



Generation of multi-axis swept mesh in a global way

Haiyan Wu¹ · Shuming Gao¹ · Rui Wang^{1,2} · Mao Ding¹

Received: 8 June 2018 / Accepted: 15 October 2018 / Published online: 30 October 2018
© Springer-Verlag London Ltd., part of Springer Nature 2018

Abstract

Multi-axis sweeping is an important tool to generate hexahedral meshes for solid models which are composed of swept volumes with different sweep directions. However, traditional multi-axis sweeping algorithms either fail to handle complex grafting relationships between swept volumes or are easy to produce hexahedral elements with bad quality around the graft surfaces. To achieve a high-quality multi-axis swept mesh, this paper proposes a global approach to multi-axis swept mesh generation, which can robustly generate hexahedral meshes for solid models composed by swept volumes with different sweep directions. We first generate all surface meshes globally by applying an optimized structured quadrilateral mesh generation algorithm. After that, we generate a swept mesh for each swept volume. Finally, we determine an appropriate way to optimize the topology of the generated mesh so as to improve the mesh quality. The experimental results show the effectiveness and efficiency of the proposed method.

Keywords Hexahedral mesh · Mesh generation · Swept mesh · Structured quadrilateral mesh

1 Introduction

1.1 Background

Finite element analysis (FEA) is a numerical method used to solve problems in the domains of structure analysis, dynamics, and solid mechanics. The 3D volume mesh is an important input for FEA. Two common categories of 3D volume meshes are tetrahedral meshes and hexahedral meshes. Compared with tetrahedral meshes, hexahedral meshes have the advantages of smaller number of elements, higher computational precision and faster convergence when applied to FEA.

There are many mature methods for high-quality tetrahedral mesh generation. However, the automated and high-quality hexahedral meshing of complex solid models is still very challenging. At present, the main hexahedral

mesh generation methods are submapping [1], sweeping [2–6], whisker weaving [7], plastering [8, 9], grid-based [10], H-morph [11], dual cycle elimination based [12, 13], frame field based [14], sheet operation based [15], etc. A comprehensive review of all these methods can be found in the survey by Sarrate et al. [16]. Among all the hexahedral mesh generation methods, sweeping is the most widely used, accounting for more than 50% of meshing applications [17].

One of the most popular ways to generate hexahedral meshes for complex solid models is to first decompose them into swept volumes, and then use sweeping algorithms to generate hexahedral meshes for each swept volume. In this way, the quality of the resulting hexahedral mesh is generally very high, however, it is difficult to achieve mesh conformity at the common surfaces between swept volumes with different sweep directions. To solve this problem, some multi-axis sweeping algorithms [18–20] have been proposed, but these algorithms are either easy to produce mesh elements with bad quality or unable to handle complex grafting relationships between swept volumes. Therefore, this paper aims to propose a multi-axis swept mesh generation method, which can generate high-quality hexahedral meshes for solid models that are composed of swept volumes with complex grafting relationships.

✉ Shuming Gao
smgao@cad.zju.edu.cn

Haiyan Wu
wuhaiyan@zju.edu.cn

¹ State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, People's Republic of China

² Faculty of Mechanical Engineering and Mechanics, Ningbo University, Ningbo 315211, People's Republic of China

1.2 Related work

1.2.1 Sweeping methods

Sweeping methods can generate high-quality hexahedral meshes for swept volumes. The procedure used in most sweeping methods first classifies the surfaces of the input swept volume as source surfaces, target surfaces and linking surfaces, where the source and target surfaces are called cap surfaces. Then, the source surface is meshed with a quadrilateral mesh and the linking surfaces are meshed with structured quadrilateral meshes. Next, the source surface mesh is projected onto the target surface to keep the mesh topology the same between the two cap surfaces. Finally, the swept mesh is generated in a layer-by-layer fashion along the sweep direction. According to the number of source surfaces and target surfaces, swept volumes can be classified into one-to-one, many-to-one, and many-to-many swept volumes, all of which are suitable for generating single-axis swept meshes, and many mature methods [2–6] have been proposed by now.

1.2.2 Multi-axis sweeping methods

To expand the application scope of sweeping methods, Miyoshi et al. [18] proposed the multi-axis cooper algorithm, which can generate a hexahedral mesh through multi-axis sweeps. However, the types of applicable solid models of this approach are limited. The biggest challenge in multi-axis swept mesh generation is how to guarantee the mesh conformity at the common surfaces between swept volumes with different sweep directions. To this end, Jankovich et al. [19] proposed the grafting algorithm, and Earp [20] further improved this algorithm. This method determines the order of the mesh generation by establishing a grafting relationship among the swept volumes with different sweep directions. A branch-swept volume is grafted on the linking surface of a trunk-swept volume, and the common surface between them is referred to as the **graft surface**, as shown in Fig. 1. The swept mesh of trunk-swept volume is generated first. Then this method locally modifies the position and connectivity of the nodes on the linking surfaces to align with the graft surfaces. Once the surface mesh is formed on the graft surface, it is swept along the branch to create a swept mesh. The grafting algorithm greatly expands the range of models that can be meshed by sweeping methods. However, it still suffers from the following drawbacks: (1) this method cannot deal with complex grafting relationships between swept volumes, such as when a branch is grafted onto two trunks at the same time, or when the grafting relationship

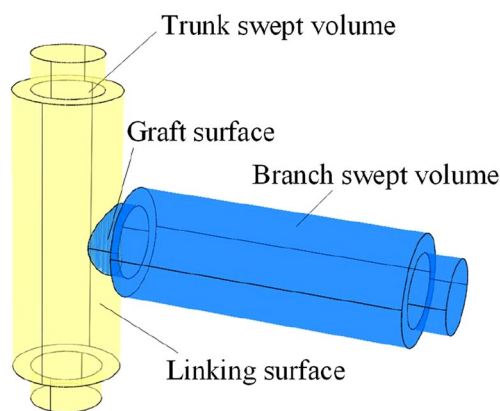


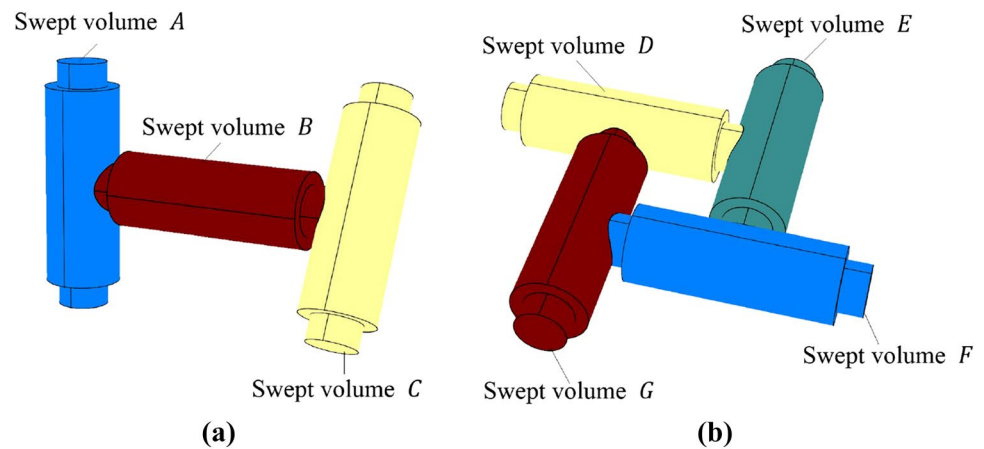
Fig. 1 Definition of graft surface

forms a loop. (2) The quality of the mesh near the graft surface is often not satisfactory although some measures have been taken to locally improve the mesh quality. Figure 2a shows an example in which a branch-swept volume is grafted onto two trunk-swept volumes at the same time. The swept volume *B* is grafted simultaneously on the linking surfaces of the swept volumes *A* and *C*. The grafting algorithm is not able to deal with this case as it cannot guarantee the mesh conformity at both of the graft surfaces. Figure 2b shows an example where the grafting relationship forms a loop. The swept volume *D* is grafted onto the linking surface of the swept volume *E*, and this in turn is grafted onto the linking surface of the swept volume *F*, the swept volume *F* is grafted onto the linking surface of the swept volume *G*, and this is grafted back to the linking surface of the swept volume *D*. For this kind of grafting relationship, this algorithm cannot find a reasonable swept mesh generation order, hence making this model unsuitable for being meshed by the grafting algorithm.

1.2.3 Structured quadrilateral mesh generation methods

To generate a swept mesh, structured quadrilateral meshes must be generated on the linking surfaces. The direction of this structured mesh should be consistent with the sweep direction. In multi-axis swept mesh generation, to ensure the mesh conformity between the trunk-swept volume and the branch-swept volume, the structured quadrilateral mesh on the linking surface of the trunk-swept volume should also be aligned with the bounding loops of the graft surfaces. In other words, the mesh on the graft surface should be structured and its direction should be consistent with the sweep direction of trunk-swept volume. The construction of structured quadrilateral meshes has been extensively studied and extended. Ruiz-Gironés et al. [21] and Cai et al. [22] improved the submapping algorithm [1] to mesh complex surfaces with structured quadrilateral meshes. However, in

Fig. 2 Two examples with complex grafting relationships which cannot be handled by the existing grafting algorithm



these algorithms, the direction of the structured quadrilateral mesh is automatically determined and cannot be specified by users. In the grafting algorithm [19], the structured quadrilateral meshes inside the graft surfaces are obtained by locally modifying the location of the mesh nodes on the linking surfaces. This often leads to the problem that the elements near the bounding loops of the graft surfaces are usually of poor quality.

1.3 Our approach

In this paper, we propose a robust multi-axis swept mesh generation approach which can greatly expand the application scope of multi-axis sweeping. Specifically, our approach has the following characteristics:

- To effectively deal with the complex grafting relationships among swept volumes, we globally generate the surface meshes and swept meshes for all the swept volumes. This eliminates the dependency on the mesh generation order of the swept volumes.
- To achieve better mesh quality on graft surfaces, we put forward an algorithm which can generate optimized structured quadrilateral meshes for graft surfaces.
- To eliminate the bad mesh elements which are generated as a compromise to achieving mesh conformity on graft surfaces, we adopt topological operations to effectively improve the quality of the hexahedral mesh generated by multi-axis sweeping.

2 Approach overview

In order to meet the requirement of high-quality hexahedral mesh generation, this paper presents a novel multi-axis swept mesh generation approach. The current approaches to the generation of multi-axis swept meshes suffer from two main problems: first, it is challenging to mesh models

with complex grafting relationships as shown in Fig. 2a, b; second, it is difficult to match the quality of the generated hexahedral mesh with the application requirements. To solve the first problem, our approach works in two steps. First, it globally generates the surface meshes of all the swept volumes while ensuring the mesh conformity on the common surfaces. Then it generates the swept mesh of each swept volume. To address the second problem, we propose an algorithm which can generate optimized structured quadrilateral meshes on graft surfaces. Subsequently, we further improve the quality of the generated hexahedral mesh by modifying the mesh topology using a set of local topological operations.

The input to this approach is the result of swept volume decomposition obtained by the method of Wu et al. [23, 24]. All of the many-to-one and many-to-many swept volumes are first decomposed into one-to-one swept volumes. The output is the hexahedral mesh whose quality can meet application requirements. As shown in Fig. 3, this approach mainly includes the following three steps:

Step 1 *Boundary surface meshing* Discretize all the curves of the input swept volumes and globally generate the quadrilateral meshes of all surfaces.

Step 2 *Swept mesh generation* Using the sweeping algorithm, generate and merge the swept meshes of all swept volumes into a global mesh.

Step 3 *Topological optimization* Determine a set of local topological modification operations to improve the mesh quality.

In our approach, to deal with complex grafting relationships among input swept volumes, all of the many-to-many and many-to-one swept volumes are first decomposed into one-to-one swept volumes. The main reasons are as follows: on the one hand, in order to achieve many-to-many sweeping, certain constraints on the source and target surface mesh topologies must be satisfied. On the other hand, in order to achieve multi-axis sweeping with mesh conformity guaranteed, certain constraints must be satisfied on the

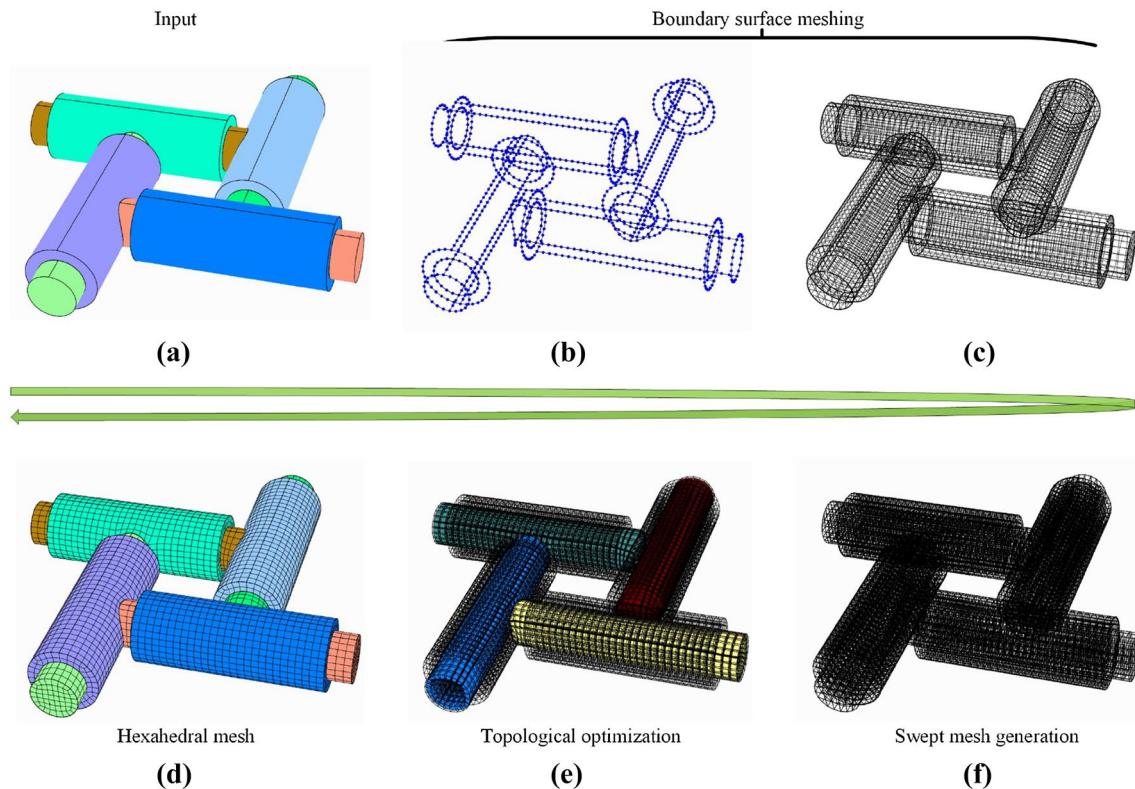


Fig. 3 An illustration of the proposed multi-axis sweeping algorithm. **a** Input solid model which has been decomposed into one-to-one swept volumes. **b** Curve discretization. **c** Generated surface meshes.

d Generated swept mesh. **e** Dual sheets are inserted to improve the topological quality of the generated mesh. **f** Final hexahedral mesh generated

interfaces between swept volumes. These constraints must spread all over the solid model by decomposing the input swept volumes into one-to-one swept volumes before mesh generation. The geometric decomposition may encounter some difficulties when dealing with extreme cases, which is currently a limitation of our approach.

As the swept mesh generation process of Step 2 can be directly accomplished by applying sweeping methods, we will mainly focus on Step 1 and Step 3 in the following sections.

3 Boundary surface meshing

To ensure that a swept mesh can be generated by the sweeping algorithm, the surface mesh must ensure that the linking surface mesh is structured, and the topologies of the target and source surface meshes are the same. In multi-axis sweeping, the surface mesh should also ensure that the meshes on the common surfaces between the different the swept volumes are identical. As shown in Fig. 4, since surfaces: f_1 and f_2 are the common surfaces of two swept volumes, the surface meshes on them must be exactly the same. In a similar way, the surface meshes on f_3 and f_4 should also

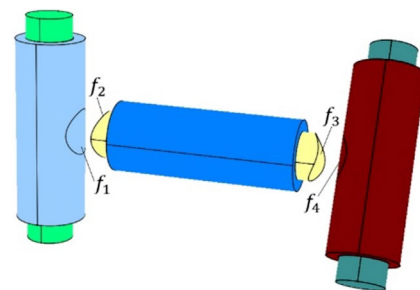


Fig. 4 A group of surfaces whose meshes cannot be generated separately

be exactly the same. In addition, since f_2 and f_3 are the two cap surfaces of a swept volume, their surface meshes must have the same topology. So, all four surfaces f_1 , f_2 , f_3 , and f_4 must have the same mesh topology. This makes it difficult to generate surface meshes separately for each swept volume. In order to effectively generate the surface meshes that both satisfy the conformity requirements of the common surfaces and meet the conditions that a swept mesh can be generated for each swept volume, this paper proposes to globally generate the surface meshes for all swept volumes. We first discretize all the geometric curves globally, then group the

surfaces that have constraints with each other. Finally, we globally generate the surface meshes, and use an optimized structured mesh generation algorithm for the graft surfaces.

3.1 Curve discretization

To effectively support the global surface mesh generation, we first discretize all geometric curves of the swept volumes. The curve discretization must satisfy the following three conditions: first, to ensure the conformity of the mesh at the common surfaces between different swept volumes, the discretization of the common curves should be the same. Second, in order to ensure the generation of structured quadrilateral meshes on the linking surfaces of each swept volume, the number of intervals on the opposing curves of a linking surface should be equal (i.e., the number of intervals in the I^+ direction in the submapping algorithm [1] should be equal to the number of intervals in the I^- direction, and the number of intervals in the J^+ direction should be equal to that in the J^- direction). Third, to generate a quadrilateral mesh on each cap surface, it is necessary to ensure that the sum of the number of intervals of all geometric curves on each cap surface is even. (For a surface, the necessary condition for generating a quadrilateral mesh is that the number of intervals is even.) In order to make the curve discretization satisfy the above three constraints, and at the same time meet the user specified mesh size d , we propose to minimize the following objective function to globally calculate the number of intervals n_c for each geometric curve.

Minimize:

$$\sum_c \left| \frac{n_c}{l_c} - \frac{1}{d} \right|,$$

subject to the following constraints:

$$n_{c_i} = n_{c_j} \text{ for all common curve pair } c_i \text{ and } c_j,$$

$$\sum_{c \in I^+} n_c = \sum_{c \in I^-} n_c \text{ for all linking surface } f_l,$$

$$\sum_{c \in J^+} n_c = \sum_{c \in J^-} n_c \text{ for all linking surface } f_l,$$

$$\sum_{c \in f_s} n_c = 2n_{f_s} \text{ for all cap surface } f_s,$$

$$n_c, n_{f_s} \in \mathbb{N}_+,$$

where l_c is the length of each curve, and $2n_{f_s}$ is the sum of the intervals of cap surface f_s . To solve the above problem using linear programming, we introduce the auxiliary variable β_c and change the objective function to the minimization of:

$$\sum_c \beta_c$$

and add the following constraints:

$$\beta_c \geq \frac{n_c}{l_c} - \frac{1}{d},$$

$$\beta_c \geq -\left(\frac{n_c}{l_c} - \frac{1}{d} \right).$$

In our implementation, we use the Ipsolve library [25] to solve the above linear problem. For the input solid model in Fig. 3a, the number of intervals calculated for each curve is shown in Fig. 5. We uniformly discretize each curve accordingly, and the result is shown in Fig. 3b.

3.2 Surface grouping

As mentioned earlier in this section, the meshes of some surfaces need to be either exactly or topologically identical. These surfaces have constraints on each other, and their meshes cannot be generated separately, so we group these surfaces, and mesh the surfaces in the same group all at once. First, to meet the conformity requirement, the common surfaces between the swept volumes are grouped. As shown in Fig. 6a, eight groups are formed for the solid model in Fig. 4. Secondly, to meet the requirement for topological identicalness of the source and target surfaces, the cap surfaces of each swept volume are grouped. As shown in Fig. 6b, another six groups are formed. Finally, we join the groups with common elements. Hence *Group-7*, *Group-8* and *Group-14* are joined together to form the surface group shown in Fig. 6c, and finally twelve surface groups are formed in total. The surfaces in *Group-1* to *Group-6* are all linking surfaces, and they are all suitable to be meshed by existing structured quadrilateral mesh generation algorithms [21, 22]. The surfaces in *Group-9* to *Group-13* are all cap surfaces. For each of these groups, we apply the paving algorithm [26] to one of the surfaces to generate a quadrilateral mesh, and then project the mesh onto the other surfaces of that group. The surface group in Fig. 6c is composed of four surfaces, and they are all graft surfaces. These surfaces should be meshed with structured quadrilateral meshes, however, existing quadrilateral meshing algorithms cannot be applied to mesh these surfaces. For the meshing of graft

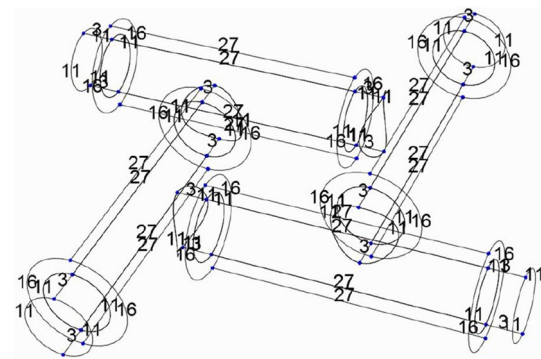


Fig. 5 The number of intervals calculated for each curve

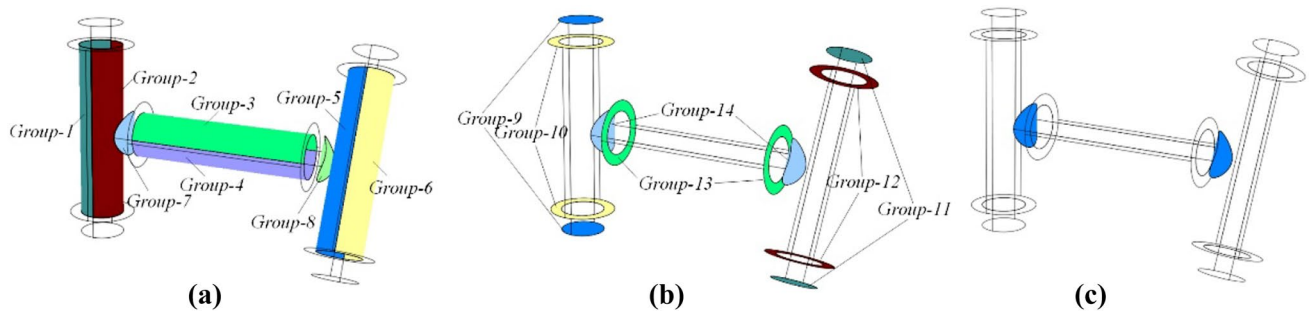


Fig. 6 Surface grouping. **a** Common surfaces are grouped. **b** Cap surfaces are grouped. **c** A group which contains graft surfaces

surfaces, we present an optimized structured quadrilateral mesh generation method in Sect. 3.3.

It should be noted that there are also some surfaces that do not belong to any group, and they can be easily meshed, hence we do not discuss about them here.

3.3 Optimized structured quadrilateral meshing of graft surfaces

The graft surfaces in Fig. 6c are generally not suitable for structured quadrilateral meshing. However, due to the structural requirement of the linking mesh, it is necessary to generate structured quadrilateral meshes that conform to the sweep direction and the given curve discretization. Therefore, this paper presents an optimized structured quadrilateral mesh generation method. The input to the algorithm is the surface whose curve discretization is given, as shown in Fig. 7a, and the direction of the structured mesh, as shown in Fig. 7b. The output is a structured quadrilateral mesh that conforms to the given curve discretization and has a direction that is consistent with the given direction.

One of the most critical problems in constructing a structured quadrilateral mesh is the parameterization of the boundary curves. As shown in Fig. 7c, each curve segment is parameterized as I^+ , J^+ , I^- , or J^- . For the problem of this paper, as long as the parameters of each curve segment can be determined, the topology of the resulting structured

quadrilateral mesh can be directly determined, as shown in Fig. 7d. Thus, the problem of structured quadrilateral mesh generation can be transformed into the problem of solving for the parameters of each curve segment. To make the direction of the generated mesh consistent with the given direction, our goal is to minimize the difference between the actual direction of each curve segment and its ideal direction in the computational domain. To achieve this, we propose to minimize the following objective function:

$$\sum_{k=1}^n \left(p_{k_{j^+}} \times x_{k_{j^+}} + p_{k_{j^-}} \times x_{k_{j^-}} + p_{k_{i^+}} \times x_{k_{i^+}} + p_{k_{i^-}} \times x_{k_{i^-}} \right),$$

$$p_{k_{j^+}} = \sin \frac{\theta_{v_{j^+}, v_k}}{2},$$

$$p_{k_{j^-}} = \sin \frac{\theta_{v_{j^-}, v_k}}{2},$$

$$p_{k_{i^+}} = \sin \frac{\theta_{v_{i^+}, v_k}}{2},$$

$$p_{k_{i^-}} = \sin \frac{\theta_{v_{i^-}, v_k}}{2},$$

constrained to

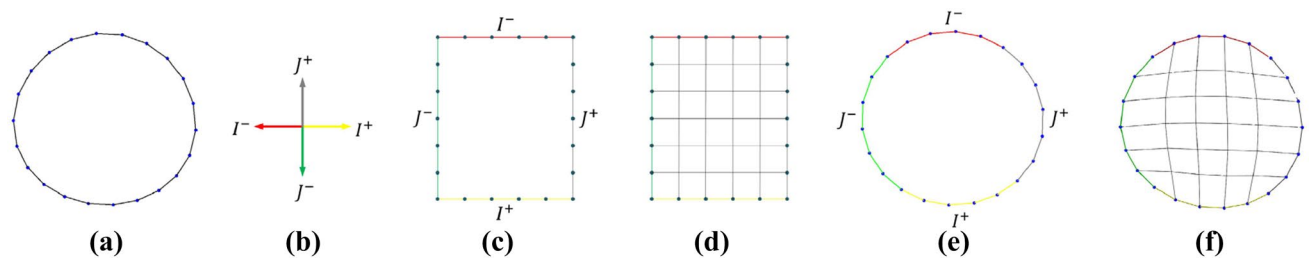


Fig. 7 Optimized structured quadrilateral meshing. **a** Curve discretization of input surface. **b** Specified direction of final structured quadrilateral mesh. **c** Curve parameterization in the computational

domain. **d** Interior node interpolation in the computational domain. **e** Curve parameterization of input surface. **f** Generated structured quadrilateral mesh

$$x_{k_{j^+}} + x_{k_{j^+}} + x_{k_{j^-}} + x_{k_{j^-}} = 1,$$

$$x_{k_{j^+}}, x_{k_{j^+}}, x_{k_{j^-}}, x_{k_{j^-}} \in \{0, 1\}.$$

Among them, $x_{k_{j^+}}, x_{k_{j^+}}, x_{k_{j^-}}$ and $x_{k_{j^-}}$ denote whether the k_{th} segment belongs to the parameters I^+, J^+, I^- or J^- . When $x_{k_{j^+}} = 1$, the k_{th} segment belongs to I^+ ; when $x_{k_{j^+}} = 1$, the k_{th} segment belongs to J^+ , and so on. $p_{k_{j^+}}, p_{k_{j^+}}, p_{k_{j^-}}$ and $p_{k_{j^-}}$ refer to the penalty coefficient of the k_{th} segment belonging to I^+, J^+, I^- and J^- , respectively. v_k refers to the direction of the k_{th} segment, $\theta_{v_{j^+}, v_k}, \theta_{v_{j^+}, v_k}, \theta_{v_{j^+}, v_k}$ and $\theta_{v_{j^+}, v_k}$ denote the angles between v_k and $v_{j^+}, v_{j^+}, v_{j^-}$ and v_{j^-} , respectively.

In order to ensure the formation of structured quadrilateral meshes, we must ensure that the number of segments belonging to I^+ and I^- are equal, and the number of segments belonging to J^+ and J^- are also equal. So we add the following constraints:

$$\sum_{k=1}^n x_{k_{j^+}} = \sum_{k=1}^n x_{k_{j^-}},$$

$$\sum_{k=1}^n x_{k_{j^+}} = \sum_{k=1}^n x_{k_{j^-}}.$$

Besides, the parameters of two successive segments cannot be I^+ and I^- , respectively, or J^+ and J^- , respectively. Therefore, we add the following constraints:

$$|t_k - t_{k+1}| \neq 2, \tag{1}$$

$$t_k = 1 \times x_{k_{j^+}} + 2 \times x_{k_{j^+}} + 3 \times x_{k_{j^-}} + 4 \times x_{k_{j^-}},$$

where t_k denotes the type of the k_{th} segment. When the value of t_k is 1, 2, 3 and 4, the type of the k_{th} segment is I^+, J^+, I^- and J^- , respectively. In order to remove the absolute value in Eq. (1), we rewrite Eq. (1) as

$$-1 \leq a_k \leq 1,$$

$$a_k = t_k - t_{k+1} + 4 \times m_k \quad m_k \in \mathbb{Z},$$

where a_k also refers to the type of the common node between the k_{th} segment and the $(k + 1)_{th}$ segment. When the value of a_k is $-1, 0$ and 1 , the type of the common node is CORNER, SIDE and END respectively. To ensure the generation of a structured quadrilateral mesh, the sum of the types of nodes must be equal to 4 [1, 21], so we add the following constraint:

$$\sum_{k=1}^n a_k = 4.$$

At last, we further add the following constraint:

$$|a_k - \bar{a}_k| \leq 1, \tag{2}$$

where the \bar{a}_k refers to the ideal type of the common node between the k_{th} segment and the $(k + 1)_{th}$ segment according to the angle between them. For example, when the angle between the segments is $\pi/2$, the ideal type of their common node should be END, and the value of \bar{a}_k is 1. This constraint is used to avoid that the type of some node deviate too much from its ideal type, so as to guarantee the mesh quality.

We use the Ipsolve library [25] to solve the above linear problem. Figure 7e shows the result of the parameterization of this surface. The segments with the parameter of I^+, J^+, I^- and J^- are colored yellow, gray, red and green, respectively. According to the result of the parameterization, it is easy to generate a structured quadrilateral mesh as shown in Fig. 7f. As the surface itself is not suitable for structured quadrilateral meshing, its mesh quality is not particularly desirable. In the next section, we will continue to optimize it.

After meshing all the surfaces, the surface mesh shown in Fig. 3c is obtained for the solid model in Fig. 3a.

4 Topological optimization of graft surface mesh

After the surface meshes are generated, we apply the sweep algorithm to generate a swept mesh for each swept volume. Figure 3d shows the generated swept mesh, and the quality measurement of the generated mesh is shown in Fig. 8a. As mentioned in the previous section, to ensure the mesh conformity between swept volumes, we have to generate structured quadrilateral meshes for graft surfaces which are generally not suitable for structured quadrilateral meshing. Therefore, some mesh elements with poor quality are generated in the hexahedral mesh. As shown in Fig. 8b, the hexahedral elements with the scaled Jacobian values below 0.3 are distributed either in the vicinity of the bounding loops of the graft surfaces, or in the swept mesh produced by the graft surface mesh. To further improve the quality of the generated mesh, we optimize the topology of graft surface mesh by inserting a set of dual chords [27] first, being described in this section, and then optimize the topology of the hexahedral mesh by inserting a set of dual sheets [27], which will be presented in the next section.

To optimize the graft surface mesh by dual chords insertion, the most important thing is to determine the insertion paths which are the input of dual chords insertion. An insertion path is composed of a set of continuous mesh edges on the surfaces. Figure 9c shows an example of an insertion path, and by inflating each mesh edge of the path, a dual chord can be generated, as shown in Fig. 9d. The insertion of this new chord can effectively improve the surface mesh quality.

To determine the insertion paths that can be used to generate the dual chords which can effectively improve the mesh

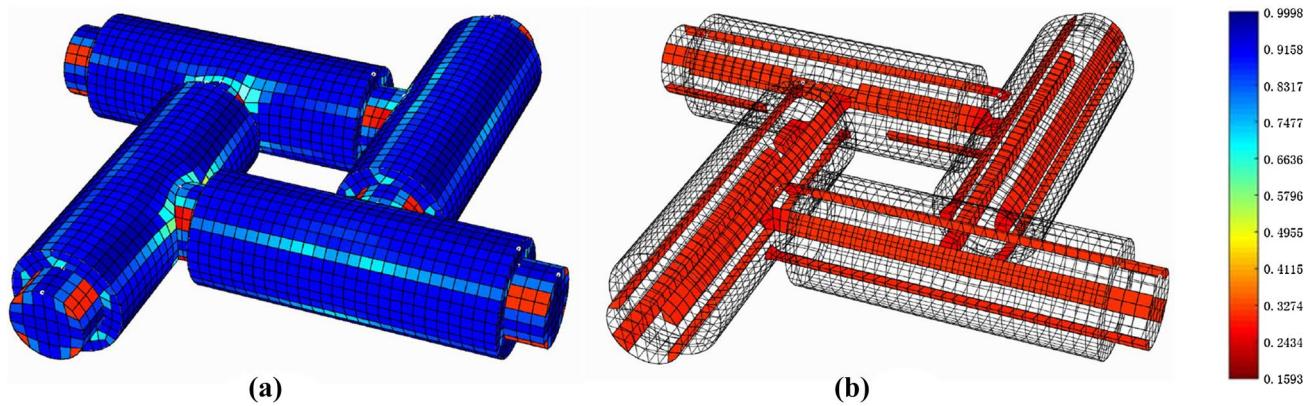


Fig. 8 Quality measurement of the generated swept mesh before topological optimization. **a** Visualization of the scaled Jacobian value distribution. The red-blue color coding is used with red indicating

smaller Jacobian values. **b** The hexahedral elements whose scaled Jacobian value is below 0.3

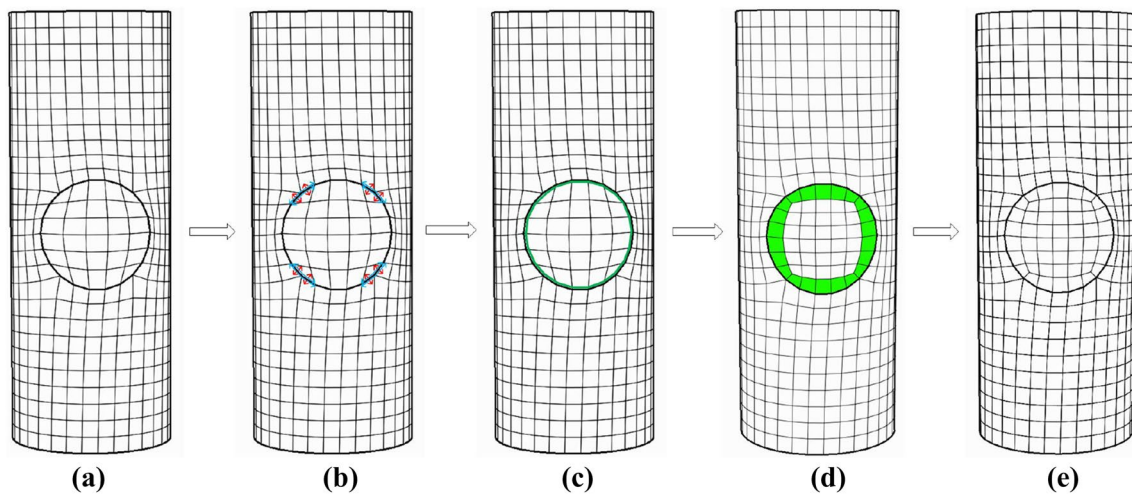


Fig. 9 Flowchart of graft surface mesh optimization. **a** The mesh on the linking surface which contains a graft surface. **b** The local topological modification templates applied to improve the mesh quality.

c The insertion path formed by connecting the local modification templates. **d** The chord generated based on the insertion path. **e** The improved graft surface mesh

quality on the linking surfaces which contains graft surfaces, we first determine a set of local topological modification templates, then globally connect these local templates to form insertion paths.

4.1 Local topological modification templates

As the quadrilateral meshes on the linking surfaces are structured, the number of quadrilaterals adjacent to each internal node of the mesh is four, which is ideal for quadrilateral meshes. Therefore, the mesh quality problems are mainly concentrated on the bounding loops of the graft surface, and the local topological modification templates are only defined for the nodes which lies on the bounding loops of the graft surface. In our previous work [28], we have enumerated

all the possible local topological modification templates as shown in Table 1. For each boundary node, the ideal number of quadrilaterals adjacent to it can be determined based on the angle between its two adjacent mesh edges, as shown in the first, fourth and seventh columns of Table 1. Besides, all the possible undesirable configurations on each boundary node can also be enumerated, since the mesh is structured and the number of possible quadrilaterals on each mesh node is limited, as shown in the second, fifth and eighth columns of Table 1. Finally, in order to convert each undesirable configuration to the ideal configuration, the local topological modification templates are defined as shown in the third, sixth and ninth columns of Table 1, where the red dotted lines refer to the dual edges locally inserted to optimize the local configuration.

Table 1 Local topological modification templates of graft surface meshes

Ideal configuration	Undesirable configuration	Modification template	Ideal configuration	Undesirable configuration	Modification template	Ideal configuration	Undesirable configuration	Modification template

Primal edges (boundary)
 Primal edges (inner)
 Dual edges
 Locally inserted dual edges

Although Table 1 have listed all the possible modification templates, some of the templates cannot be directly achieved through chord insertion. For example, the first template in the ninth column cannot be generated by determining insertion paths to be inflated as there are too few edges around the node in the original configuration. Besides, as the original node type is CORNER which is so different from its ideal type REVERSAL, it is difficult to guarantee the mesh quality after the topological modification, even if these templates are applicable. Fortunately, these extreme templates are not necessarily used in this work, as we have avoided these extreme cases during the structured quadrilateral mesh generation process of the graft surface by adding the constraints in Eq. (2) to the optimization function. The difference between the value of original type and ideal type of each node is at most 1, so only a subset of the templates in Table 1 are needed, and Fig. 10 shows all the templates used in this paper.

In Fig. 10, by inflating the blue edge segments with arrows along the direction denoted by the red arrows, the new quadrilaterals colored by blue are generated, and the local mesh quality around each node is improved. We define these blue edges as **ports**, and each port is composed of two mesh edges. As shown in Fig. 9b, by applying these

templates to mesh on the linking surface which contains a graft surface, eight ports are determined.

4.2 Insertion paths formation

The determined ports only show the way to improve the mesh quality locally. To optimize the surface mesh, these ports should be properly connected to form insertion paths so as to conduct dual chords insertion. The formed insertion paths should meet four requirements: (1) to effectively improve the mesh quality, they should pass through all the important ports. (2) To be a valid input to dual chords insertion, they should either be closed loops or start and end at the surface boundary. (3) To avoid diminishing the mesh quality, they should not introduce unnecessary singularities to the mesh. (4) To avoid the excessive impact on mesh density after chord insertion, they should be as short as possible. To form the insertion paths which satisfy these requirements, we first determine the connection between every pair of ports which can be connected through mesh edges with no turns on the mesh, then extend each port to the surface boundary to generate virtual ports and build the connections with virtual ports, finally we select a subset from all the port connections to form insertion paths by

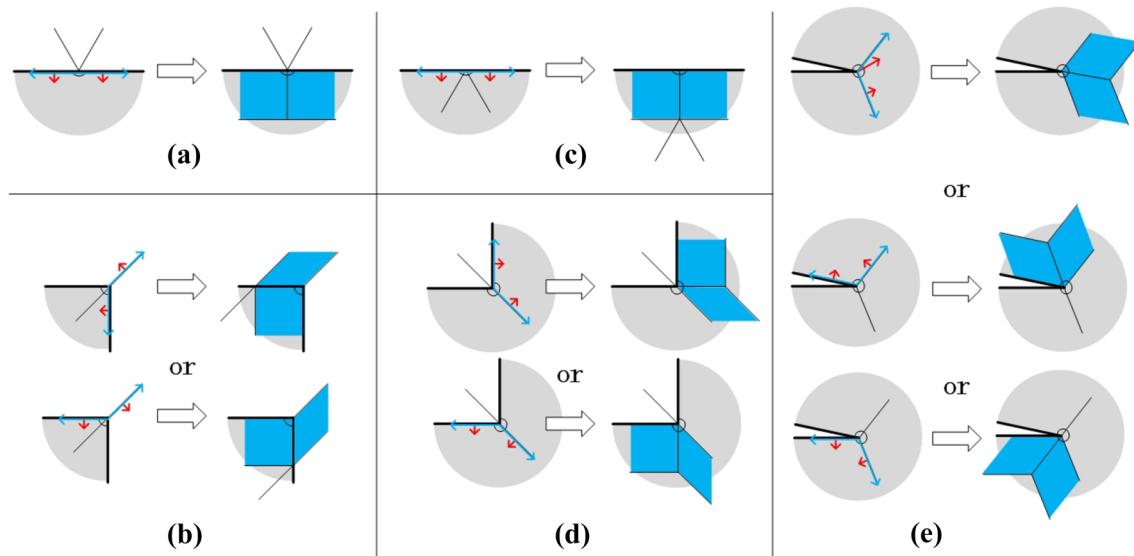


Fig. 10 Refined templates of local topological modification. The thick black lines refer to the geometric curves on the graft surface loop. The gray shading represents the area within the graft surface

loop. **a** END->SIDE template. **b** SIDE->END templates. **c** CORNER->SIDE template. **d** SIDE->CORNER templates. **e** CORNER->REVERSAL templates

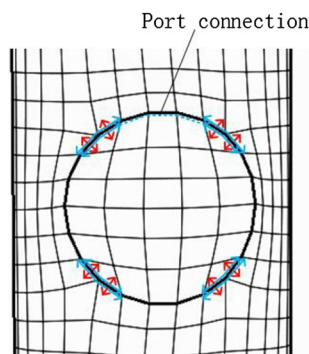


Fig. 11 The port connection between two ports

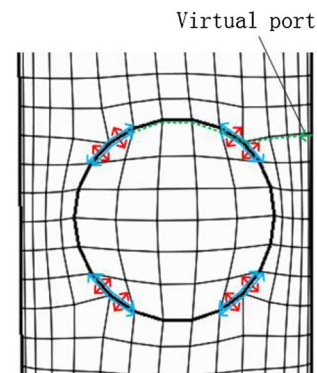


Fig. 12 The virtual port generated by extending a port and the port connection between the original port and the virtual port

using an optimization function. The main steps of the optimized insertion paths formation algorithm are as follows:

- (1) *Determination of port connections* A port connection C_{ij} is formed between every two ports which can be connected through mesh edges with no turns on the structured quadrilateral mesh, as shown in Fig. 11. This is because unnecessary singularities which are bad for the mesh quality will be introduced by the turns on the insertion paths.
- (2) *Generation of virtual ports* By extending each port to the surface boundary, virtual ports are formed on the boundary, and new port connections are formed with these virtual ports, as shown in Fig. 12. This is because a valid insertion path should either be a closed loop or start and end at the surface boundary.
- (3) *Selection of port connections* To select a subset of all the port connections to form the optimized insertion paths, we propose to minimize the following objective function:

$$\min \sum l_{ij} x_{ij},$$

where $x_{ij} \in \{0, 1\}$ refers to whether port connection C_{ij} is used in the final insertion paths; when $x_{ij} = 1$, C_{ij} is a part of the final insertion paths. l_{ij} refers to the length (number of mesh edges) of C_{ij} . The above optimization function can effectively minimize the length of the insertion paths. Besides, to ensure the validity of the insertion paths, the following constraint is added:

$$\sum x_{pl} = \sum x_{pr}, \tag{3}$$

where $\sum x_{pl}$ refers to the number of selected port connections which connect port p with its left mesh edge, and $\sum x_{pr}$ refers to the number of selected port connections which connect port p with its right mesh edge. Equation (3) is used to make sure that insertion paths are either closed or end at the surface boundary. Finally, we add the following constraint to the optimization function:

$$\sum x_{pl} \geq 1 \text{ for each important port } p. \tag{4}$$

During ports determination, we have determined all the ports that can be used to improve the mesh quality. However, if all these ports are used in the final insertion paths, too many new mesh elements will be generated. Generally, the more ports are used, the better the mesh quality, at the cost of mesh density. In our approach, we only make sure that all the important ports are used in the final insertion paths by Eq. (4). A port is considered important if the original mesh quality around the node is greater than a threshold value ϵ . The mesh quality q of a node is $|\theta/n - \pi/2|$, where θ refers to the corner angle at the node, and n refers to the number of quadrilateral elements at the node. The smaller the value of q , the better the mesh quality. The value of ϵ can be specified by users according to their requirements of mesh quality and mesh density.

Figure 9c shows the final optimized insertion path formed to connect the ports in Fig. 9b with $\epsilon = \pi/3$.

5 Topological optimization of hexahedral mesh

In the previous section, we have discussed how to improve the surface mesh quality through dual chords insertion. However, it should be pointed out that dual chords insertion

operation cannot be applied directly to the surface mesh of a hexahedral mesh, otherwise the surface mesh will be incompatible with the inner hexahedral mesh. To simultaneously optimize the surface mesh and the hexahedral mesh, we insert a set of dual sheets whose boundaries are the dual chords we need.

As sheet inflation [29] is a general operator used for inserting hexahedral sheets, we apply sheet inflation operations to insert the sheets which can improve the mesh quality. Sheet inflation takes a continuous quadrilateral set as input and generates a new sheet by inflating the quadrilateral set. Figure 13a shows a continuous quadrilateral set inside the hexahedral mesh, and Fig. 13c shows the sheet generated by inflating this quadrilateral set. In our approach, to optimize the topological structure of the generated hexahedral mesh by sheet inflation, the main problem is to determine the quadrilateral sets. The determined quadrilateral sets should meet five requirements: (1) to effectively improve the surface mesh quality, the quadrilateral sets should pass through the insertion paths we have determined in Sect. 2. To effectively improve the hexahedral mesh quality, the quadrilateral sets should be able to improve the local topological quality around the region where the mesh quality is not satisfactory. (3) To be a valid input to sheet inflation, the quadrilateral sets should be inflatable. An inflatable quadrilateral set is a set of connected quadrilaterals that terminates at the boundary or closes upon itself into a ball. (4) To avoid diminishing the mesh quality, the quadrilateral sets should introduce as few singularities as possible to the hexahedral mesh by sheet inflation. (5) To avoid the excessive impact on mesh density after sheet inflation, the quadrilateral sets should be as small as possible. To form the quadrilateral sets which satisfy the above requirements, we propose an algorithm to optimized quadrilateral sets formation. Regarding the first requirement, we form a quadrilateral set for each insertion path by extending them into the trunk mesh and branch mesh, respectively. Regarding the second requirement, the quadrilateral sets are

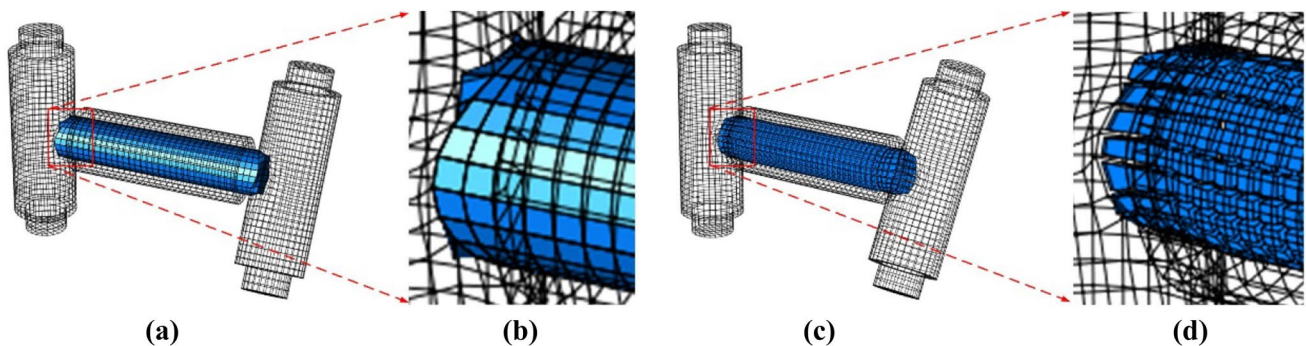


Fig. 13 Topological optimization of the hexahedral mesh by sheet inflation (the inner edges of the hexahedral mesh are not rendered). **a** A quadrilateral set inside the hexahedral mesh. **b** A zoom-in view

near the graft location of **a**. **c** A sheet generated by sheet inflation. **d** A zoom-in view near the graft location of **c**

properly extended inside the branch mesh which is generated by sweeping the graft surface mesh with quality issues. Regarding the third requirement, for each insertion path, the corresponding quadrilateral set inside the trunk mesh is formed by first determining a shrink set which is a continuous set of hexahedra inside the mesh, and the outer boundary of the shrink set is an inflatable quadrilateral set. Regarding the fourth requirement, our approach makes full use of the structure of the generated swept mesh to form the quadrilateral sets with good quality. Regarding the fifth requirement, a shortest path search algorithm is used during the quadrilateral sets formation to reduce the number of quadrilaterals contained by each quadrilateral set.

Figure 14 shows the main process of our algorithm to form an optimized quadrilateral set for an insertion path. The main steps are as follows:

- (1) *Linking surface mesh subdivision* Figure 14a shows a multi-axis swept mesh which is composed of a trunk mesh and a branch mesh, and an insertion path is formed on the linking surface mesh of the trunk-swept volume. The insertion path subdivides the linking surface mesh into two parts, as shown in Fig. 14b.
- (2) *Quadrilateral patches subdivision* We choose the part with smaller number of quadrilaterals in Fig. 14b and further subdivide it into patches vertically, as shown in Fig. 14c. The smaller part is chosen to reduce the size of the final quadrilateral set. Each quadrilateral patch can be represented by a set of starting edges and
- (3) *Starting quadrilaterals determination* For each quadrilateral patch, we search for a shortest path within the layer mesh of its starting edges to form a closed loop, and the quadrilaterals inside the loop are obtained. As shown in Fig. 14d, the determined starting quadrilaterals are shown in the same color with the corresponding quadrilateral patch. The shortest path search algorithm used in this step is similar to the algorithm in [30], so we omit the details.
- (4) *Sub-shrink set determination* For each quadrilateral patch, a set of hexahedra which is called sub-shrink set can be obtained according to the determined starting quadrilaterals and the number of layers, as shown in Fig. 14e.
- (5) *Shrink set determination* We join all the sub-shrink sets to form a shrink set, as shown in Fig. 14f.
- (6) *Trunk quadrilateral set formation* The inner boundary of the shrink set forms the quadrilateral set inside the trunk mesh, as shown in Fig. 14g.
- (7) *Branch quadrilateral set formation* Finally, we extend the trunk quadrilateral set inside the branch mesh along

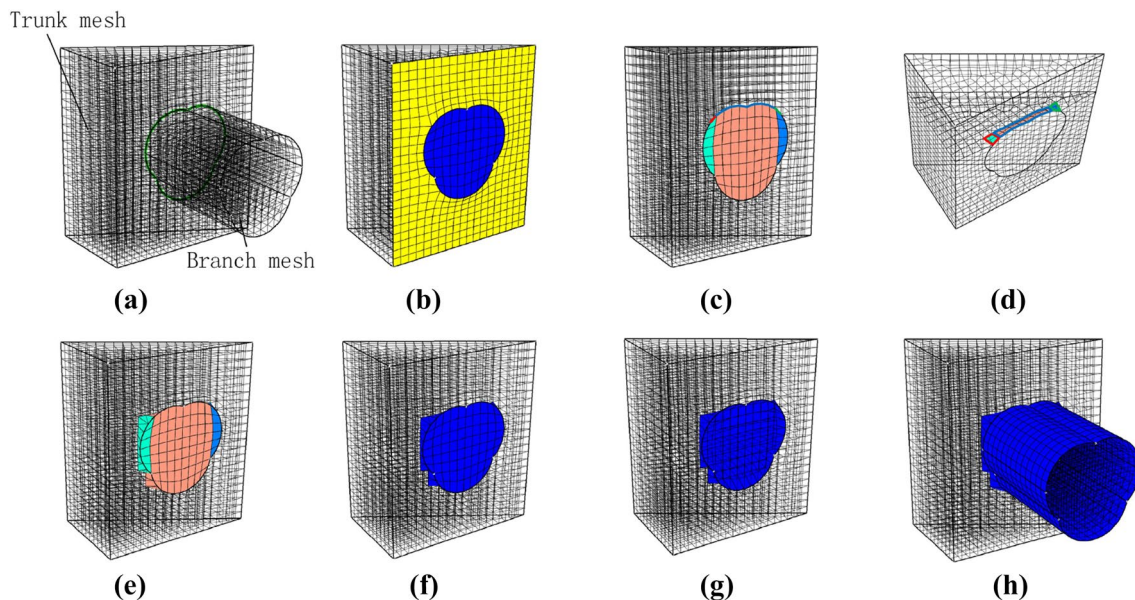


Fig. 14 Main process of optimized quadrilateral set formation for one insertion path. **a** A multi-axis swept mesh with an insertion path on the linking surface mesh of the trunk-swept volume. **b** Linking surface mesh subdivision. **c** Quadrilateral patches subdivision of the blue

part in **b**. **d** Starting quadrilaterals determination (the inner edges of the hexahedral mesh are not rendered except for the particular layer). **e** Sub-shrink set determination. **f** Shrink set determination. **g** Trunk quadrilateral set formation. **h** Branch quadrilateral set formation

the sweep direction of the branch-swept volume to form the entire quadrilateral set, as shown in Fig. 14h. It is very easy to extend the quadrilateral set inside the branch mesh, since the branch mesh is generated by sweeping the graft surface mesh, and the mesh on each layer has the same topology as the graft mesh.

Figure 15 shows the quadrilateral sets formed to optimize the hexahedral mesh in Fig. 3d. Figure 3e shows the dual sheets generated based on these quadrilateral sets. After the topology of the hexahedral mesh is changed, we use the mesh smoothing algorithm [31, 32] to further optimize the hexahedral mesh geometrically.

6 Results and discussion

The algorithm presented in this paper was implemented using C++ as the programming language and ACIS [33] as the geometry engine. A collection of solid models was used to test our automatic multi-axis sweeping approach. Several representative examples are presented in this section.

The example in Fig. 3a is a model whose grafting relationship forms a loop. Existing multi-axis sweeping algorithms cannot generate hexahedral meshes for this kind of models. Our global strategy can successfully generate the swept mesh for this model, as shown in Fig. 16a. After inserting the four sheets as shown in Fig. 3e, the minimum scaled Jacobian value of the final hexahedral mesh is 0.5508.

The example in Fig. 17a shows a case where a swept volume is grafted simultaneously onto two swept volumes. Existing multi-axis sweeping algorithms cannot generate hexahedral meshes for it. The final hexahedral mesh generated for this example is shown in Fig. 17e. The minimum scaled Jacobian value of the final mesh is 0.4203.

The solid models in Figs. 18, 19 and 20 are used to test our ability to deal with complex graft surfaces. It is generally very difficult to generate structured quadrilateral meshes for these graft surfaces. Besides, it is hard to achieve high mesh quality around the graft areas. Our approach successfully generates the optimized structured quadrilateral meshes for

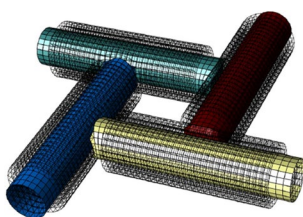


Fig. 15 The quadrilateral sets formed for sheet inflation to optimize the hexahedral mesh in Fig. 3d

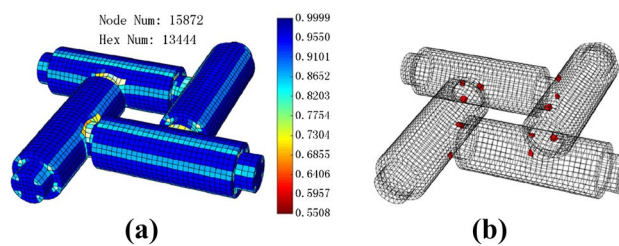


Fig. 16 Example-1. a Visualization of the scaled Jacobian value distribution of the final hexahedral mesh generated for the input solid model in Fig. 3a. b The hexahedral elements whose scaled Jacobian value is below 0.6

these graft surfaces, and the generated hexahedral meshes are of high quality after the insertion of some dual sheets. Compared with the grafting algorithm [19], our algorithm generally can generate hexahedral meshes with higher quality at similar mesh density.

The solid model in Fig. 21 is used to test our ability to deal with concave graft surfaces. The graft surface is composed of one outer loop with concave vertices and an

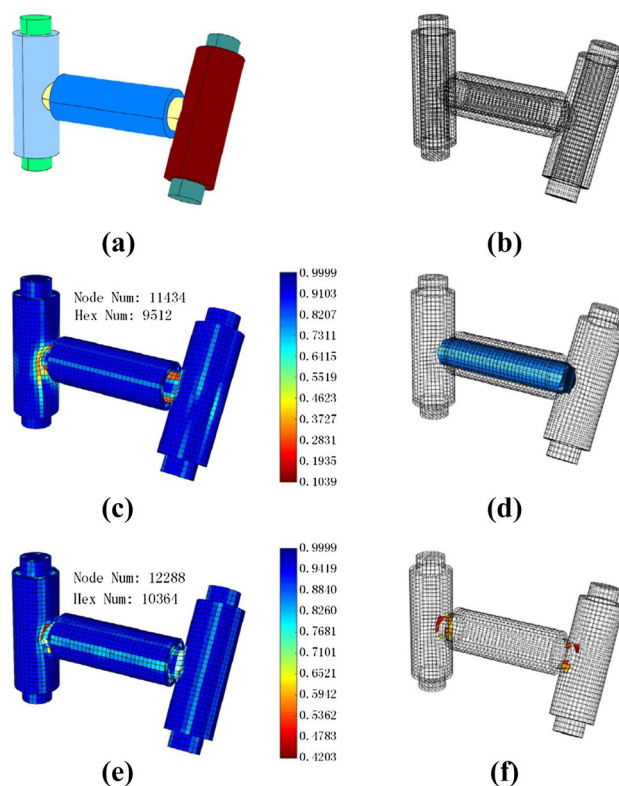
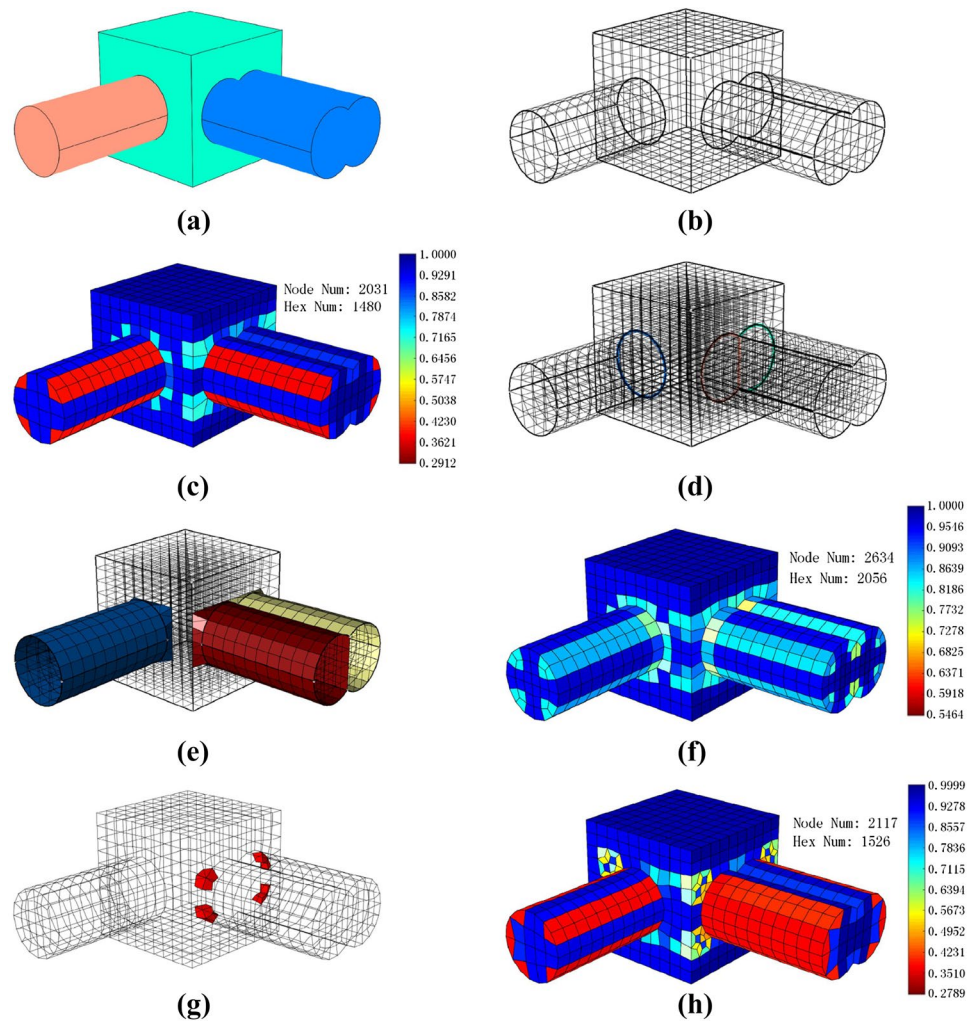


Fig. 17 Example-2. a The input solid model which is composed of six swept volumes. b The generated surface meshes. c Visualization of the scaled Jacobian value distribution of the generated multi-axis swept mesh. d The quadrilateral set determined for sheet inflation to improve mesh quality. e Visualization of the scaled Jacobian value distribution of the final mesh. f The hexahedral elements whose scaled Jacobian value is below 0.6

Fig. 18 Example-3. **a** The input solid model which is composed of three swept volumes. **b** The generated surface meshes. **c** Visualization of the scaled Jacobian value distribution of the generated multi-axis swept mesh. **d** Three insertion paths formed to optimize the surface mesh quality. **e** Three quadrilateral sets formed to optimize the hexahedral mesh quality. **f** Visualization of the scaled Jacobian value distribution of the final mesh. **g** The hexahedral elements whose scaled Jacobian value is below 0.6 in the final mesh. **h** Visualization of the scaled Jacobian value distribution of the hexahedral mesh generated using the existing grafting algorithm [19]



inner loop. Our optimized structured quadrilateral meshing algorithm successfully generates a structured quadrilateral mesh for the graft surface, as shown in Fig. 21b, and the generated swept mesh is shown in Fig. 21c. After inserting two dual sheets to the generated mesh based on the quadrilateral sets shown in Fig. 21d, the minimum scaled Jacobian value of the final hexahedral mesh is 0.5257.

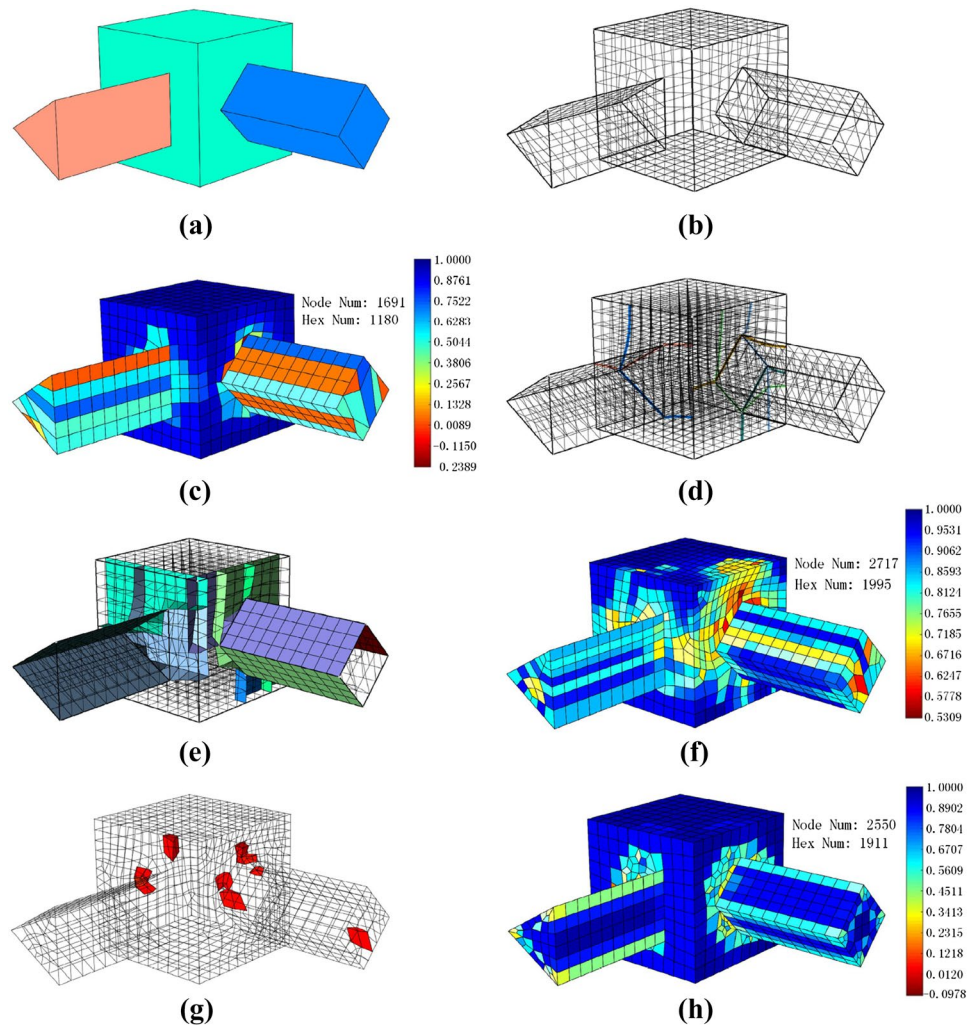
Figure 22a shows a complex solid model which is composed of 25 swept volumes. It is difficult to determine a valid mesh generation order for these swept volumes using previous multi-axis sweeping methods. Our global strategy successfully generated the surface mesh and hexahedral mesh for this model, as shown in Fig. 22b, c. After inserting eight dual sheets to the generated mesh based on the quadrilateral sets shown in Fig. 22d, the minimum scaled Jacobian value of the final hexahedral mesh is 0.4496.

7 Conclusions and future work

In this paper, a novel approach to multi-axis swept mesh generation is proposed to meet the requirement of high-quality hexahedral mesh generation. First, the surface meshes of all swept volumes are generated globally. Next, the swept meshes are generated for each swept volume. Finally, the quality of the generated mesh is improved by inserting a set of dual sheets. Compared with other existing methods, this approach has the following advantages:

- It can effectively deal with the complex grafting relationships among swept volumes. This is achieved by first globally generating the surface meshes of all swept volumes, and then generating the swept meshes.
- It can generate hexahedral meshes with higher quality for solid models which are composed of swept volumes

Fig. 19 Example-4. **a** The input solid model which is composed of three swept volumes. **b** The generated surface meshes. **c** Visualization of the scaled Jacobian value distribution of the generated multi-axis swept mesh. **d** Nine insertion paths formed to optimize the surface mesh quality. **e** Nine quadrilateral sets formed to optimize the hexahedral mesh quality. **f** Visualization of the scaled Jacobian value distribution of the final mesh. **g** The hexahedral elements whose scaled Jacobian value is below 0.6 in the final mesh. **h** Visualization of the scaled Jacobian value distribution of the hexahedral mesh generated using the existing grafting algorithm [19]



with different sweep directions. This is achieved by applying an algorithm which can generate optimized structured quadrilateral meshes for graft surfaces and locally inserting a set of dual sheets to optimize the topological structure of the generated hexahedral mesh.

The future work of this paper will mainly focus on the following three aspects: first, currently our method does not support the insertion of self-intersecting dual sheets. However, self-intersecting sheets are needed if the formed insertion paths are self-intersecting, although we have not encountered this situation yet. Therefore, the algorithm to

insertion paths formation should be improved to theoretically guarantee that the formed insertion paths are not self-intersecting. Secondly, we plan to parallelize the swept mesh generation process, this is because after the surface meshes are generated, the generation of swept mesh for each swept volume is independent of each other. Thirdly, in the final step of topological mesh optimization, we use sheet inflation operation to improve the quality of the mesh, which will locally increase the mesh density. Therefore, we plan to combine the sheet extraction operation to reduce the impact on the mesh size.

Fig. 20 Example-5. **a** The input solid model which is composed of three swept volumes. **b** The generated surface meshes. **c** Visualization of the scaled Jacobian value distribution of the generated multi-axis swept mesh. **d** Six insertion paths formed to optimize the surface mesh quality. **e** Six quadrilateral sets formed to optimize the hexahedral mesh quality. **f** Visualization of the scaled Jacobian value distribution of the final mesh. **g** The hexahedral elements whose scaled Jacobian value is below 0.6 in the final mesh. **h** Visualization of the scaled Jacobian value distribution of the hexahedral mesh generated using the existing grafting algorithm [19]

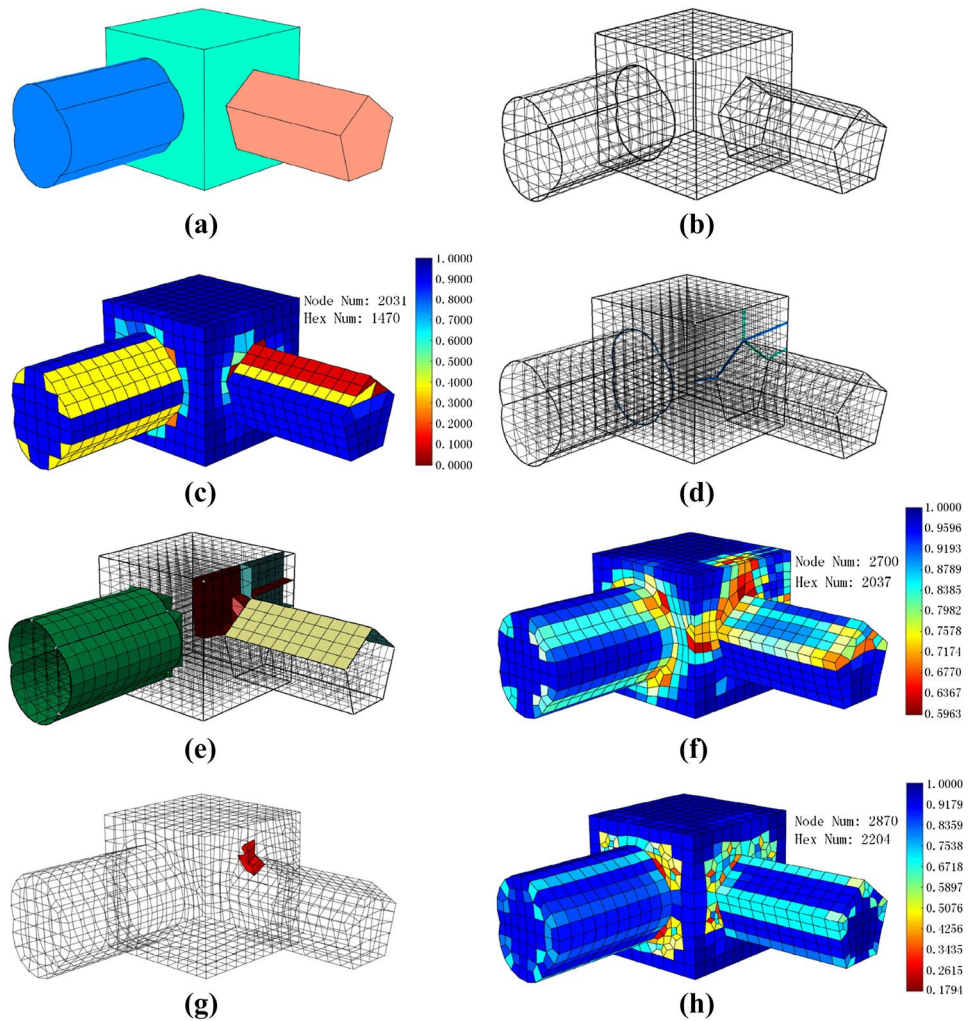


Fig. 21 Example-6. **a** The input solid model which is composed of two swept volumes. **b** The generated surface meshes. **c** Visualization of the scaled Jacobian value distribution of the generated multi-axis swept mesh. **d** Two quadrilateral sets determined for sheet inflation to improve mesh quality (the inner edges of the hexahedral mesh are not rendered). **e** Visualization of the scaled Jacobian value distribution of the final mesh. **f** The hexahedral elements whose scaled Jacobian value is below 0.6

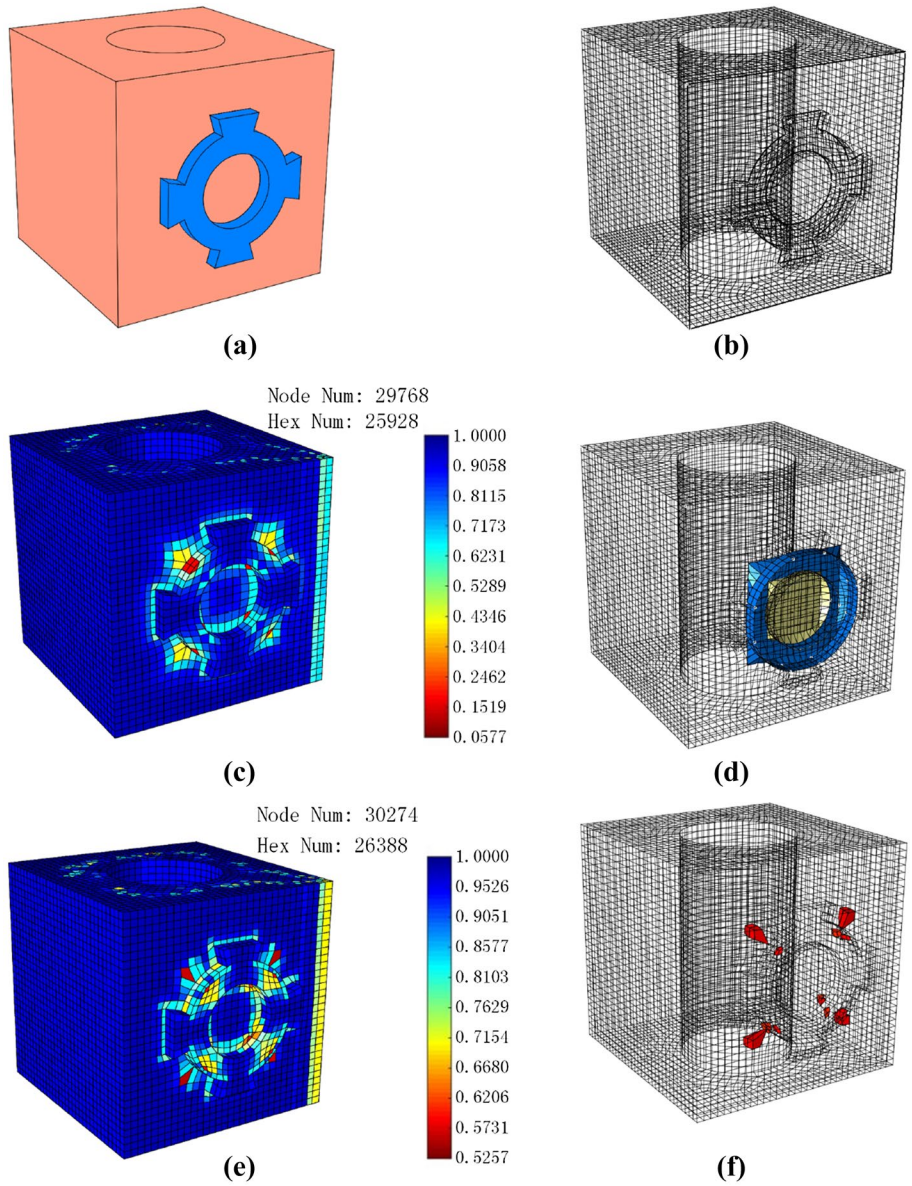
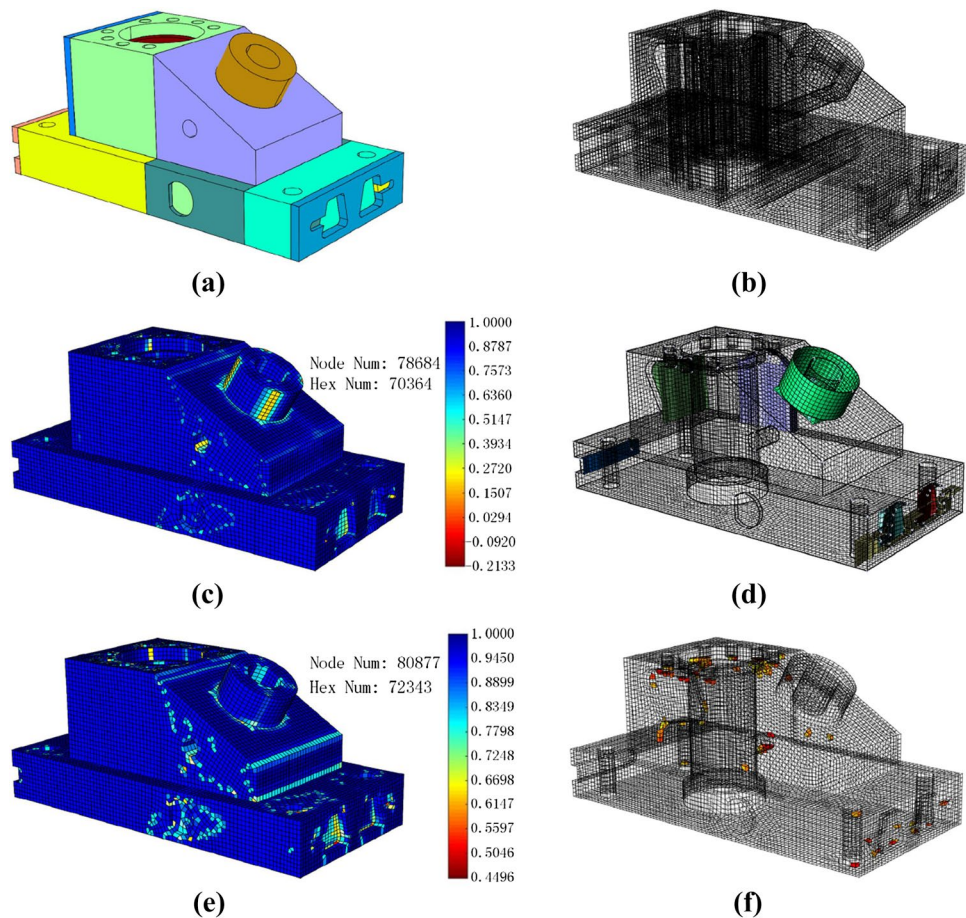


Fig. 22 Example-7. **a** The input solid model which is composed of 25 swept volumes. **b** The generated surface meshes. **c** Visualization of the scaled Jacobian value distribution of the generated multi-axis swept mesh. **d** Eight quadrilateral sets determined for sheet inflation to improve mesh quality (the inner edges of the hexahedral mesh are not rendered). **e** Visualization of the scaled Jacobian value distribution of the final mesh. **f** The hexahedral elements whose scaled Jacobian value is below 0.6



Acknowledgements The authors are very grateful to the financial supports from NSF of China (61572432, 61802211) and National 863 High Technology Plan (2013AA041301).

References

- Whiteley M, White D, Benzley S, Blacker T (1996) Two and three-quarter dimensional meshing facilitators. *Eng Comput* 12:144–154
- Knupp PM (1998) Next-generation sweep tool: a method for generating all-hex meshes on two-and-one-half dimensional geometries. In: *IMR*, pp 505–513
- Blacker T (1996) The cooper tool. In: *5th International Meshing Roundtable*, SAND 95-2130. Sandia National Laboratories, Citeseer
- White DR, Saigal S, Owen SJ (2004) Ccsweep: automatic decomposition of multi-sweep volumes. *Eng Comput* 20(3):222–236
- Scott MA, Benzley SE, Owen SJ (2005) Improved many-to-one sweeping. *Int J Numer Meth Eng* 65(3):332–348
- Lai M, Benzley S, White D (2000) Automated hexahedral mesh generation by generalized multiple source to multiple target sweeping. *Int J Numer Methods Eng* 49(12):261–275
- Tautges TJ, Blacker T, Mitchell SA (1996) The whisker weaving algorithm: a connectivity-based method for constructing all-hexahedral finite element meshes. *Int J Numer Methods Eng* 39(19):3327–3349
- Canann S (1992) Plastering—a new approach to automated, 3-d hexahedral mesh generation. In: *33rd Structures, structural dynamics and materials conference*, p. 2416
- Staten ML, Kerr RA, Owen SJ, Blacker TD, Stupazzini M, Shimada K (2010) Unconstrained plastering-hexahedral mesh generation via advancing-front geometry decomposition. *Int J Numer Methods Eng* 81(2):135–171
- Schneiders R, Bünten R (1995) Automatic generation of hexahedral finite element meshes. *Comput Aided Geom Des* 12:693–707
- Owen SJ, Saigal S (2000) Hmorph: an indirect approach to advancing front hex meshing. *Int J Numer Methods Eng* 49(12):289–312
- Müller-Hannemann M (1999) Hexahedral mesh generation by successive dual cycle elimination. *Engineering Comput* 15:269–279
- Kremer M, Bommers D, Lim I, Kobbelt L (2014) Advanced automatic hexahedral mesh generation from surface quad meshes. In: *Proceedings of the 22nd international meshing roundtable*, Springer, pp 147–164
- Huang J, Jiang T, Wang Y, Tong Y, Bao H (2012) Automatic frame field guided hexahedral mesh generation. In: *Tech. report*, State Key Lab of CAD&CG, College of Computer Science at Zhejiang University
- Wang R, Shen C, Chen J, Wu H, Gao S (2017) Sheet operation based block decomposition of solid models for hex meshing. *Comput Aided Des* 85:123–137

16. Sarrate J, Ruiz-Gironés E, Roca X (2014) Unstructured and semi-structured hexahedral mesh generation methods. *Comput Technol Rev* 10:35–64
17. White DR, Tautges TJ (1999) Automatic scheme selection for toolkit hex meshing. *Int J Numer Methods Eng* 49(12):127–144
18. Miyoshi K, Blacker T (2000) Hexahedral mesh generation using multi-axis cooper algorithm. In: IMR, pp 89–97
19. Jankovich SR, Benzley SE, Shepherd JF, Mitchell SA (1999) The graft tool: an all-hexahedral transition algorithm for creating a multidirectional swept volume mesh. In: IMR, pp. 387–392
20. Earp MN (2005) All hexahedral meshing of multiple source, multiple target, multiple axis geometries via automatic grafting and sweeping. In: M.Sc. Thesis, Brigham Young University
21. Ruiz-Gironés E, Sarrate J (2010) Generation of structured meshes in multiply connected surfaces using submapping. *Adv Eng Softw* 41(2):379–387
22. Cai S, Tautges TJ (2015) Optimizing corner assignment of sub-map surfaces. *Proc Eng* 124:83–95
23. Wu H, Gao S (2014) Automatic swept volume decomposition based on sweep directions extraction for hexahedral meshing. *Proc Eng* 82:136–148
24. Wu H, Gao S, Wang R, Chen J (2018) Fuzzy clustering based pseudo-swept volume decomposition for hexahedral meshing. *Comput Aided Des* 96:42–58
25. <http://sourceforge.net/projects/lpsolve>. Accessed 2017
26. Blacker TD, Stephenson MB (1991) Paving: a new approach to automated quadrilateral mesh generation. *Int J Numer Methods Eng* 32:811–847
27. Ledoux F, Shepherd J (2009) Topological and geometrical properties of hexahedral meshes. *Eng Comput* 26(4):419432
28. Wu H, Gao S, Wang R, Ding M (2017) A global approach to multi-axis swept mesh generation. *Proc Eng* 203:414–426
29. Staten ML, Shepherd JF, Ledoux F, Shimada K (2010) Hexahedral mesh matching: converting non-conforming hexahedral-to-hexahedral interfaces into conforming interfaces. *Int J Numer Methods Eng* 82:1475–1509
30. Borden M, Benzley S, Mitchell SA, White DR (2000) The cleave and fill tool: an all-hexahedral refinement algorithm for swept meshes. In: International Meshing Roundtable, pp 69–76
31. Gargallo-Peiró A, Roca X, Sarrate J (2014) A surface mesh smoothing and untangling method independent of the cad parameterization. *Comput Mech* 53:587–609
32. Ruiz-Gironés E, Roca X, Sarrate J, Montenegro R, Escobar JM (2015) Simultaneous untangling and smoothing of quadrilateral and hexahedral meshes using an object-oriented framework. *Adv Eng Softw* 80:12–24
33. 3D ACIS modeler, spatial corporation, <https://www.spatial.com/products/3d-acis-modeling>. Accessed 2017

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.