Computer-Aided Design 146 (2022) 103228

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Construction of IGA-suitable Volume Parametric Models by the Segmentation–Mapping–Merging Mechanism of Design Features

Long Chen [a,*], Ningyuan Bu [a], Yao Jin [b], Gang Xu [c], Baotong Li [d]

[a] School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai 200093, PR China
[b] School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, PR China
[c] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, PR China
[d] Key Laboratory of Education Ministry for Modern Design&Rotor-Bearing System, Xi'an Jiaotong University, Xi'an 710049, PR China

ARTICLE INFO

Article history:
Received 8 June 2021
Received in revised form 26 November 2021
Accepted 28 January 2022

Keywords:
Volume parametric models
Segmentation–Mapping–Merging
Complex shapes
Feature network
Isogeometric analysis

ABSTRACT

Volumetric parameterization is the key and bottleneck issue in the current research of constructing complex models for isogeometric analysis(IGA). Many researchers used reconstruction methods to convert the models presented in the boundary form such as B-rep, constructive solid geometry(CSG) or the finite element(FE) meshes. In these methods, recognition and understanding of topology is a difficult and unavoidable issue. What is more, the reconstruction methods are not compatible with the current modeling methods and processes adopted in the prevailing CAD software. In this paper, a method is proposed to construct the complex volume parametric models based on the Segmentation–Mapping–Merging mechanism of the design features. The semantic feature network is obtained by the interactive input or the feature elements extracted from existing CAD models, which allows great geometric flexibility. Then the semantic feature network is partitioned into the geometric feature networks and the complete feature networks under the constraints of volume parametric modeling. Each complete feature network is constructed as a volume parametric patch by volume parametric mapping. Finally, the volume parametric patches are merged into a whole model and are adjusted to be suitable for IGA. Some complex shapes including the mechanical parts, free-form models are presented to verify the effectiveness, availability and robustness of our method.

1. Introduction

The parametric design based on the parametric model which is represented by the boundary such as B-rep, CSG is the core technology of most CAD software, because of the variability and reusability of the parametric model [1–3]. But the parametric model should be transformed into non-parametric mesh model to adapt to analysis and optimization. So, during the whole design process, the parametric models and the non-parametric models need to be converted back and forth, which takes a lot of time. Meanwhile, it is hard to establish the direct analytic relationship among the geometric parameters, the material parameters and the performance parameters, which makes the integration of modeling, analysis and optimization be a great challenge.

Various methods have been developed to overcome the difficulties in the model conversion among modeling, simulation and optimization. Among them, the IGA method based on the volume parametric model can realize the seamless integration of CAD and CAE [4]. Because of the tensor properties of spline basis, a region that is not quadrilateral or hexahedral in topology cannot be represented by a single spline surface or volume [5]. Therefore, the model suitable for IGA must be volume parameterization as a set of quadrilateral surfaces in two-dimension or hexahedral volumes in three-dimension. While the geometric models designed by most CAD software are based on the boundary representation, so these models do not meet the requirements of IGA.

The lack of available software to construct IGA-suitable models is also a great challenge. In the existing literature, there are two ways to solve the problem above. The first is to integrate CAD with existing commercial finite element software like Abaqus [6–8], trying to bridge the gap between design and analysis with the help of a mature commercial platform. The second is to develop an open software package [9,10], generating polycube-based mesh by the given boundary and promoting the use of IGA significantly.

In this paper, a method is proposed to construct complex volume parametric models based on the mechanism called Segmentation–Mapping–Merging of the design features. The method consists of three steps: construction and segmentation of design features, volume parametric mapping and merging of parametric patches. To allow great geometric flexibility, the semantic

* Corresponding author.
E-mail address: cl@usst.edu.cn (L. Chen).

https://doi.org/10.1016/j.cad.2022.103228
0010-4485/

feature network can be constructed by interactive input. Then the semantic feature network is partitioned into multiple simple sub-network under the constraints of volume parametric modeling. Then the volume parametric mapping constructs each sub-network as a volume parametric patch. Finally, under the constraints of IGA, a volume parametric model is obtained by optimizing and merging operations. The complex models including mechanical parts and free-form models are selected to test and verify our method. The contributions can be concluded here.

- A platform of integration design based on volumetric parameterization is put forward. The ideas and methods of parametric design and feature design, which are widely used in the current CAD and CAE software, are also integrated together.
- A modeling mechanism or operation called segmentation–mapping–merging for the volume parametric patches or primitives is studied. Based on the mechanism, the models suitable to IGA can be directly created other than conversion from the models represented in boundary surfaces.
- The proposed method is realized and verified on the complex three-dimensional mechanical parts and free-form models. This gives an expectation to an extensive application for volume parameterization and IGA on complex shapes.

2. Related works and overviews

Seamless integration of CAD and CAE is a bottleneck problem in today's product design and has attracted interest of many researchers. To solve this problem, Gujarathi [11] put forward the concept of "Common Data Model", containing all the parameter information needed for modeling and analysis to connect CAD and CAE from data. Wassermann [12] and Schillinger [13] proposed the "DESIGN-THROUGH-ANALYSIS" method, which explored hierarchical refinement of NURBS as a basis for adaptive isogeometric and immersed boundary analysis. Cohen [14] proposed analysis-aware modeling to construct analysis-suitable CAD models directly, where the quality of the model and the accuracy of the results of IGA were improved. There are also some works using the analysis-oriented modeling method to create or convert the model automatically. In order to reduce the complexity of the mesh and improve the mesh quality, Li [15] automatically simplified the model by removing some features, but it was difficult to generate hexahedral meshes automatically.

There is another method called the finite cell method(FCM) [16–19], which is motivated by octree-based all-hexahedral mesh generation or image-based isocontouring method [20] for hex mesh generation. This method is suitable for B-rep models [21], composite domains made up of heterogeneous materials [22] and microstructure materials [23], and even used for volumetric data [24]. The boundary of the analysis model is approximated through the adaptive voxels that are placed in the quadtree (2D) and octree (3D) data structures.

Among all the methods for constructing analysis-suitable models, the volume parametric modeling method gains more and more attention [25]. It possesses many advantages in converting a model to a parametric volume, including providing good smoothness, using less storage space and avoiding the "gluing" and "staircase" problems existing in the FEM discrete meshes. Based on the parametric volume, the IGA method is directly applied to the model. Compared with other methods, IGA has many advantages when being applied to the product design and has gained many research interests since it was first proposed by T. J. R. Hughes in 2005 [26] and NURBS-based IGA is an important branch [27].

Current research on volume parametric modeling mainly focuses on several key aspects. The first is how to express the parametric volume by using different spline basis including B-spline, non-uniform rational B-spline (NURBS) [28], T-splines [29] and PHT splines [30]. The second is how to construct a parametric volume suitable for IGA from a boundary representation model, such as point cloud model, polygonal mesh model, the model with standard CAD formats such as STEP or IGES, or FEM mesh model [31,32]. The third is how to improve the parameterization quality by using different optimization methods which have different influences and efficiency on the analysis results, such as isoparametric orthogonal optimization [33], harmonic functions [34], Jacobian optimization [35], curve drift [36], and other methods.

There exist two ways to construct a volume parametric model. One is constructing the model from the given model. But in these methods, the topological segmentation of the complex models is a bottleneck problem [37–39]. Xiao [40] used a polysquare structure in the construction of the two-dimensional complex parametric model. Liu [41] used the skeleton curves of the model to generate the polycube structure inside the model and then obtained the volume parametric patches through the mapping between the polycubes and the model surfaces. Fu [42] constructed the polycube structure by segmentation and mapping of the tetrahedral mesh model. Hu [43] proposed a two-step surface segmentation algorithm by using gCVT and then obtained the hexahedral and T-spline models through polycube mapping. Chen [44] generated the polycube structure by minimizing the l_1-norm of the normals on the input triangular mesh model and obtained a parametric volume for complex shapes without internal singularity points. It is still troublesome to create the volume parametric models such as mechanical parts in the reconstruction approach [45].

The other way to create or generate a volume parametric model is using the interactive parametric construction method. There are also some researches focusing on this topic, such as the parametric design method [46], the Boolean operation method for parametric models [47], and the modeling method by using modeling software. These methods are trying to use interactive drawing and feature modeling methods to create new volume parametric models.

In this paper, a construction method for volume parametric models is proposed based on the mechanism of Segmentation–Mapping–Merging, which could combine the feature design and the parametric design. The modeling process includes the construction and segmentation of the feature network, the volume parametric mapping, and the merging of patches. The modeling process is shown in Figs. 1 and 2.

3. Basic definitions and generation of the semantic feature network

The input of the feature network is the set of various size parameters which describe the topological structure and the shape of the model. They are converted from interactive input or extracted from the existing models.

3.1. Basic definitions

Design feature is a kind of semantic information to describe the global and local details of geometry in the aspects of size, shape and topology. Here are the relevant definitions.

- Feature point *Fp*: a point to describe a place with special meaning of feature curve or feature surface.

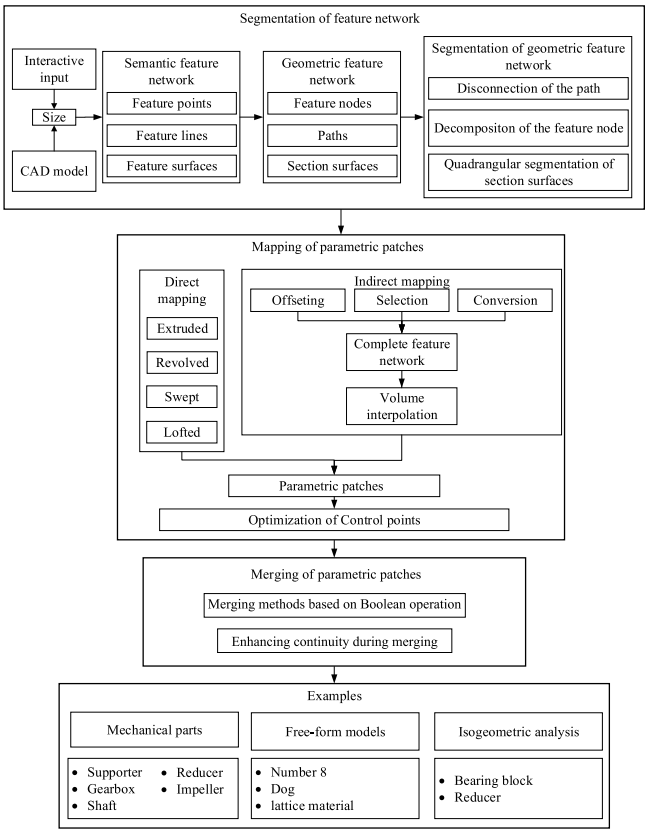

Fig. 1. Parametric modeling process of complex models.

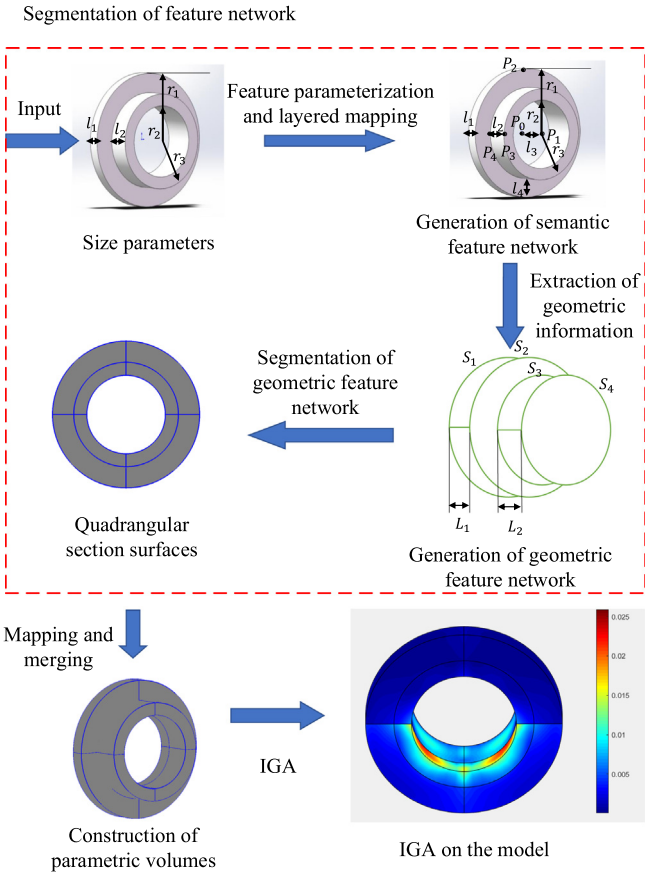

Fig. 2. Volume parametric modeling process.

- Feature curve *Fc*: a closed or open curve to describe the boundary of a feature surface or the generation path of the model.
- Feature surface *Fs*: a closed and N-sided surface to describe an area with special meanings.
- Feature node B_n: the intersection point of multiple paths.
- Path *L*: a feature curve to describe the movement direction of a shape. The start and the end control point are defined as the anchors to edit it.
- Section surface *S*: one of the six boundary surfaces from which a parametric volume can be obtained by the volume interpolation method.
- Feature network *F*: a network composed of fore-defined features to describe the main topology, shape and sizes of a model. It falls into three groups called the semantic feature network *Sem_F*, the geometric feature network *Geo_F*, and the complete feature network *Comp_F*. The first is composed of feature points, curves and surfaces. The second is composed of feature nodes, paths and section surfaces. The third denotes the six enclosed section surfaces through which a parametric volume can be interpolated.

3.2. Generation of the semantic feature network

The purpose of constructing the semantic feature network is to parameterize the features and establish layered control of the model. The feature surfaces, curves and points contain a lot of semantic information, which are not suitable for modeling, so the input size parameters are used to represent these features and mapped into multiple layers for ease of modification.

- The high-level semantic layer *Fh* uses as least parameters as possible to describe the overall size, shape and topology of the geometry, such as the length and width of a rectangle, the radius of a arc, and so on. The high-level layer is exposed to users to modify the overall sizes of the model.
- The middle-level semantic layer *Fm* uses relatively more parameters to describe the shape of a geometry, such as the distance from the starting point to the curve inflection point. The middle-level layer is encapsulated.
- The low-level semantic layer *Fl* describes the underlying geometric information of a surface or curve, such as the corner point on a surface, the extreme value point on a curve. To avoid parameter redundancy, the feature points other than the control points of a curve or surface are taken as the underlying parameters.

According to the above definitions, the semantic feature network can be written as:

$$Sem\_F = \{Fs, Fc, Fp\} = \{Fh, Fm, Fl\} \tag{1}$$

Three parameter sets are defined on *Fh*, *Fm*, *Fl* respectively as $\{P_{h,i}, i \in [1, H]\}$, $\{P_{m,j}, j \in [1, M]\}$ and $\{P_{l,k}, k \in [1, N]\}$. Two mapping operations are also defined among these parameters. $\Phi_1 = \{F_{hm}ij\}$ is the mapping from $P_{h,i}$ to $P_{m,j}$. The item $F_{hm}ij$ presents the *i*th parameter in $P_{h,i}$ how to contribute to the *j*th parameter in $P_{m,j}$. $\Phi_2 = \{F_{ml}jk\}$ is the mapping from P_m to P_l. The item $\{F_{ml}jk\}$ presents the *j*th parameter in $P_{m,j}$ how to contribute to the *k*th parameter in $P_{l,k}$.

The mappings also reflect the constraints among the multilayered parameters. The modeling process is to establish the mapping operations, accordingly the mapping from high-level parameters to low-level parameters can be realized. As a result, it is convenient for users to create and modify the local features with a few high-level parameters or design parameters.

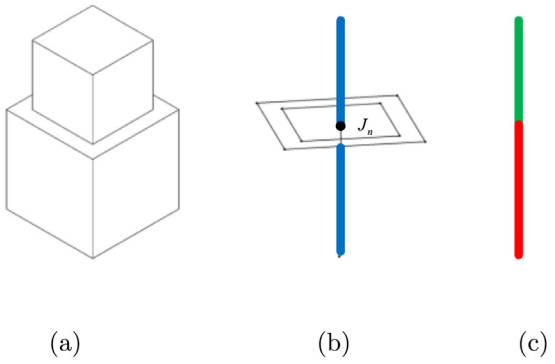

Fig. 3. Path disconnection: (a) a model with two patches on the same path; (b) two geometric feature networks for the model; (c) the paths after disconnection.

4. Generation and segmentation of the geometric feature network

The geometric feature network can be extracted from the semantic feature network. During the extraction, some of the feature curves are treated as paths, and the intersections of paths are taken as feature nodes. The unclosed feature surfaces are selected as the section surfaces. Considering all the cases, the geometric feature network falls into four types:

- Type I, it has multiple section surfaces and zero path.
- Type II, it has a single section surface and one path.
- Type III, it has multiple section surfaces and a single path.
- Type IV, it has multiple section surfaces and multiple paths.

The more difficult cases will occur for the latter two types. In these two cases, the geometric feature networks should be segmented by a series of operations, finally all the geometric feature networks will fall in the type I and the type II. For the case of type I, it will be easier to generate the complete feature network only by adding some section surfaces directly from the existing section surfaces and interpolating the parametric volume with the method presented in Section 5.2. For the case of type II, the parametric patches will be obtained by direct modeling in Section 5.1.

4.1. Disconnection of the intersecting paths

Since there is no path in the final complete feature network, all the paths should be treated to help create the section surfaces. So in type III, if one path is related to multiple section surfaces, the path should be disconnected. As shown in Fig. 3(a), there are two section surfaces related to one path, so the model has two parts, upper and lower, as shown in Fig. 3(b). At the indicated point *Jn*, the path is broken, as shown in Fig. 3(c). Then the broken path is used to create the extruded volumes.

4.2. Decomposition of the feature node

In type IV, some paths converge at one feature node. In this case, the feature network should be segmented at the feature node to make the paths be separated, and each section surface should be assigned to its corresponding path. In this way, the feature network of type IV will be converted into to type III. As shown in Fig. 4(a), a geometric feature network F^{B_n} containing only one feature node B_n can be expressed as follows:

$$Geo\_F^{B_n} = \{S_1, S_2, \ldots, S_m, L_1, L_2, \ldots, L_m, B_n\}$$

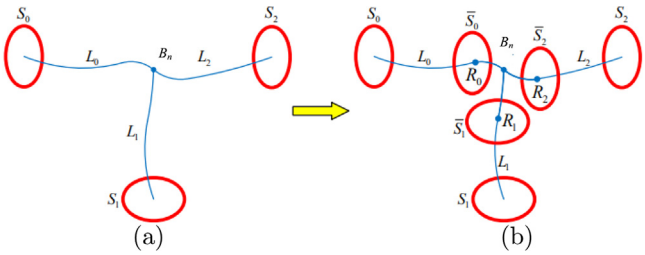

Fig. 4. Decomposition of the feature node.

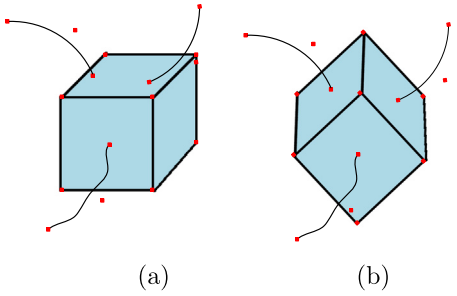

Fig. 5. Use hexahedral block to segment paths: (a) hexahedral block before optimization (b) hexahedral block after optimization.

Where $S_i, i \in \{0, m\}$ is the section surface, and $L_j, j \in \{0, n\}$ is the path, m and n are the number of section surfaces and paths respectively. The intersecting paths are should be separated at the feature node B_n to make each part contain at most one path, as shown in Fig. 4(b). Some additional faces $\bar{S}_i$ need to be inserted to enclose the segmented part.

The method called path-aware hexahedral block method [48] is enhanced to segment the intersecting paths. For the convenience of modeling, we subdivide this block according to the directions and the number of the intersecting paths on each surface, making each surface as orthogonal as possible with respect to the directions of incoming paths, shown in Fig. 5. For intuitive understanding, the paths through the hexahedral block are named as incident curves. The following optimization problems is required to solve on each B_n:

$$\min \sum_{j=1}^{k} \left|d_j U\right| + \left|d_j V\right| + \left|d_j W\right| \tag{2}$$

Here, *UVW* is the orthonormal basis, d_j is the direction of the *j*th incident curves. Different from other linear optimization problems, Eq. (2) contains absolute values and the quadratic constraints of the orthonormal basis *UVW*.

$$s.t. \begin{cases} |U| = |V| = |W| = 1 \\ UW = UV = VW = 0 \end{cases}$$

For simplicity, we employ a smooth approximation of the l_1 norm introduced in [49], in this way, the l_1 minimization is turned into a smooth optimization problem. The element $|x|$ is replaced by $\sqrt{x^2 + \varepsilon}$, where $\varepsilon > 0$ is a regularizing parameter and it is decreased gradually throughout the minimization to balance smoothness and accuracy. Specific mathematical proofs of this method can be found in [50,51]. So Eq. (2) can be replaced with Eq. (3), then Eq. (3) is minimized with ε >0 and approaching 0.

$$\min \sum_{j=1}^{k} \sqrt{(d_j U)^2 + \varepsilon} + \sqrt{(d_j V)^2 + \varepsilon} + \sqrt{(d_j W)^2 + \varepsilon} \tag{3}$$

This minimization is considered as a nonlinear constrained problem. The basis *UVW* is initialized by computing the PCA of

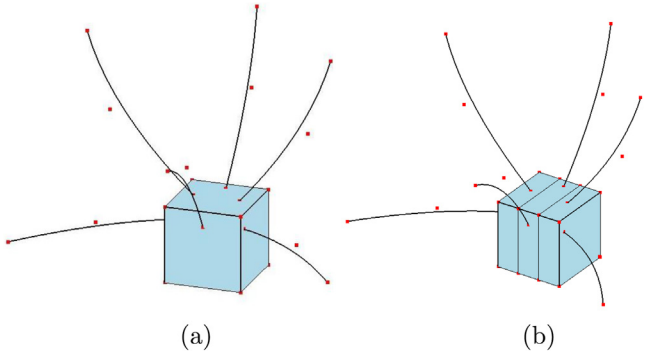

Fig. 6. Hexahedral block: (a) hexahedral box before subdivision (b) hexahedral sub-boxes after subdivision.

the set of directions $\{d_1, d_2, \ldots, d_k\}$. The problem starts to be resolved with an empirical value $\varepsilon = 0.2$. Each solution is taken as a warm start for the next iteration by halving the value of ε. Experimentally, five iterations will be good to end the optimization process. After optimization solution, a set of hexahedral boxes which are oriented associated with each incident curves will be obtained. However, if there are multiple incident paths on one face, the box is still needed to be split again. All the intersection points on this face f and the number n of these points are needed to be computed. f is split into n parallel rectangular facets. The directions of the sub-rectangular are decided by projecting the intersection points to the two cardinal directions parallel to the sides of f. Finally, each face of f is assigned to its corresponding path. The result is shown in Fig. 6.

The feature network F^{Bn} is now:

$$Geo\_F^{Bn} = \{S_1, S_2, \ldots, S_m, L_1, L_2, \ldots, L_k, \overline{S}_1, \overline{S}_2, \ldots, \overline{S}_l\}$$

Where k is the number of the paths of the whole block, which is greater than or equal to m. $\overline{S}_i$ are the boundary surfaces of the hexahedral boxes. Considering the close distance between S_i and $\overline{S}_i$, the path L_i between them can be regarded as straight lines. Until now, the intersecting paths are cut at B_n and the hexahedral boxes will be applied in Section 5.2.3.

4.3. Quadrangular segmentation of section surfaces

In the common case, IGA requires that the volume parametric patches should be hexahedrons and share exactly the same adjacent quadrangular surface. For a geometric feature network $Geo\_F = \{S_1, S_2, \ldots, S_m, L_1, L_2, \ldots, L_n\}$, these sections surfaces should be all quadrilateral patches and have no genus. So a quadrangular segmentation method is proposed to deal with section surfaces in type III and type IV to obtain quadrilateral patches and make sure the adjacent surfaces should have one consistent boundary curve.

The input of the quadrangular segmentation is the contours of the section surfaces, shown in Fig. 7(a), which have both outer contours and inner contours. In our method, the section surfaces are quadrangulated by segmenting their contours based on some curve features. A geometric containing tree is constructed to represent the hierarchy of inclusion relationship among the contours that are considered as nodes of the tree, shown in Fig. 7(b). The root node is the first outer contour, and its children are the inner contours. The odd and even layers correspond to the outer and inner contours. A connected domain is formed by the inner and outer contours of each two layers. There exists a connected domain with only outer contour but no inner contour.

Since all contours are segmented, the segmentation operation is performed layer by layer, starting from the contours of the leaf

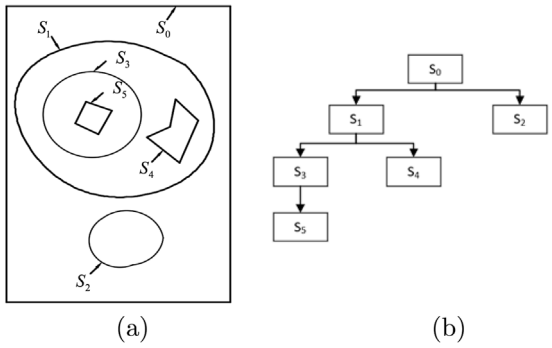

Fig. 7. Geometric containing tree.

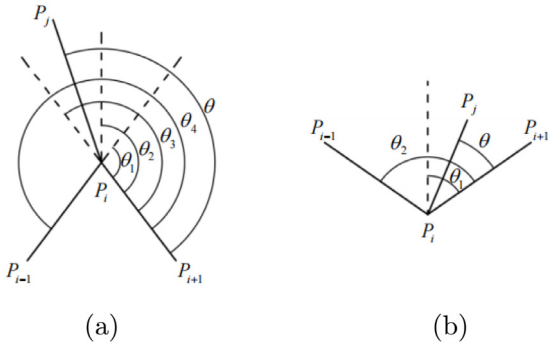

Fig. 8. Concave and convex points on the boundary: (a) parameters in $f(\theta)$; (b) parameters in $g(\theta)$.

node. For the connected domain with multiple genus, it should be converted into the domain with zero genus by adding some cutting lines as guidelines. The weighting method is used to select cutting lines via pre-processing for the segmentation. At first, traverse the geometric containing tree in a hierarchical order. Next, add a straight line as the cutting line between the existing curve endpoints of different geometric domains. Considering the quality of the quadrilateral patches in isogeometric analysis, we calculate the weight of each feasible cutting line, and then select the optimal one.

From the literature [34], it is clear that the angle between adjacent edges has a significant effect on patch quality. And the concave points are preferred to connect, so the weight of each cutting line is as follows:

$$f(\theta) = \begin{cases} \theta_4 \cdot \frac{\theta}{\theta_1} & 0 < \theta \le \theta_1 \\ \theta_4 \cdot \left(1 + \frac{\theta - \theta_1}{\theta_2 - \theta_1}\right) & \theta_1 < \theta < \theta_2 \\ \theta_4 \cdot \left(2 - \frac{\theta - \theta_2}{\theta_3 - \theta_2}\right) & \theta_2 < \theta \le \theta_3 \\ \theta_4 \cdot \left(1 - \frac{\theta - \theta_3}{\theta_4 - \theta_3}\right) & \theta_3 < \theta \le \theta_4 \\ 0 & \text{others} \end{cases}$$

$$g(\theta) = \begin{cases} 2^{\left[\frac{\theta_2}{\pi/3} - 1\right]} \cdot \theta_2 \cdot \frac{\theta}{\theta_1} & 0 < \theta < \theta_1 \\ 2^{\left[\frac{\theta_2}{\pi/3} - 1\right]} \cdot \theta_2 \cdot \left(1 - \frac{\theta - \theta_1}{\theta_2 - \theta_1}\right) & \theta_1 \le \theta < \theta_2 \\ 0 & \text{others} \end{cases}$$

$$w_i = \begin{cases} f(\theta) & P_i \text{ is a concave point} \\ g(\theta) & P_i \text{ is a convex point} \end{cases}$$

$$w_{ij} = w_i + w_j$$

Here, i and j are the indexes of two endpoints. Other parameters are shown as Fig. 8. When θ is too large or too small, the

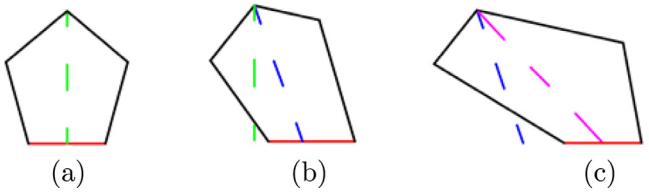

Fig. 9. Segmentation of the curve $\delta = 1$.

weight will become small, so as to avoid sharp angles. When the contours are curves, we take the tangent direction to calculate these angles. The larger the weight, the more the cut line meets the requirements. Finally, the cutting lines are selected according to the weights. After obtaining the cutting lines, the quadrangular segmentation can be performed. δ is defined to denote whether a contour can be segmented. If δ is equal to 0, it means that the curve cannot be subdivided. Before processing the contours, δ is 0 if the contour is an inner contour, otherwise it is 1. For the contour of a convex quadrilateral, the δ is modified to zero, and stored into the set Q.

Assuming that each contour is connected in straight lines, the geometric centers points O_i for all the contours are calculated, along with the outermost contour O. All the center points are sorted by the values $\{|O - O_i| \parallel i \in [0, n]\}$ from smallest to largest, along with all the contours being sorted in the same order. All the quadrilateral patches can be obtained one by one by complying the following algorithm.

(1) All pairs of visible points in a contour are found and stored in the set V. If a sharp angle is produced when the visible points are connected, the pair should be moved from V to set U.
(2) For a curve on one contour whose $\delta = 1$, the visible point formed by two endpoints of the curve is found. The perpendicular point from the visible point to the dividable curve is denoted by P_v. When P_v is on the dividable curve and is not an endpoint, as shown in Fig. 9(a), the red curve is the dividable curve and the green dotted line is the vertical line with the valid connection. When P_v is not on the dividable curve, the intersection of the angular bisector and the dividable curve is denoted by P_b. As shown in Fig. 9(b), the dotted blue line represents the angular bisector. The connection is valid when P_b exists and is not an endpoint. If P_b is nonexistent, the midpoint of the dividable curve is selected as the intersection point for wiring, as shown in Fig. 9(c). According to [52], the weight of the connecting line can be calculated. Then put the connecting line into the set L.
(3) The weights of the connecting lines in V, U are calculated as well. For V, U, L, if there is a concave point in the pair of visible points, such pairs are sorted by their weights from large to small, the others are sorted in the same way and placed at the back of the set.
(4) The choice of connecting lines of a contour is prioritized in $V - L - U$ order. If a connecting line belongs to L, according to the proportion of the distance between the intersection point and the two endpoints, the segmentation point is taken on the original curve and connected with the view-able point.
(5) Two new contours are obtained after each connecting line is created. If the contour is a convex quadrilateral contour, put it into set Q. If the result is a contour whose number of edges is greater than 4, its subdivision priority is raised to the highest.

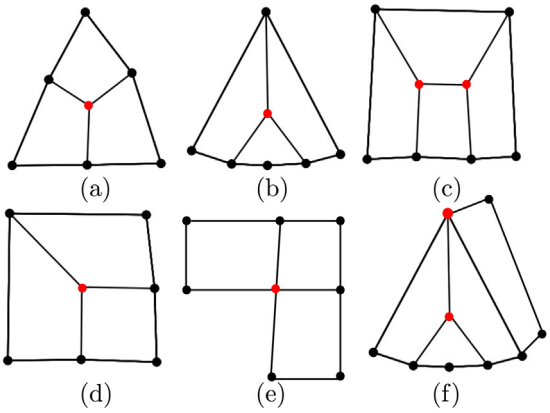

Fig. 10. A set of polygon regions which containing extraordinary vertices: (a) and (b) are quasi triangle type; (c) and (b) are quasi rectangle type; (e) and (f) are other combinations. The endpoints are marked in black and the extraordinary vertices are marked in red.

(6) If no valid result can be obtained by traversing $V - L - U$, tracing back to the upper contour layer, the connection with lower priority should be selected until all contours are convex quadrilateral contours.
(7) The inclusion relationships of all convex quadrilateral contours to the original inner and outer contours are determined, so that each contour can be associated with a path. The Coons interpolation method is used to interpolate the four boundary curves of a contour into a NURBS surface, and a parametric volume is generated according to the surface and the corresponding path.

The placement of extraordinary vertices is indeed tough work. In [53], sparsely distributed directional constraints are used to automatically determine the extraordinary vertices. In [54], the extraordinary vertices are obtained by enumerating all the mesh topologies in a patch, which is similar to our method. Based on our current method called the enumeration subdivision, the goal is to produce as few quadrilateral patches as possible because of the huge size of stiffness matrix in isogeometric analysis, and try to ensure the quality of quadrilateral patches. Via the subdivision mechanism, the distribution of all extraordinary vertices can be obtained by enumeration.

Extending from [52], a set of polygon regions with the minimum number of edges which contains extraordinary vertices can be listed. According to the location of endpoints, they are divided into the two categories: quasi triangle type, shown in Fig. 10(a) and (b); quasi rectangle type, shown in Fig. 10(c) and (d). Other regions can be combined by the two cases and quadrilateral patches, shown in Fig. 10(e) and (f). Therefore, the regions mentioned above should be avoided in segmentation to reduce the number of extraordinary vertices.

There are some quadrilateral segmentation examples shown in Fig. 11. From left to right, the pictures denote the section surfaces to be segmented, the section surfaces added some cutting lines, the result of segmentation and their Jacobian values.

5. Complete feature network and volume parametric mapping

The complete feature network can be achieved after the segmentation operation of the geometric feature network. Then the volume parametric patches can be created by the mapping operation in the following two ways.

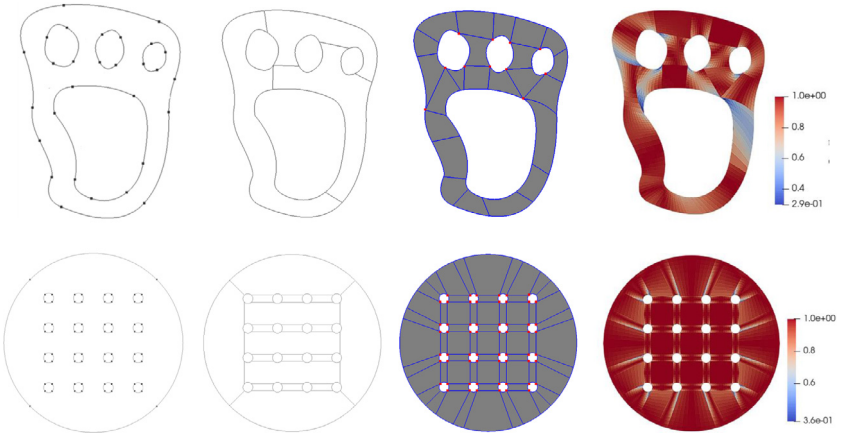

Fig. 11. The examples of quadrangular segmentation.

5.1. Direct modeling

Feature modeling is commonly used in mainstream modeling software. It is used to construct the parametric feature primitives including the extruded volume, revolved volume, swept volume, and lofted volume.

5.1.1. Extruded volume

For the extruded volume, the path $L(w)$ is a straight line. The extruded volume can be obtained by translating the section $S(u, v)$ along $L(w)$. Fig. 12(a) shows an extruded volume.

5.1.2. Revolved volume

A revolved volume is shown in Fig. 12(b). Compared with the modeling of the extruded volume, the path $L(w)$ of the rotated volume is a quadratic circle. Since the weights of the control points are not always ones, they need to be pre-emitted when calculating the control points. The control points are calculated as follows.

$$P_{i,j,k}=T \cdot Rx \cdot Ry \cdot Rz \cdot M \cdot (-Ry) \cdot (-Rx) \cdot (-T) \cdot P_{i,j}$$

Where T is the translation matrix relative to the origin. Rx, Ry, Rz are the rotation matrices around the corresponding axes. M is the weighted pre-elimination matrix, which is given by

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1/\omega_{i,j} & 1/\omega_{i,j} & 1 & 1 \end{bmatrix}.$$

5.1.3. Swept volume

A swept volume $V(u, v, w)$ can be created by sweeping a section surface $S(u, v)$ along a curve path $L(w)$. The different sections with number K along $L(w)$ are interpolated and fitted. Then, the control points are calculated according to the fitted sections. The number of K needs to be determined. Generally, the larger K is, the more consistent with the variation trend of the obtained model along the path $L(w)$. It is common practice to make the number of knot vectors in the W direction $N_{\bar{W}^{knots}} = K + r + 2$ by increasing the number of knots in the W direction or increasing the number of control point of sections. The control point sections can be obtained by the affine transformations of S (u,v) through the Frenet frame as follows.

$$B(w) = \frac{L'(w) \times L''(w)}{|L'(w) \times L''(w)|}$$

The Frenet frame defines a local orthogonal coordinate system that ensures the stability of the affine transformation of $S(u, v)$. The section $S(u, v)$ is then transformed to the positions $L(w_k)$ $(k = 1, \ldots, K)$. After the transformation, the corresponding point of the control point $P_{i,j}$ at w_k is

$$Q_{i,j,k} = \omega_k \cdot T_2 \cdot T_1 \cdot P_{i,j}$$

Where ω_k is the weight of the corresponding control point of $L(w)$; T_1 is the projection transformation; and T_2 is the Frenet frame transformation.

After obtaining all the control point sections, we can find the control point with the weight $P^{(\omega)}$ by solving the equation

$$P^{(\omega)} \cdot N = Q$$

Since the weight of $P^{(\omega)}_{i,j,k}$ is $\omega_{i,j,k}$, the control point of the swept volume, $V(u, v, w)$ is $P_{i,j,k} = P^{(\omega)}_{i,j,k}/\omega_{i,j,k}$. A swept volume obtained by sweeping a circular section along an arc path with K=10 is shown in Fig. 12(c).

5.1.4. Lofted volume

When lofting the different section surfaces, all of those parameters including orders, knot vectors, and the number of control points must be unified. By using degree elevation, the orders of all sections in the same parameter direction are unified to the highest order among them. After that, the knot vectors of the same parameter direction are combined into a union set, and the corresponding knot vectors of each surface are refined into as same as the set. A lofted volume $V(u, v, w)$ is shown in Fig. 12(d). In the figure, the red sections are the given section surfaces.

5.2. Indirect modeling

In indirect modeling method, complete feature network can be generated by using one of the three operations including offsetting, selection and conversion. Then a parametric volume can be interpolated through the six section surfaces by using the method called Coons interpolation [33].

5.2.1. Offsetting operation

For the feature network $F = \{S_1, S_2, \ldots, S_m, L\}$, there are some constraints to prevent them from being swept or lofted along the path. For example, according to the feature network shown in Fig. 13(a), modeling with a sweeping operation produces the result shown in Fig. 13(b). Note that the lower section surface S_1 is given in the feature network, but the model in Fig. 13(b) does not satisfy the requirements of the complete feature network. The path L is offset to the corners of the unconstrained section, and the boundary curves of the unconstrained section are offset along the path, as shown in Fig. 13(c). Finally we construct the volume parametric model according to the Coons method, as shown in Fig. 13(d).

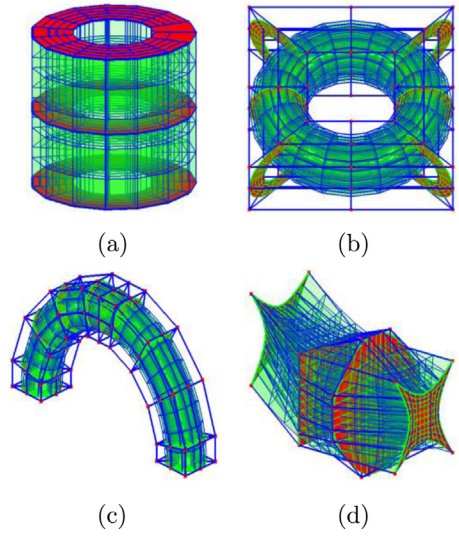

Fig. 12. NURBS volume parametric modeling of features: (a) extruded volume; (b) revolved volume; (c) swept volume; (d) lofted volume.

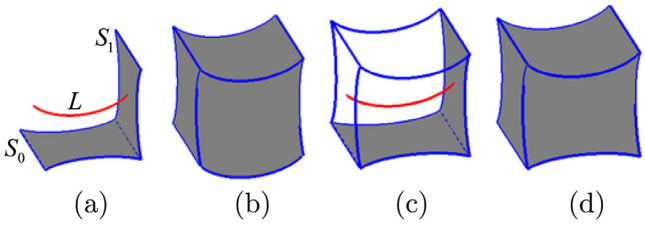

Fig. 13. Offsetting method.

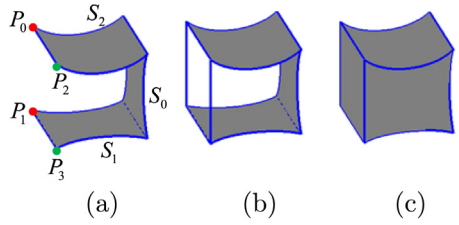

Fig. 14. Selection method.

5.2.2. Selection operation

For the feature network $F = \{S_1, S_2, \ldots, S_m\}$ without a path, the corner points are selected interactively from the feature network to generate the ridge lines by linear interpolation, and then the volume parametric model is generated by interpolation. As shown in Fig. 14(a), the red corner points P_0, P_1 and the green corner points P_2, P_3 are selected to generate the ridge lines as shown in Fig. 14(b). The resulting volume parametric model is shown in Fig. 14(c).

5.2.3. Conversion operation

Conversion method is mainly used for constructing the structure of the junction, shown in Fig. 15. NURBS can only construct quadrilateral surfaces while toric surfaces can interpolate complex surfaces in the polygon domain [55].

In this section, the toric surfaces are sampled and fitted into NURBS surface according to the inner hexahedral structures. Finally, each parametric volume is constructed by interpolating the surfaces and the corresponding surfaces of the inner hexahedral structure.

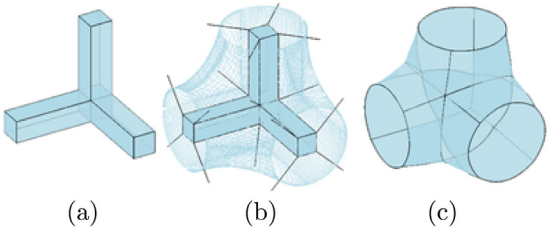

Fig. 15. Junction model: (a) construction of inner hexahedral structures (b) toric surface sampling (c) toric surface fitting.

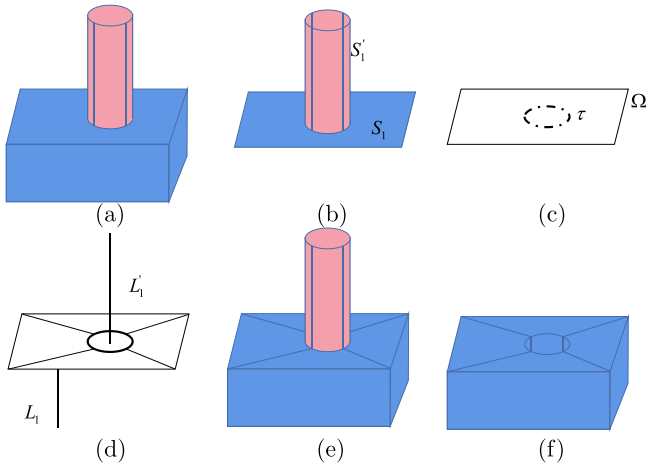

Fig. 16. A simple case of merging methods.

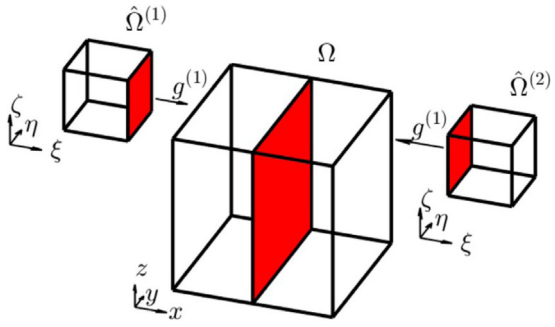

Fig. 17. Mapping from parametric space to physical space.

As mentioned in Section 4.2, the hexahedral box method is used to divide the intersecting paths. The geometric feature network is obtained after decomposition of feature node. The inner hexahedral structures are generated using $\overline{S}_i$ and L_i, where each surface is modeled as a swept body along a corresponding path, shown in Fig. 15(a). Then the sampling vectors and their intersections with the toric surfaces are obtained. There are two cases for the external vector at the corner point P of the hexahedral structure.

(1) The hexahedral structure in which P is located has one end face φ on the same plane as the end face of the model.

$$\vec{v}_P = -\sum_{i=0}^{m} \vec{v}_i$$

Where m is the number of boundary curves on the end face φ with P as the endpoint, and $\vec{v}_i$ is the unit vector of the ith edge from P to the other endpoint of the boundary curve.

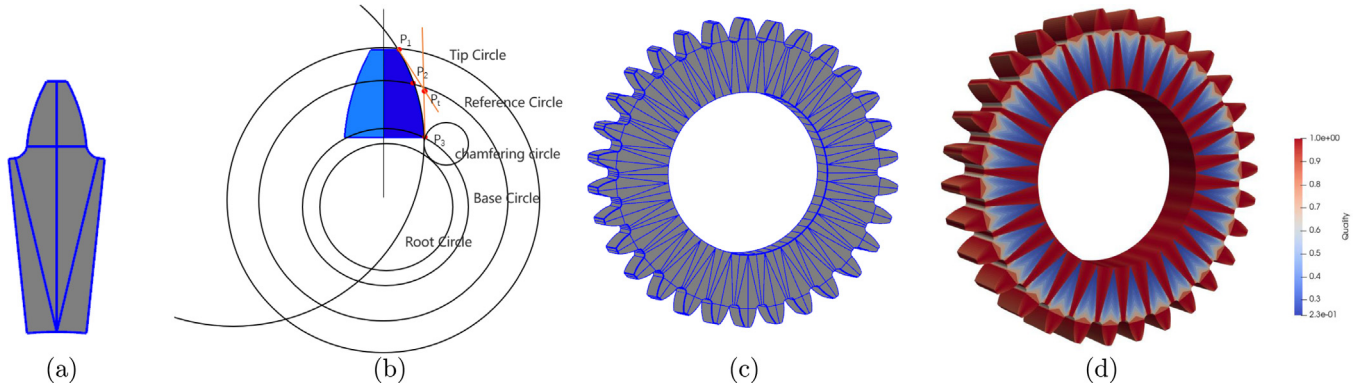

Fig. 18. Modeling of a standard cylindrical spur gear: (a) shows a single gear feature surface; (b) shows the parameters for approximating the profile; (c) shows the whole gear; (d) shows the Jacobian values of the model with the minimum value being 0.23.

(2) Any end face of the hexahedral structure in which P is located is not on the same plane as the end face of the model.

$$\vec{v}_P = -\sum_{j=0}^{n} \vec{v}_j$$

Where n is the number of all boundary curves with P as the endpoint, and $\vec{v}_i$ is the unit vector of the jth edge from P to the other endpoint of the boundary curve.

The two ends of the boundary curve $C_i(u)(i = 0, 1, 2, 3)$ correspond to the two external vectors $\vec{v}_P, \vec{v}_Q$ with the sampling vector $\vec{v}_k$ interpolated along the boundary curve.

$$\vec{v}_k = C_i(u_k) + Rotate(\vec{v}_R, \vec{v}_P, \alpha)(k = 0, \ldots, K-1)$$
$$u_k = k/(K-1)$$
$$\vec{v}_R = \vec{v}_P \times \vec{v}_Q$$
$$\alpha = \theta \cdot u_k$$

Here $C_i(u_k)$ is uniformly sampled in the parametric space, while the boundary curves of inner hexahedral structures are straight lines, so the sampling points are distributed uniformly on boundary curves as well; $Rotate(\vec{v}_R, \vec{v}_P, \alpha)$ means rotating $\vec{v}_P$ by angle α anticlockwise with v_r as the rotation axis; and θ is the angle between $\vec{v}_P$ and $\vec{v}_Q$.

This method results in K sampling vectors distributed on the boundary curve. We then use the Newton iterative [56] method to solve the intersections of the sampling vectors and the toric surfaces. Fig. 15 shows a junction model that can be easily obtained from feature network.

6. Merging of parametric patches

After all the parametric patches are generated, they should be merged to form a whole model. In our modeling process, we require that the adjacent two volume parametric patches share one complete surface. So enhancing the continuity among the patches is a very important thing and to be discussed here. If the designer wants to modify a built model, such as adding some features to the model or cutting some features from the model, in other words, how to merge newly generated features and the built model is also a problem. In conventional modeling methods, this operation can be finished by Boolean operation. Although we can redefine the feature network that includes the new features and rebuild the whole model, minimizing workload by keeping the unmodified part to be unchanged is still a good idea. So how to implement the Boolean operation in our modeling method is another issue to be discussed.

6.1. Enhancing continuity during merging

Since the shared surfaces between two patches have been extracted through segmentation, keeping these surfaces unchanged during interpolation is a good idea. The Continuity method [57] can be used to improve the continuity of the contact surfaces. As shown in Fig. 17, there are two parameter spaces $\hat{\Omega}^{(1)}, \hat{\Omega}^{(2)}$, mapping geometrically to a same physical space $\Omega = \Omega^{(1)} \cup \Omega^{(2)}$. In physical space, a common surface connects two patches $\tau = G^{(1)}(1, \eta, \varsigma) = G^{(2)}(0, \eta, \varsigma)$. In our work, C^1 continuity is required, so all the basis functions $N_i(\xi, \eta, \varsigma)$ possess C^0 continuity at τ need to be removed and replaced with C^1 continuity functions. The removed functions are called as $R_i(\xi, \eta, \varsigma)$. Set $\Re_i(x, y, z)$ as the corresponding functions in physical domain of removed functions. New function $\psi(x, y, z)$ is constructed to give a linear combination of $\Re_i(x, y, z)$:

$$\psi(x, y, z) = \sum_{i=1}^{n_{\Re}} c_i \Re_i(x, y, z)$$

Where c_i are the scalar coefficients of $\Re_i(x, y, z)$ and $n_{\Re}$ represents the number of removed functions. Suppose that $\psi_i(x, y, z) = \psi_i^{(1)}(x, y, z) + \psi_i^{(2)}(x, y, z)$, where $\psi_i^k(x, y, z)$ have support on patch k. The requirement of C^1 continuity condition is:

$$\nabla \psi_i^{(1)}(x, y, z) = \nabla \psi_i^{(2)}(x, y, z)$$

Set $\phi^{(k)}(x, y, z)$ in physical domain as the associated functions of $\psi^{(k)}(x, y, z)$, the continuity condition can be imposed through the surface of $\psi_i^k(x, y, z)$:

$$F^{(k)}(\xi, \eta, \varsigma) = (G^{(k)}(\xi, \eta, \varsigma), \phi^{(k)}(\xi, \eta, \varsigma))^T$$

The continuity constraint is as follows:

$$\det\left(\frac{\partial F^{(1)}(1,\eta,\zeta)}{\partial \xi}, \frac{\partial F^{(1)}(1,\eta,\zeta)}{\partial \eta}, \frac{\partial F^{(1)}(1,\eta,\zeta)}{\partial \zeta}, \frac{\partial F^{(2)}(0,\eta,\zeta)}{\partial \xi}\right) = 0$$

6.2. Merging methods based on Boolean operation

Similar to the conventional Boolean operation, the added feature is defined as real feature and the cut feature is defined as virtual feature. So two cases will turn out, including real–real merging and real–virtual merging. The literature [58] and [59] demonstrate the volumetric Boolean operations by exploiting algorithms of B-rep Boolean operations. T-spline is also considered for high quality volumetric Boolean operations [47]. Executing the Boolean operation on two solids with heterogeneous material is also very interesting [60]. Implementing a real–virtual Boolean

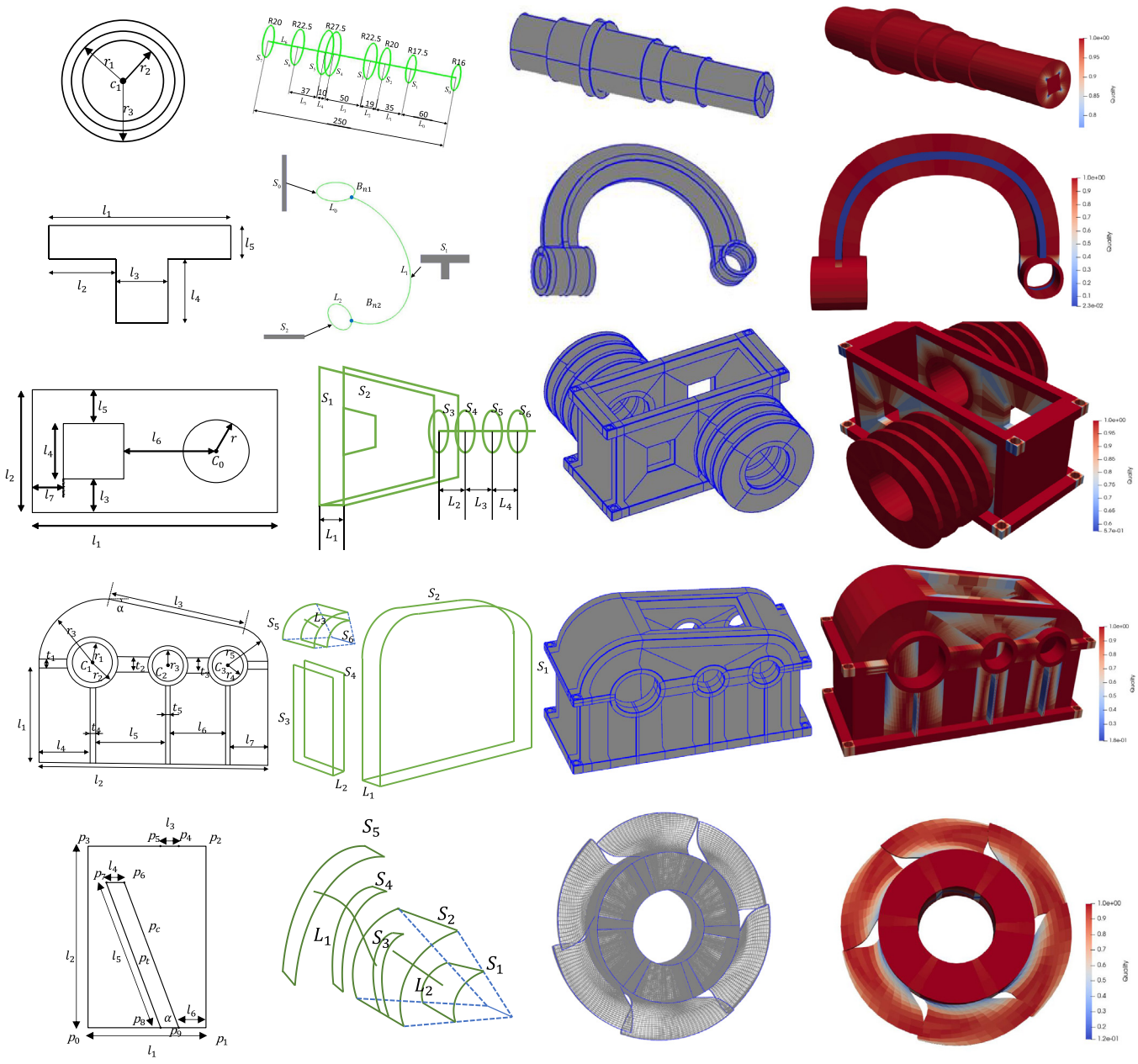

Fig. 19. Volume parametric modeling of non-standard mechanical parts.

operation on a simple volume parametric solid is not an easy thing [61].

Here we give a simple example to demonstrate our idea which will be well studied in the future. Instead of applying the Boolean operations on the generated parametric volume, which will lead to tedious and difficult work for searching the boundary of the volume, we still execute the Boolean operations on the feature network. In our method, volumetric Boolean operations are reduced to surface Boolean operations. In future work, we will extend this merging method to more complex models. As shown in Fig. 16(a), a cylinder and a hexahedron need to be merged, the merging steps are listed as follows:

(1) Extract the six surfaces of the volume parametric patches and determine which surfaces intersect. Then set the intersecting surfaces as S_1 and S_1', as in Fig. 16(b).
(2) Calculate the set of intersecting lines τ of S_1 and S_1', then put τ into the initial set of contours Ω, as in Fig. 16(c)
(3) Perform quadrangular segmentation of Ω and get the quadrilateral patches as the section surfaces in feature network.
(4) Obtain the corresponding paths L_1 and L_1' of the section surfaces shown in Fig. 16(d). If it is the case of real–real merging, the cylinder and hexahedron are constructed by the section surfaces and the paths shown in Fig. 16(e), otherwise the set of intersection lines τ is considered as genus in quadrangular segmentation, the path L_1 is used to constructed hexahedron, shown in Fig. 16(f).

7. Examples

To demonstrate the effectiveness of our methods, some typical mechanical parts and free-form models are constructed and evaluated by their Jacobian values. What is more, IGA is applied to verify the utilization of the generated models. Due to space limitations, the formulae for only some of the parameters are

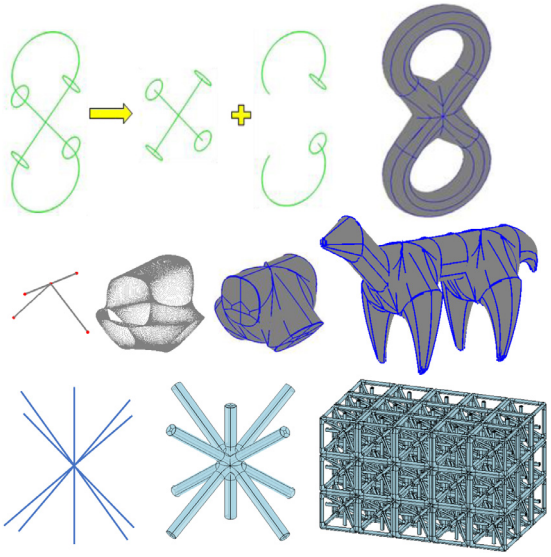

Fig. 20. Volume parametric modeling of free-form models.

given. We use C++ for modeling and OpenGL for rendering in VS2017.

7.1. Modeling of mechanical parts and free-form models

7.1.1. Standard mechanical parts

The cylindrical spur gear is a standard mechanical part whose shape is mainly defined by the 8 parameters: normal module m, number of teeth z, normal pressure angle α, coefficient of addendum h_{ax}, coefficient of bottom clearance c_x, modification coefficient x, gear axial width B, gear bore diameter D_x. Among them, the last two parameters are generally fixed and x is equal to zero. The first five parameters are selected as high-layer parameters. The overall shape of the spur gear is obtained, with a signal gear feature surface as example in Fig. 18(a).

The reference diameter $D = mz$, addendum $h_a = (h_{ax} + x)m$, dedendum $h_f = (h_{ax} + c_x - x)m$, addendum circle diameter $D_a = D + 2h_a$, base circle diameter $D_b = D\cos\alpha$, dedendum circle diameter $D_f = D - 2h_f$, fillet radius $r = 0.38m$ and tooth thickness $S = m\pi/2$ are selected as the middle-layer parameters to describe the local surface and line features.

The control points on the boundary are treated as the low-level parameters. Among them, the control points on the gear tooth profile are the most complex, shown in Fig. 18(b). The gear tooth profile is approximated by circular arcs and then is expressed by NURBS [62]. P_1, P_2 and P_3 denote respectively the intersection points of the tooth profile with the tip circle, the reference circle and the root circle. D_a, D and D_b can be brought into the involute equation to obtain these points respectively. Then these three points are used to find the approximating circle. The point p_t is the intersection of the tangents at points p_1 and p_2. The dimensional parameters are mapped to the control points at the tooth profile.

After the outer profile of the gear is obtained, the gear unit can be segmented into quadrilaterals. Finally, they are arrayed according to the number of teeth, mapped to swept volumes and merged to a whole spur gear shown in Fig. 18.

7.1.2. Non-standard mechanical parts

The non-standard mechanical parts are usually irregular, as shown in Fig. 19. In first column, we show the size parameters of the input as an example of a major feature surface. In the second column, we give some schematic diagrams of the main parts of the geometric feature network.

The impeller model in the last row is taken as an example to demonstrate the general modeling process. In the first picture, all the parameters in semantic feature network are shown.

The contact feature surfaces of the fan blade and hub are regular circular surfaces, so they can be projected onto the plane easily. The projection of both hub and fan blade are rectangular. In the first picture, $Fh = \{l_1, l_2, l_3, \alpha, P_t\}$, l_1 and l_2 determine the shape of the hub, l_3 and α affect the shape of the fan blade, P_t defines the center point of the fan blade. According to the F_h, we can obtain $Fm = \{l_4, l_5, l_6, P_c\}$. $l_5 = 2P_t.y/\sin\alpha$, $l_6 = l_1 - P_t.x - l_3/2 - P_t.y/\tan\alpha$. And the control points on the surface are set as low-level parameters $Fl = \{P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9\}$, $P_7 = (l_1 - l_6 - l_3 - l_5 cos\alpha, l_5 sin\alpha)$, the semantic feature network is obtained and mapping from input parameters to control points is created. In the second column, the geometric feature network $Geo\_F = \{S_1, S_2, S_3, S_4, S_5, L_1, L_2\}$. S_5 and S_4 can be obtained by affine transformation of S_3, S_1 can be obtained by affine transformation of S_2. We need to segment the projection of contact section surfaces and map segmentation results to surfaces. Then the parametric volumes are obtained by sweeping and revolving. Finally, all the parts are merged and the whole impeller is obtained in the third picture.

7.1.3. Free-form models

Free-form models always have twisted shapes or irregular structures. In these models, the paths can be input by interactive operations or automatic extraction from existing models. Since the paths of these models are very complex, the segmentation of the feature nodes is not easy. After applying the segmentation, mapping by the conversion method, along with the merging operation, the whole model turns up. As shown in Fig. 20, the models are of eight-model, dog and lattice structure respectively.

7.2. IGA for complex volume parametric models

With the help of our modeling method, IGA can directly be applied on the complex three-dimensional shapes, which provides a chance to extend our method into many fields. We realize our IGA method by improving the method presented by Hughes [38] without details of the algorithm for its wide coverage. A bearing block and a reducer are selected to be analyzed with their analysis results listed in Fig. 21.

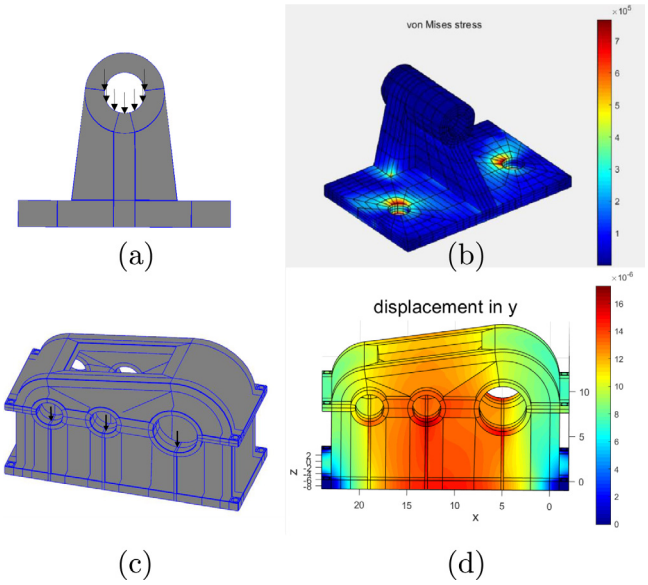

Fig. 21. Isogeometric analysis of mechanical parts.

In the figure, the bearing block is listed in (a), the loads on the half of the bottom top hole and the constraints on the two holes at the bottom are applied on the model. The reducer is listed in (c), the loads on the six holes on two sides and the constraints on the four holes at the bottom. The analysis results presented in the form of colorful shape are shown in (b) and (d). The deviation of analysis result can be checked by comparing with the results from other commercial CAE software.

8. Conclusions

This paper presents a volumetric parameterization method for mechanical parts and free-form models. Different from most of the current methods, our method creates a model rather than converts it from a B-rep model. The essence of the parametric design and the feature design is also adopted into our method since their concepts, ideas, methods and technologies are widely employed by the current CAD and CAE software. Besides, the proposed segmentation–mapping–merging mechanism will help generate and simulate the complex three-dimensional models in geometry and physics, which is the bottleneck problem in the study of volume parameterization and IGA. Benefit from above all, the volumetric parameterization and IGA will be accepted and integrated into the current mainstream commercial design software. What is more, our method makes it possible to realize the real-time parametric simulation of complex models, which will be our feature work.

However, imposing more constraints on the volume parametric model during its modeling process is under study, such as the distribution of singularities and the quality control of the operations. Adding heterogeneous material onto complex volume parametric solid is also worth to be studied in the future.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is supported by the National Nature Science Foundation of China under Grant No. 52075340, the Zhejiang Provincial Science and Technology Program in China under Grant 2021C01108, and the NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization under Grant No. U1909210.

References

[1] Oxman R. Thinking difference: Theories and models of parametric design thinking. Des Stud 2017;52(06):4–39.
[2] Monizza GP, Bendetti C, Matt DT. Parametric and generative design techniques in mass-production environments as effective enablers of industry 4.0 approaches in the building industry. Autom Constr 2018;92(02):270–85.
[3] Harding JE, Shepherd P. Meta-parametric design. Des Stud 2017;52(09):73–95.
[4] Jabi W, Soe S, Theobald P, et al. Enhancing parametric design through non-manifold topology. Des Stud 2017;52(04):96–114.
[5] Xu G, Mourrian B, Duvineau R, et al. Parameterization of computation domain in isogeometric analysis: methods and comparison. Comput Methods Appl Mesh Engrg 2011;200(23–24):2021–31.
[6] Lai Y, Zhang YJ, Liu L, et al. Integrating CAD with abaqus: A practical isogeometric analysis software platform for industrial applications. Comput Math Appl 2017;74(7):1648–60.
[7] Lai Y, Lei L, Zhang YJ, et al. Rhino 3D to abaqus: a T-spline based isogeometric analysis software framework[M]. Springer International Publishing; 2016.
[8] Hsu M, Wang C, Herrema AJ, et al. An interactive geometry modeling and parametric design platform for isogeometric analysis. Comput Math Appl 2015;70(7):1481–500.
[9] Yu Y, Wei X, Li A, et al. HexGen and Hex2Spline: Polycube-based hexahedral mesh generation and spline modeling for isogeometric analysis applications in LS-DYNA[J]. In: Springer INdAM series: Proceedings of INdAM workshop geometric challenges in isogeometric analysis. 2020.
[10] Yu Y, Liu JG, Zhang YJ. HexDom: Polycube-based hexahedral dominant mesh generation[J]. In: The edited volume of mesh generation and adaptation: cutting-edge techniques. SEMA-SIMAI Springer Series; 2021.
[11] Gujarathi GP, Ma YS. Parametric CAD/CAE integration using a common data model. J Manuf Syst 2011;30(3):118–32.
[12] Wassermann B, Bog T, Kollmannsberger S, et al. A design-through-analysis approach using the finite cell method[C]. In: Proceedings of the 7th European congress on computational methods in applied sciences and engineering. 2016, p. 2601–13.
[13] Schillinger D, Dede L, Scott MA, et al. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. Comput Methods Appl Mech Engrg 2012;249–252(12):116–50.

[14] Cohen E, Martin T, Kirby RM, et al. Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis. Comput Methods Appl Mech Engrg 2010;199(5–8):334–56.
[15] Zhu L. Li M, Martin RR. Direct simulation for CAD models undergoing parametric modifications. Comput Aided Des 2016;78(9):3–13.
[16] Rank E, Ruess M, Kollmannsberger S, et al. Geometric modeling, isogeometric analysis and the finite cell method. Comput Methods Appl Mech Engrg 2012;249(2):104–15.
[17] Schillinger D, Ruess M. The finite cell method: A review in the context of higher-order structural analysis of CAD and image-based geometric models. Arch Comput Methods Eng 2015;22(3):391–455.
[18] Wassermann B, Kollmannsberger S, Bog T, et al. From geometric design to numerical analysis: A direct approach using the finite cell method on constructive solid geometry. Comput Math Appl 2017;74(7):1703–26.
[19] Zhang YJ. Geometric modeling and mesh generation from scanned images[m]. Int J Radiat Biol Relat Stud Phys Chem Med 2016.
[20] Zhang Y, Bajaj C, Sohn BS. 3D finite element meshing from imaging data. Comput Methods Appl Mech Engrg 2005;194(48–49):5083–106.
[21] Qian J, Zhang Y. Automatic unstructured all-hexahedral mesh generation from B-reps for non-manifold CAD assemblies. Eng Comput 2012;28(4):345–59.
[22] Zhang Y, Hughes T, Bajaj CL. An automatic 3D mesh generation method for domains with multiple materials.. Comput Methods Appl Mech Eng 2010;199(5–8):405–15.
[23] Jin Q, Zhang Y, Wang W, et al. Quality improvement of non-manifold hexahedral meshes for critical feature determination of microstructure materials. Internat J Numer Methods Engrg 2010;82(11):1406–23.
[24] Zhang Y, Bajaj C. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. Comput Methods Appl Mech Engrg 2006;195(9–12):942–60.
[25] Herrema AJ, Wiese NM, Darling CN, et al. A framework for parametric design optimization using isogeometric analysis. Comput Methods Appl Mech Engrg 2017;316(4):944–65.
[26] Hughes T, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput Methods Appl Mech Engrg 2005;194(39–41):4135–95.
[27] Zhang Y, Bazilevs Y, Goswami S, et al. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. Comput Methods Appl Mech Eng 2007;196(29–30):2943–59.
[28] Martin T, Cohen E, Kirby M. Volumetric parameterization and trivariate b-spline fitting using harmonic functions. Comput Aided Geom Design 2009;26(6):648–64.
[29] Zhang Y, Wang W, Hughes TJ. Conformal solid T-spline construction from boundary T-spline representations. Comput Mech 2013;51(6):1051–9.
[30] Chan CL, Anitescu C, Rabczuk T. Volumetric parametrization from a level set boundary representation with PHT-splines. Comput Aided Des 2017;82(1):29–41.
[31] Akhras H, Elguedj T, Gravouil A, et al. Isogeometric analysis-suitable trivariate NURBS models from standard B-rep models. Comput Methods Appl Mech Engrg 2016;307(8):256–74.
[32] Zuo BQ, Huang ZD, Wang YW, et al. Isogeometric analysis for CSG models. Comput Methods Appl Mech Engrg 2015;285(3):102–24.
[33] Xu G, Mourrain B, Duvigneau R, et al. Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis. Comput Aided Des 2013;45(4):812–21.
[34] Xu G, Mourrain B, Duvigneau R, et al. Constructing analysis-suitable parameterization of computational domain from CAD boundary by variational harmonic method. J Comput Phys 2013;252(6):275–89.
[35] Wang X, Qian X. An optimization approach for constructing trivariate b -spline solids. Comput Aided Des 2014;46(8):179–91.
[36] Nguyen VP, Kerfriden P, Bordas S, et al. Isogeometric analysis suitable trivariate NURBS representation of composite panels with a new offset algorithm. Comput Aided Des 2014;55(5):49–63.
[37] Wang W, Zhang Y, Liu L, et al. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. Comput Aided Des 2013;45(2):351–60.
[38] Cottrell JA, Hughes T, Bazilevs Y. Isogeometric analysis: toward integration of CAD and FEA[M]. Wiley Publishing; 2009.
[39] Buchegger F, Jüttler B. Planar multi-patch domain parameterization via patch adjacency graphs. Comput Aided Des 2017;82(5):2–12.
[40] Xiao S, Kang H, Fu XM, et al. Computing IGA-suitable planar parameterizations by PolySquare-enhanced domain partition. Comput Aided Geom Design 2018;62(3):29–43.
[41] Liu L, Zhang Y, Liu Y, et al. Feature-preserving T-mesh construction using skeleton-based polycubes. Comput Aided Des 2015;58(1):162–72.
[42] Hu K, Zhang YJ, Liao T. Efficient volumetric PolyCube-map construction. Comput Graph Forum 2016;35(7):97–106.
[43] Kangkang Hu, Zhang Yongjie, Jessica Liao Tao. Surface segmentation for polycube construction based on generalized centroidal voronoi tessellation. Comput Methods Appl Mech Engrg 2017;316(4):280–96.
[44] Chen L, Xu G, Wang S, et al. Constructing volumetric parameterization based on directed graph simplification of polycube structure from complex shapes. Comput Methods Appl Mech Engrg 2019;351(7):422–40.
[45] Perduta A, Putanowicz R. Tools and techniques for building models for isogeometric analysis. Adv Eng Softw 2019;127(10):70–81.
[46] Benzaken J, Herrema AJ, Hsu MC, et al. A rapid and efficient isogeometric design space exploration framework with application to structural mechanics. Comput Methods Appl Mech Engrg 2017;316(12):1215–56.
[47] Lei L, Zhang Y, Hughes T, et al. Volumetric T-spline construction using boolean operations. Eng Comput 2014;30(4):425–39.
[48] Livesu M, Muntoni A, Puppo E, et al. Skeleton-driven adaptive hexahedral meshing of tubular shapes[C]. John Wiley & Sons, Ltd; 2016, p. 237–46.
[49] Huang J, Jiang T, Shi Z, Tong Y, Bao H, Desbrun M. l1-based construction of polycube maps from complex shapes. ACM Trans Graph 2014;33(3 June):25:1–25:11.
[50] Andersen KD. An efficient newton barrier method for minimizing a sum of euclidean norms. SIAM J Optim 1996;6(1):74–95.
[51] El-attar R, Vidyasagar M, Dutta S. An algorithm for l1-norm minimization with application to nonlinear l1 approximation. SIAM J Numer Anal 1979;16(1):70–86.
[52] Nowottny D. Quadrilateral mesh generation via geometrically optimized domain decomposition[C]. In: Proceedings of 6th international meshing roundtable. 1997, p. 309–20.
[53] Bommes D, Zimmer H, Kobbelt L. Mixed-integer quadrangulation. ACM Trans Graph 2009;28(3):1–10.
[54] Peng CH, Barton M, Jiang C, Wonka P. Exploring quadrangulations. ACM Trans Graph 2014;33(1):1–13.
[55] Liu H, Zhu C, Li C. Constructing N-sided toric surface patches from boundary curves. J Inf Comput Sci March 2012;9(3):737–43.
[56] Guan H, Qiang W, Ping X. A fast algorithm for intersection calculation of ray and NURBS surface in predicting radar cross section calculation. J Eng Graph 2006;27(1):87–91.
[57] Chan CL, Anitescu C, Rabczuk T. Strong multipatch C1-coupling for isogeometric analysis on 2D and 3D domains. Comput Methods Appl Mech Engrg 2019;357(12):112599.
[58] Masalha R, Cirillo E, Elber G. Heterogeneous parametric trivariate fillets. Comput Aided Geom Design 2021;86:101970.
[59] Massarwi F, Elber G. A B-spline based framework for volumetric object modeling. Comput Aided Des 2016;78:36–47.
[60] Biswas A, Shapiro V, Tsukanov I. Heterogeneous material modeling with distance fields. Comput Aided Geom Design 2004;21(3):215–42.
[61] Massarwi P, Elber G. Volumetric untrimming: Precise decomposition of trimmed trivariates into tensor products. Comput Aided Geom Design 71:1–15.
[62] Wang Z, Ma J, Kang X, et al. Involute gear contact pattern analysis and simulation based on NURBS surface. J Mech Transm 2014;38(1):44–9.