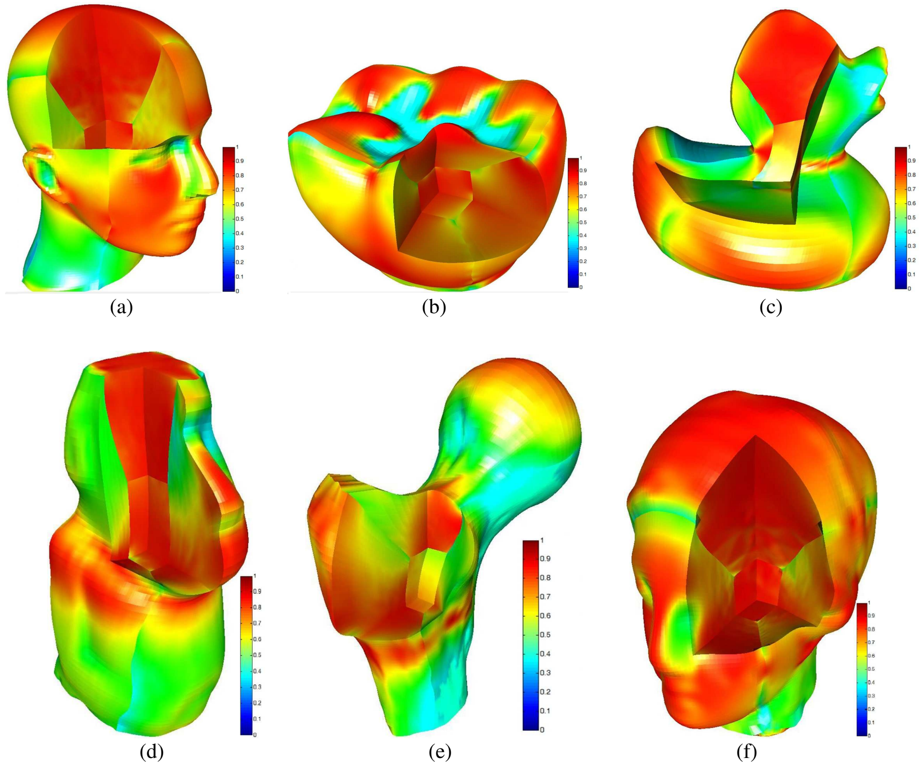


Figure 8 The cut-away views of TBSs generated by our method. (a) Mannequin. (b) Tooth. (c) Duck. (d) Moai. (e) Ball joint. (f) Venus.

6 Results

The algorithm developed in this paper is implemented using Visual Studio C++ 2010, and run on a PC with Intel Core2 Quad CPU Q9400 2.66 GHz and 4GB memory. In this section, we demonstrate six TBS models generated by the developed method. In Fig. 8, the cut-away views of the TBS models are illustrated. It can be seen that the iso-parametric curves vary smoothly not only inside a single TBS, but between two adjacent TBSs as well. Moreover, in Fig. 9, the distribution of the scaled Jacobian values [41] of the TBSs is visualized with different colors. The darker the red color, the higher the scaled Jacobian values. As shown in Fig. 9, the scaled Jacobian values of these models are all positive. A majority of region of the six TBS models is in red. Actually, in each of the six TBS models, the Jacobian values are larger than 0.5 in over 80% region.



(a) (b) (c)

(d) (e) (f)

Figure 9 The distribution of scaled Jacobian values on TBSs generated by our method. (a) Mannequin. (b) Tooth. (c) Duck. (d) Moai. (e) Ball joint. (f) Venus.

The statistics of TBSs generated by our method are listed in Table 1. The third column of Table 1 is the number of control points of TBSs, reading as $m \times n \times k \times l$. It means that, the number of control points of the central TBS is $m \times n \times k$, those of the six surrounding TBSs are $m \times n \times l$, $m \times n \times l$, $m \times k \times l$, $m \times k \times l$, $n \times k \times l$, and $n \times k \times l$, respectively. The fourth column is the fitting error, which is defined as,

$$\frac{\sqrt{\sum_{l=0}^{6}\frac{\sum_{i=0}^{N_i-1}\|\boldsymbol{H}_l(u_i,v_i,w_i)-\boldsymbol{V}_i\|^2}{N_i}}}{L},$$

where N_i is the number of tet mesh vertices in the i^{th} sub-volume $(i = 0, 1, \cdots, 6)$, and L is the diagonal length of the bounding box of the whole model. The fitting errors for all of the six models are in the order of magnitude 10^{-2}. Moreover, in the fifth column, the ratios of the volume of the region with scaled Jacobian in $(0, 0.2]$ to the whole volume of the TBS are listed. It can be seen that the largest ratio (for the model Ball joint) is 1.96%, and the ratios for the other models are all below 0.5%, meaning that

Table 1 Statistical data of the TBS generation method developed in this paper.

model	#vert.[1]	#control[2]	fitting error	ratio[3]	time[4]			
					curve	surface	TBS	quality
Mannequin	44152	$50 \times 50 \times 90 \times 30$	5.06×10^{-2}	0.11%	3.26	344.85	1268.87	605.72
Tooth	61311	$30 \times 26 \times 26 \times 26$	5.07×10^{-3}	0.12%	1.62	198.54	464.13	124.66
Duck	41998	$30 \times 30 \times 24 \times 20$	7.65×10^{-2}	0.09%	1.37	166.14	393.67	117.90
Moai	8831	$30 \times 30 \times 30 \times 20$	3.89×10^{-2}	0.34%	1.38	230.28	538.54	111.10
Ball joint	43994	$30 \times 22 \times 34 \times 15$	2.37×10^{-2}	1.96%	1.41	280.62	685.98	130.53
Venus	35858	$30 \times 30 \times 30 \times 24$	1.91×10^{-2}	0.00%	1.34	228.32	573.67	165.90

[1] Number of vertices of the input tet mesh model.
[2] Number of the control points of the TBSs generated by our method.
[3] The ratio of the volume of the region with scaled Jacobian in (0,0.2] to the whole volume of the TBS.
[4] Time (in second) cost in curve, surface, TBS fitting and quality improvement (smoothness and fairness improvement).

regions with scaled Jacobian in $(0, 0.2]$ in generated TBSs are very small. In the last four columns, we list the time cost (in second) in curve, surface, TBS fitting, and quality improvement (i.e., smoothness and fairness improvement). The whole process for the six models takes time ranging from 11 min to 37 min.

Table 2 Comparison of the TBSs generated by our method and the methods in Ref. [6] [23].

model	min Jac.[1]			avg Jac[2]		
	our method	method in [6]	method in [23]	our method	method in [6]	method in [23]
Mannequin	0.1048	-0.5678	-0.9771	0.8713	0.8005	0.8267
Tooth	0.1576	0.0557	-0.1171	0.9357	0.8393	0.9321
Duck	0.0851	-0.6855	-0.9588	0.8154	0.6471	0.8377
Moai	0.1127	0.1833	-0.9850	0.8412	0.9118	0.8804
Ball joint	0.0987	-0.5664	-0.5972	0.8367	0.7437	0.8189
Venus	0.1243	0.0001	-0.7239	0.9176	0.7484	0.8821

[1] Minimum scaled Jacobian value.
[2] Average scaled Jacobian value.

Moreover, for comparison, the minimum scaled Jacobian values and average Jacobian values of TBSs generated by the two methods developed in Refs. [6], [23] and our method, respectively, are presented in Table 2. The average Jacobian value is calculated as,

$$avgJac = \frac{\int\int\int_{\Omega} J(x,y,z)dxdydz}{\int\int\int_{\Omega} dxdydz},$$

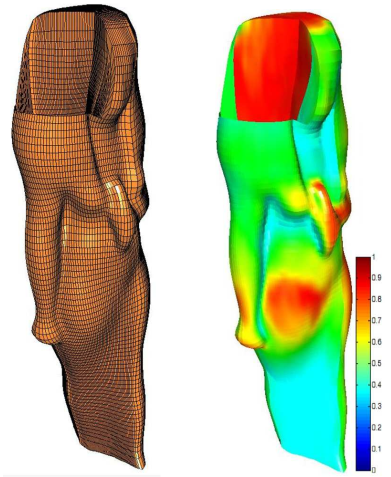
where $J(x, y, z)$ is the scaled Jacobian value [41] at (x, y, z). Specifically, on one hand, inputs to the method in [6] are the boundary B-spline patches of TBSs generated by our method. On the other hand, the resolutions of control grids of boundary B-spline patches employed in the method in [23] are the same as those in our method, and fitting errors of TBSs all reach the same order of magnitude, i.e., 10^{-2}. In Ref. [6], the divide-and-conquer technique was employed to reduce the original large optimization problem into a set of separable small sub-problems. And then, isolated groups of invalid Bézier patches are optimized separately. However, the movements of B-spline control points in optimizing an invalid patch will affect its neighbor patches. Therefore, even an isolated invalid patch becomes valid by optimization, the movements of its B-spline control points can make a valid neighbor patch turn into an invalid one. In Table 2, minimum scaled Jacobian values of three TBSs generated by the method in Ref. [6] are negative, and those generated by the method in Ref. [23] are all negative. It is worth noting that, minimum scaled Jacobian values of all TBSs produced by our method are positive. Additionally, in the six models, the average scaled Jacobian values of four TBSs generated by our method are superior to those of other methods (refer to Table 2).

It should be pointed out that, because the values of energy functions, i.e., E^c_{fit} and E^c_{val} in Eq. (19), E^s_{fit}, E^s_u, E^s_v, and E^s_{uv} in Eq. (23), $E^h_{fit}, E^h_u, E^h_v, E^h_w, E^h_{uv}, E^h_{uw}$, and E^h_{vw} in Eq. (27), and E^h_{smooth}, E^h_{fair} in Eq. (28), differ much for different models, different weights should be chosen in solving the minimization problems (Eqs. (19) (23) (27) (28)), in order to balance the energy functions well, and achieve desirable results. The weights employed in generating the six TBSs are presented in Table 3.

Table 3 Weights employed in generating TBSs.

	Mannequin	Tooth	Duck	Moai	Ball joint	Venus
λ_c	0.0001	0.99	0.98	0.1	0.0001	0.0001
λ_s, λ_h	0.0001	0.9	0.9	0.1	0.0001	0.0001
μ_s, μ_h	0.0001	0.09	0.09	0.1	0.0001	0.0001
λ_f	0.9	0.0001	0.0001	0.6	0.9	0.9

Limitations: There are several limitations of the developed method. First, the segmentation by pillowing operation is not suitable for some types of tet mesh models. For example, because the bottom of the *Isis* model shown in Fig. 10 is slim, the quality of the TBS generated by our method is not very good, though the Jacobian value can be guaranteed positive. The fitting precision can just reach the order of magnitude 10^{-1}, and the Jacobian values in 16.69% region lie in the interval $(0, 0.2]$. Therefore, a more reasonable tet mesh segmentation method should be developed in the future. Second, the validity of the generated TBS model depends on an valid initial TBS. Although the developed initial TBS construction method can generate a valid initial TBS when the input tet mesh has no extreme concave or even folder over dihedral angle, it does not guarantee the validity of constructed initial TBSs in arbitrary cases.



(a) (b)

Figure 10 The pillow operation is not suitable for some types of tet mesh models. For example, scaled Jacobian values in 16.69% region of the *Isis* TBS model lie in the interval $(0, 0.2]$. (a) Cut away view of the TBS model. (b) Distribution of the scaled Jacobian values.

7 Conclusion

In this paper, we developed a method to generate a TBS by fitting a tet mesh model. The input to this method is a tet mesh model with six surfaces segmented on its boundary mesh. To improve the Jacobian values in the regions close to the boundaries, the tet mesh model was first partitioned into seven sub-volumes using the pillow operation. Then, a geometric iterative fitting algorithm was developed to fit the boundary curves, boundary surfaces, and the sub-volumes, separately. Because the validity conditions are integrated into the geometric iterations, the Jacobian values of the generated TBSs are guaranteed to be positive, if the initial TBSs are valid. The method developed in this paper provides a practical solution for producing valid TBSs for a wide range of models.

Appendix: Convergence analysis of the GFD algorithm

In this appendix, the convergence analysis of the GFD algorithm (Algorithm 1) will be presented. Let,

$$\begin{aligned} D &= \{(\boldsymbol{P}_0, \boldsymbol{P}_1, \cdots, \boldsymbol{P}_m) \mid \text{the constraints in (11) are satisfied}\}, \\ L &= \left\{(\boldsymbol{P}_0, \boldsymbol{P}_1, \cdots, \boldsymbol{P}_m) \mid E(\boldsymbol{P}_0, \cdots, \boldsymbol{P}_m) \leqslant E(\boldsymbol{P}_0^{(0)}, \cdots, \boldsymbol{P}_m^{(0)})\right\}. \end{aligned}$$

The analysis of the convergence of the GFD algorithm (Algorithm 1) depends mainly on the following lemma [39].

Lemma 1. Suppose $\nabla E(\boldsymbol{P}_0, \boldsymbol{P}_1, \cdots, \boldsymbol{P}_m)$ (11) is uniformly continuous on the region $D \cap L$, and the angle θ_k between the feasible direction $\boldsymbol{D}^{(k)}$ generated by the GFD algorithm (Algorithm 1) and $-\nabla E^{(k)}$ satisfies,

$$0 \leqslant \theta_k \leqslant \frac{\pi}{2} - \mu, \quad \text{for some } \mu > 0.$$

Then, $\nabla E^{(k)} = 0$ for some k, or $E(\boldsymbol{P}_0^{(k)}, \boldsymbol{P}_1^{(k)}, \cdots, \boldsymbol{P}_m^{(k)}) \to -\infty, (k \to \infty)$, or $\nabla E^{(k)} \to 0, (k \to \infty)$.

Then, the convergence theorem for the GFD algorithm (Algorithm 1) is followed.

Theorem 1. If $\nabla E(\boldsymbol{P}_0, \boldsymbol{P}_1, \cdots, \boldsymbol{P}_m)$ (11) is uniformly continuous on the region $D \cap L$, and the objective function $E(\boldsymbol{P}_0, \boldsymbol{P}_1, \cdots, \boldsymbol{P}_m)$ (11) is bounded, the GFD algorithm (Algorithm 1) is convergent.

Proof: Denote θ_k as the angle between the feasible direction $\boldsymbol{D}^{(k)}$ and $-\nabla E^{(k)}$. In the iteration of the GFD algorithm, the following inequality holds,

$$-\frac{\boldsymbol{D}^{(k)} \cdot \nabla E^{(k)}}{\left\|\boldsymbol{D}^{(k)}\right\|_2 \left\|\nabla E^{(k)}\right\|_2} > \delta_d.$$

So, there exists some $\mu > 0$, such that,

$$0 \leqslant \theta_k \leqslant \frac{\pi}{2} - \mu.$$

Therefore, based on Lemma 1, together with that the objective function $E(\boldsymbol{P}_0, \boldsymbol{P}_1, \cdots, \boldsymbol{P}_m)$ (11) is bounded, the GFD algorithm (Algorithm 1) is convergent. □

Acknowledgements This work is supported by the Natural Science Foundation of China (Grant No. 61379072).

References

1 Hughes J, Cottrell J, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Computer methods in applied mechanics and engineering, 2005, 194 (39) : 4135–4195.

2 Mitchell S A, Tautges T J, Pillowing doublets: refining a mesh to ensure that faces share at most one edge. In: Proceedings of the 4th International Meshing Roundtable, Citeseer, 1995. 231–240.

3 Zhang Y, Bazilevs Y, Goswami S, et al. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. Computer methods in applied mechanics and engineering, 2007, 196 (29) : 2943–2959.

4 Martin T, Cohen E, Kirby R M. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. Computer Aided Geometric Design, 2009, 26 (6) : 648–664.

5 Xu G, Mourrain B, Duvigneau R, et al. Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. Computer-Aided Design, 2013, 45 (2) : 395–404.

6 Wang X, Qian X. An optimization approach for constructing trivariate B-spline solids. Computer-Aided Design, 2014, 46 : 179–191.

7 Aigner M, Heinrich C , Jüttler B, et al. Swept volume parameterization for isogeometric analysis. In Proceedings of Mathematics of Surfaces XIII, Berlin, 2009. 19–44.

8 Akhras H A, Elguedja T, Gravouila A, et al. Isogeometric analysis-suitable trivariate NURBS models from standard B-Rep models. Computer Methods in Applied Mechanics and Engineering, 2016, 307 : 256–274.

9 Chan C L, Anitescu C, Rabczuk T. Volumetric parametrization from a level set boundary representation with PHT-splines. Computer-Aided Design, 2017, 82 : 29–41.

10 Fang M, Lu J, Peng Q. Volumetric data modeling and analysis based on seven-directional box spline. Science China Information Sciences, 2014, 57(6): 1-14.

11 Sederberg T W, Cardon D L, Finnigan G T, et al. T-spline simplification and local refinement. ACM Transactions on Graphics, 2004, 23 : 276–283.

12 Escobar J, Cascón J, Rodríguez E, et al. A new approach to solid modeling with trivariate T-splines based on mesh optimization. Computer Methods in Applied Mechanics and Engineering, 2011, 200 (45) : 3210–3222.

13 Zhang Y, Wang W, Hughes T J. Solid T-spline construction from boundary representations for genus-zero geometry. Computer Methods in Applied Mechanics and Engineering, 2012, 249 : 185–197.

14 Wang W, Zhang Y, Liu L, et al. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. Computer-Aided Design, 2013, 45 (2) : 351–360.

15 Liu L, Zhang Y, Hughes T J R, et al. Volumetric T-spline construction using boolean operations. Engineering with Computers, 2014, 30 (4) : 425–439.

16 Zhang Y, Wang W, Hughes T J R. Conformal solid T-spline construction from boundary T-spline representations. Computational Mechanics, 2013, 51 (6) : 1051–1059.

17 Burkhart D, Hamann B, Umlauf G, Iso-geometric finite element analysis based on Catmull-Clark subdivision solids. Computer Graphics Forum, 2010, 29 : 1575–1584.

18 Hua J, He Y, Qin H. Multiresolution heterogeneous solid modeling and visualization using trivariate simplex splines. In: Proceedings of the Ninth ACM Symposium on Solid Modeling and Applications, Eurographics Association, 2004. 47–58.

19 Li B, Li X, Wang K, et al. H. Generalized polycube trivariate splines. in: Shape Modeling International Conference (SMI), 2010, IEEE, 2010, pp. 261–265.

20 Wang K, Li X, Li B, et al. Restricted trivariate polycube splines for volumetric data modeling. IEEE Transactions on Visualization and Computer Graphics, 2012, 18 (5) : 703–716.

21 Si H. Tetgen: A quality tetrahedral mesh generator and three-dimensional delaunay triangulator. 2007, URL http://tetgen. berlios. de.

22 Schöberl J. Netgen: An advancing front 2d/3d-mesh generator based on abstract rules. Computing and Visualization in Science, 1997, 1 (1) : 41–52.

23 Lin H, Jin S, Hu Q, et al. Constructing B-spline solids from tetrahedral meshes for isogeometric analysis. Computer Aided Geometric Design, 2015, 35 : 109–120.

24 Lin, H, Wang G, Dong C. Constructing iterative non-uniform B-spline curve and surface to fit data points. Science in China Series F : Information Sciences, 2004, 47(3): 315–331.

25 Lin H, Bao H, Wang G. Totally positive bases and progressive iteration approximation. Computers and Mathematics with Applications, 2005, 50 (3-4) : 575–586.

26 Lin H, Zhang Z. An extended iterative format for the progressive-iteration approximation. Computers & Graphics, 2011, 35 (5) : 967–975.

27 Deng C, Lin H. Progressive and iterative approximation for least squares B-spline curve and surface fitting. Computer-Aided Design, 2014, 47 : 32–44.

28 Shi L, Wang R. An iterative algorithm of NURBS interpolation and approximation. Journal of Mathematical Research and Exposition, 2006, 26 (4) : 735–743.

29 Lin H, Zhang Z. An efficient method for fitting large data sets using T-splines. SIAM Journal on Scientific Computing, 2013, 35 (6) : A3052–A3068.

30 Cheng F, Fan F, Lai S, et al. Loop subdivision surface based progressive interpolation. Journal of Computer Science and Technology, 2009, 24 (1) : 39–46.

31 Fan F, Cheng F, Lai S. Subdivision based interpolation with shape control. Computer Aided Design & Applications, 2008, 5 (1-4) : 539–547.

32 Chen Z, Luo X, Tan L, et al. Progressive interpolation based on Catmull-Clark subdivision surfaces. Computer Grahics Forum, 2008, 27 (7) : 1823–1827.

33 Kineri Y, Wang M, Lin H, et al. B-spline surface fitting by iterative geometric interpolation/approximation algorithms. Computer-Aided Design, 2012, 44 (7) : 697–708.

34 Yoshihara H, Yoshii T,Shibutani T, et al. Topologically robust B-spline surface reconstruction from point clouds using level set methods and iterative geometric fitting algorithms. Computer Aided Geometric Design, 2012, 29 (7) : 422–434.

35 Okaniwa S, Nasri A, Lin H, et al. Uniform B-spline curve interpolation with prescribed tangent and curvature vectors. IEEE Transactions on Visualization and Computer Graphics, 2012, 18 (9) : 1474–1487.

36 Lin H, Qin Y, Liao H, et al. Affine arithmetic-based B-Spline surface intersection with gpu acceleration. IEEE Transactions on Visualization and Computer Graphics, 2014, 20 (2) : 172–181.

37 Lin H, Maekawa T, Deng C. Survey on geometric iterative methods and their applications. Computer-Aided Design, 2018, 95 : 40–51.

38 Farin G E. Curves and surfaces for CAGD: A Practical Guide. Morgan Kaufmann, 2002.

39 Yuan Y, Sun W, Theory and method of optimization. Beijing: Science Publishing House, 1997.

40 Burden R, Faires J. Numerical analysis. Thomson Brooks/Cole, Boston, 2001.

41 Knupp P. A method for hexahedral mesh shape optimization. International Journal for Numerical Methods in Engineering, 2003, 58 (2) : 319–332.