Engineering Analysis with Boundary Elements 152 (2023) 473–495

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

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

Automatic surface mesh generation by a binary-tree method

Chuanming Ju [a], Jianming Zhang [b,*], Rongxiong Xiao [b], Baotao Chi [c]

[a] School of Electromechanical and Automotive Engineering, Yantai University, 264005 Yantai, China
[b] State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, 410082 Changsha, China
[c] College of Mechanical Engineering, Shandong University of Technology, 255000 Zibo, China

ARTICLE INFO

Keywords:
Binary-tree
Adaptive
Mesh generation
Dual interpolation boundary face method

ABSTRACT

In order to realize the automation of surface mesh generation for arbitrary complex models, this paper proposes a binary-tree method (BTM) which can generate discontinuous grids. Compared with continuous grids, the adjacency relationship between discontinuous grids generated by BTM is more flexible, so it is easier to discretize "dirty" geometry without geometry repair than continuous grids. The BTM can adaptively generate meshes by features based on the curvature of curves and surfaces, and thus it does not need to construct a sizing field. Anisotropic meshes can also be directly generated by selecting the direction of subdividing meshes. Moreover, according to the periodic characteristics of periodic surfaces, the topological structure of the initial mesh can be designed to achieve mesh generation on periodic surfaces. Successful numerical examples are used to illustrate the effectiveness and robustness of the presented methods for meshing various types of geometric surfaces and reveal the excellent performance of the BTM. In addition, the BTM has been employed by the dual interpolation boundary face method (DiBFM) to solve 3D problems.

1. Introduction

Automatic mesh generation remains the main obstacle and technical bottleneck for CAE automation [1]. Towards the goal of CAE analysis automation, the competition among CAE software mainly focuses on pre-processing, such as geometry repair, mesh generation, etc. Compared with mesh generation, the geometric repair is more difficult, so it is necessary to develop a mesh generation method without a need for geometry repair. The study of mesh generation algorithms has become a separate research area, and a wide variety of algorithms have been proposed [2]. Unlike numerical calculation methods that have a well-developed theoretical basis, the theoretical basis of mesh generation algorithms is relatively weak. Different mesh algorithms are used for different structural shapes and cell types, most of which are empirical heuristic algorithms. Surface mesh generation algorithms have been comprehensively reviewed in the literature [1,2]. Currently, the popular mesh generation methods include advance front method(AFM) [3–6], Delaunay Triangulation Method (DTM) [7–10], mapping method [11, 12] and Quadtree method [13]. Among them, the AFM and DTM which are the main methods for continuous mesh generation, have great adaptability for arbitrary complicated geometry and achieve controllable mesh quality. However, there are also disadvantages such as: The mesh size transition is not flexible enough, additional input (sizing field) is required for generating non-uniform meshes and anisotropic meshes [1,2,14]; The parallelism of the algorithm is low, and the sub-domain decomposition [15] is required for parallel mesh generation; For "dirty" geometric models, the geometric repair is required, and it is difficult to fully automate the process from geometric model to mesh model [16]. For the mapping method, it is only applicable to simple and regular geometry, which requires a higher quality of geometric models and is more difficult to achieve automation.

The quadtree method can obtain regular rectangular meshes inside a surface. This method generates meshes quickly and does not require prior discretization of the boundary curves of the surface. When the geometric model contains "noise" such as slits and short edges, this method can generate surface meshes directly without geometric repair. This process requires little manual intervention and has a high degree of automation. For getting the continuous meshes required by the finite element method, the quadtree needs to fill the cells of different levels by a template method to ensure that the final mesh does not contain overhanging points. And the continuity requirement for meshes makes the boundary treatment more complicated. However, the double-layer interpolation boundary face method (DiBFM) [17–20], which does not require a continuous mesh, can approximate both continuous and

* Corresponding author.
E-mail address: zhangjianm@gmail.com (J. Zhang).

https://doi.org/10.1016/j.enganabound.2023.04.023
Received 20 January 2023; Received in revised form 25 March 2023; Accepted 16 April 2023
Available online 27 April 2023

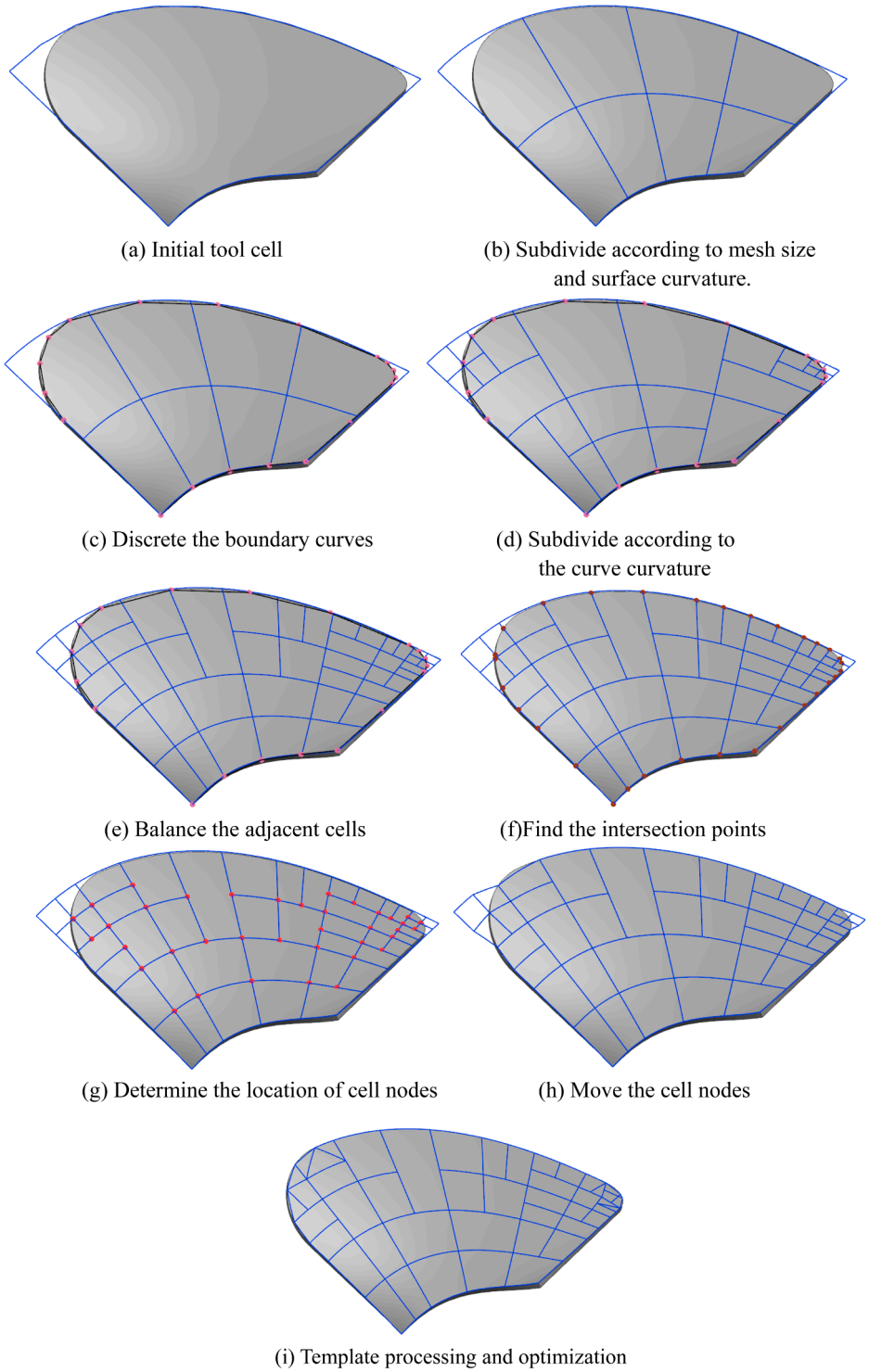


Fig. 1. The implementation process of the BTM [18].

discontinuous fields naturally and accurately. The method greatly reduces the requirement of the mesh shape and especially the continuity requirements for adjacent meshes. Without any requirement for mesh continuity, the tree structure approach is easier to realize automatic grid generation for any complex models. Thus, an adaptive binary-tree mesh method (BTM) will be proposed in this paper.

In view of the advantages of the quadtree method, the Binary-tree method inherits the advantages of the automation of the tree structure approach and avoids its shortcomings of requiring grid continuity. Different from the quadtree method which must subdivide a cell into four sub-cells, the BTM can select the subdivision direction to subdivide the cell into two sub-cells, so the BTM will be more flexible to generate anisotropic meshes. The method splits the entire initial root cell into two sub-cells and continues to recursively split the sub-cells until the curve and surface curvature requirements are met. This method has two most attractive advantages: one is the ability to generate discontinuous meshes, and allowing for the existence of so-called hanging points will radically ease the task of mesh generation. The other is the ability of the

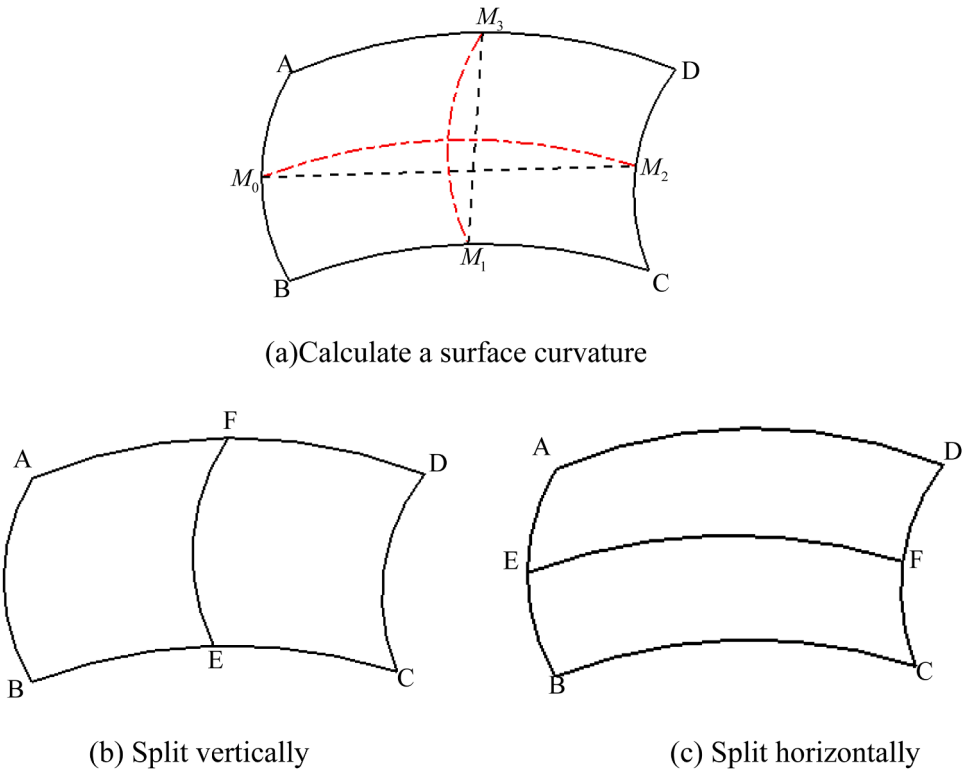


Fig. 2. Schematic diagram of subdivision by surface curvature.

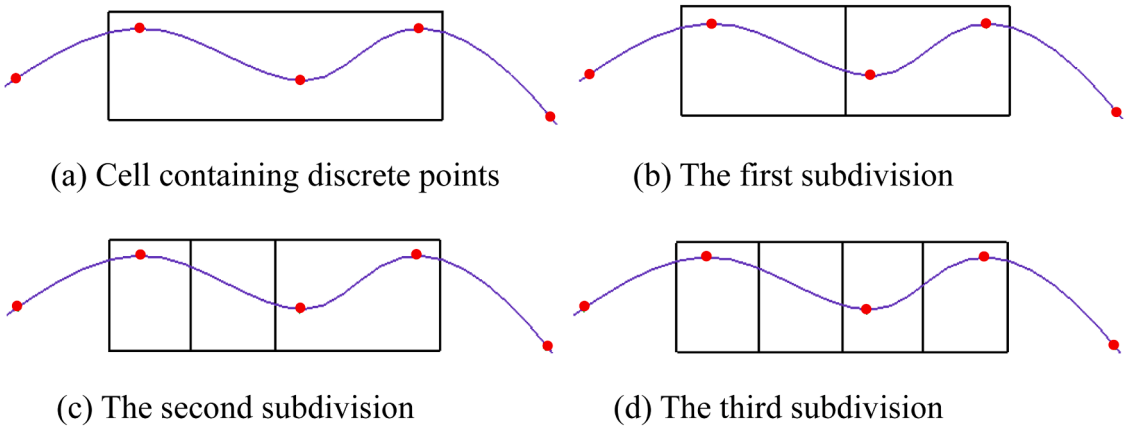


Fig. 3. Cell subdivision based on curve curvature.

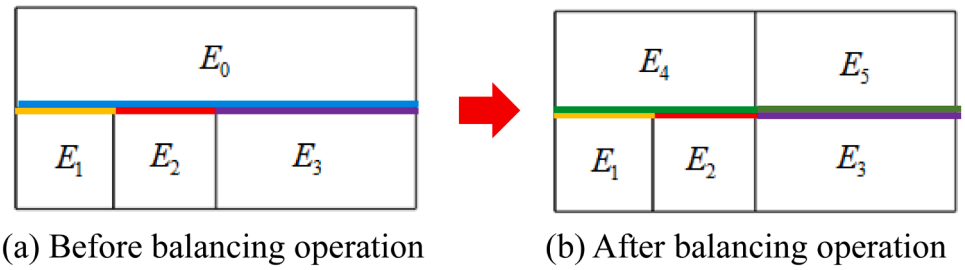


Fig. 4. Balancing adjacent cells.

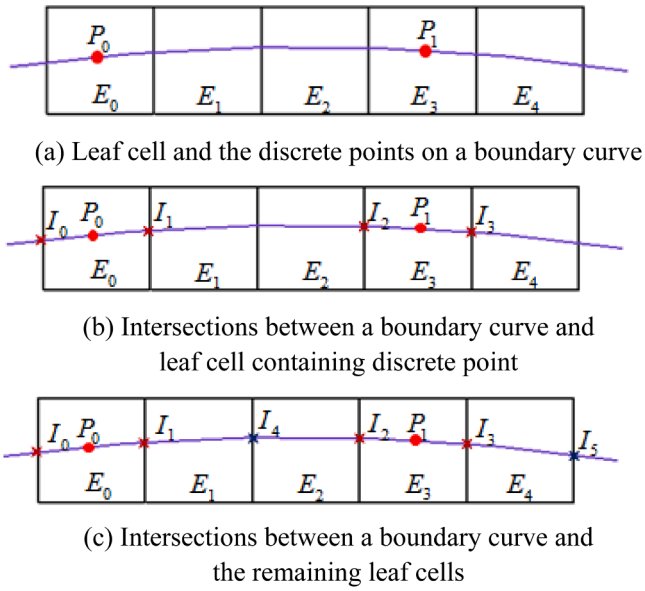


Fig. 5. The process of obtaining intersections between leaf element and boundary curve.

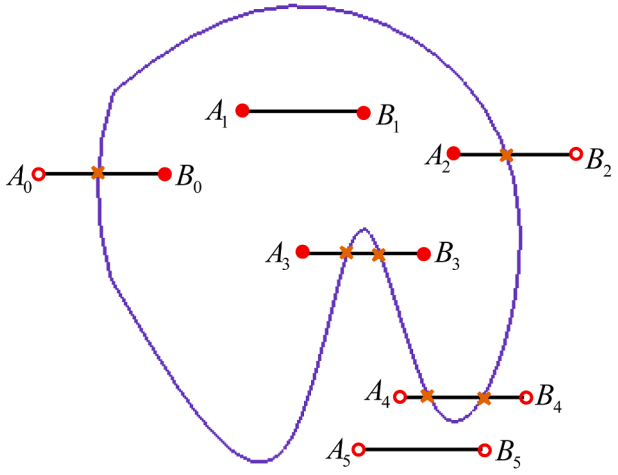


Fig. 6. Obtaining the position station of an endpoint by the number of intersections on a cell edge.

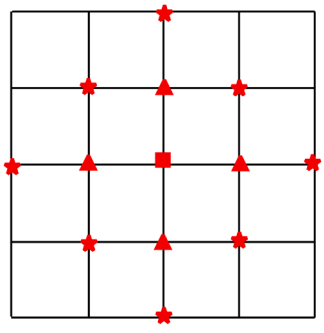


Fig. 7. Traversing the cell nodes.

method to adaptively refine the local mesh considering geometric factors. According to this advantage, this method gets rid of the need to create the background meshes as a sizing field. The quality of the meshes generated by the BTM is high, and most of the meshes are rectangular in shape. A variety of examples are presented to demonstrate the effectiveness and robustness of this algorithm and the ability to mesh periodic surfaces and geometric surfaces with "noise".

This paper is organized as follows. Section 2 describes the overall scheme of BTM. The details of BTM are given in Section 3 and Section 4. In Section 5, examples are given to illustrate the effectiveness of BTM. The paper ends with conclusions in Section 6.

2. Overall scheme of BTM

In order to demonstrate the implementation process of the BTM, the upper surface of a fan blade is used as an example to be discretized by BTM. The special process is shown in Fig. 1.

The subdivision process of BTM is performed in the parameter space of the surface. Firstly, the smallest rectangular enclosing box of the surface parametric domain is used as an initial root cell, as shown in Fig. 1(a). Secondly, the subdivision is performed based on a given mesh size and the curvature of the surface. The obtained leaf cells are shown in Fig. 1(b). Thirdly, the boundary curves of the surface are discretized to obtain the discrete points on the boundary curves, as shown in Fig. 1(c). The leaf cells are further subdivided according to the distribution of the discrete points on the boundary curves, as shown in Fig. 1(d). This process indirectly implements the subdivision of leaf cells by the curvature of the boundary curves. Fig. 1(e) shows the results of balancing the adjacent leaf cells. The size of adjacent leaf cells can be smoothly transitioned. The intersection points between the boundary curves and the cell edges can be calculated by their curvilinear functions. The intersection points are shown in Fig. 1(f), and marked as dark red points. The position state of all cell nodes can be judged from the intersection cases on the cell edges, and the interior mesh nodes of the surface are marked in red, as shown in Fig. 1(g). The mesh nodes close to a boundary curve are moved to the boundary curve by a shift point operation, as shown in Fig. 1(h). Finally, after template processing and optimization, the final mesh generation result is obtained, as shown in Fig. 1(i).

3. Details of BTM

In this section, more details of BTM are described. And the key point of mesh generation for periodic surfaces will be mentioned.

3.1. Initial root cell

The implementation of the BTM is performed in the parameter space of a surface, and the initial root cell is the smallest rectangular enclosing box covering the surface parameter domain. Surfaces have three expressions: explicit, implicit and parametric. From a computer graphics point of view, the parametric form of a surface is more convenient for computer representation and construction [21]. The parametric representation of a surface is shown as follows:

$$\begin{aligned} &P(u,v) = (x(u,v), y(u,v), z(u,v)) \\ &u_0 \le u \le u_1 \text{and} v_0 \le v \le v_1, \end{aligned} \tag{1}$$

where u and v are the parameter coordinates defined in the parameter domain of the surface.

It can be seen from Eq. (1) that the parameter coordinate range of u and v is a rectangle. The initial root cell is a rectangle box covering the parameter domain of the surface, and thus the parameter coordinates of four vertices of the initial root cell are set by (u_0,v_0),(u_0,v_1),(u_1,v_0),(u_1,v_1).

3.2. Subdivision based on curvature

After subdividing the root cell according to a given size, the

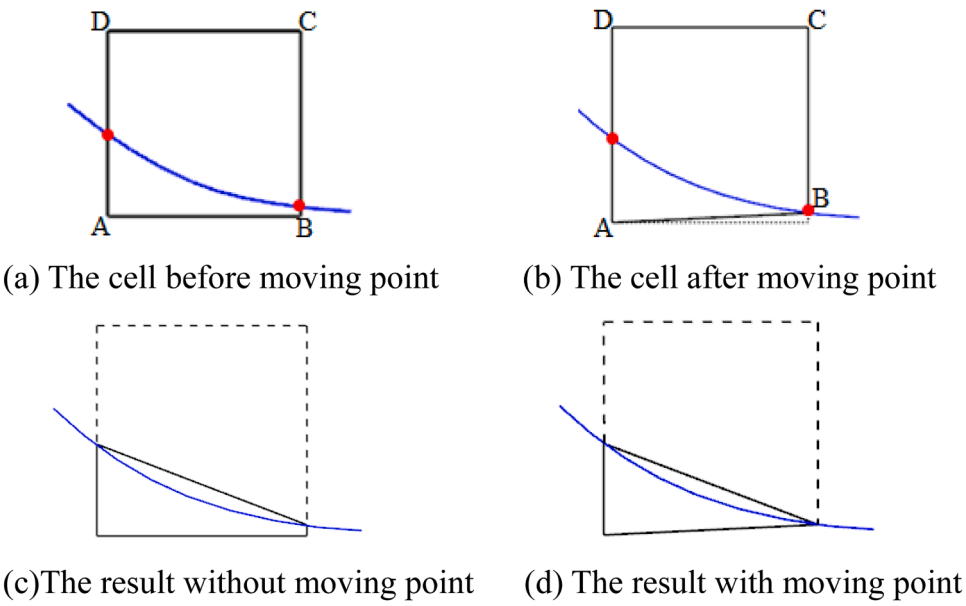


Fig. 8. The cell node close to an intersection point.

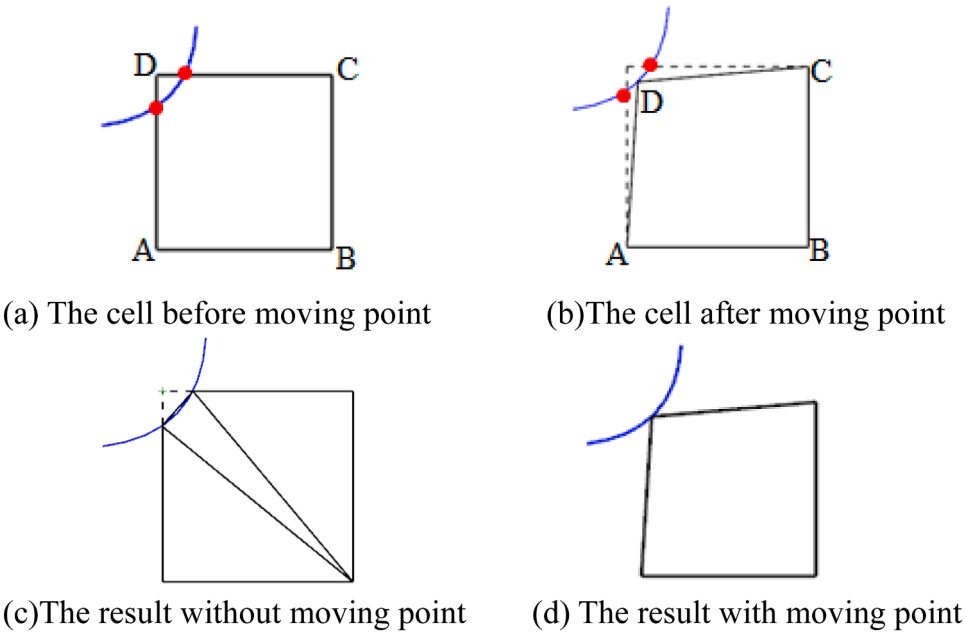


Fig. 9. The cell node close to two intersection points.

dimension of leaf cells already satisfies the given size. However, all leaf cells obtained by subdivision still need to satisfy the curvature of the surface and curves.

3.2.1. Subdivision based on a surface curvature

According to Eq. (2), whether the element satisfies the requirement of the curvature on a surface is judged. The curvature of a surface region where the cell is located can be approximately defined as follows:

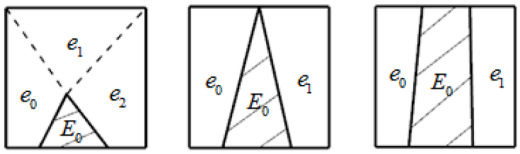
$$Ratio = \frac{l_{\overset{\frown}{ab}} - l_{\overline{ab}}}{l_{\overline{ab}}} \tag{2}$$

$l_{\overset{\frown}{ab}}$ is the distance between point a and point b on the surface, $l_{\overline{ab}}$ is the actual distance between point a and point b in physical space. The value of ratio reflects the degree of curvature of the surface between point a and point b, and they are positively correlated.

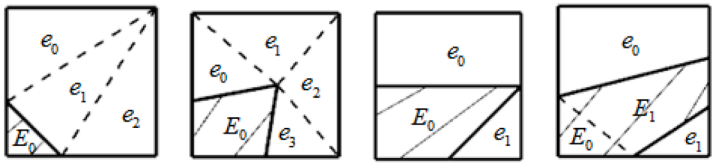
As shown in Fig. 2(a), the cell ABCD is a cell on a surface. Based on Eq. (2), if the ratio between the points M0 and M2 is larger than a given value, the cell is subdivided vertically. Then two cells (cell ABEF and cell CDFE) are generated and shown in Fig. 2(b). If the ratio between the points M1 and M3 is greater than the given value, the cell is split horizontally to produce the two cells shown in Fig. 2(c). If the two ratios are less than the given value, the cell ABCD is not subdivided.

3.2.2. Subdivision according to a curve curvature

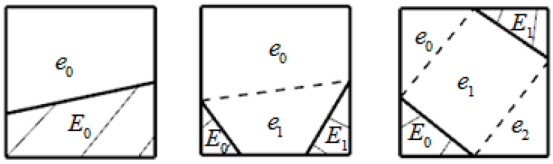
The leaf cells conforming to the surface curvature are obtained in the previous section, and the leaf cells close to a boundary curve also need to satisfy the curve curvature. In order to make the cell density vary with



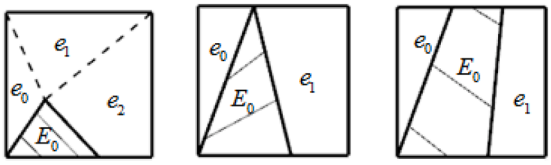
(a) (4,0,0) and (0,4,0)



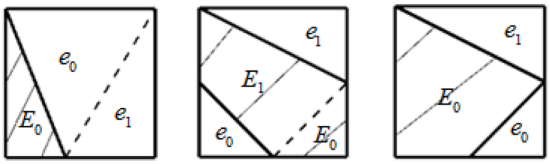
(b) (3,1,0) and (1,3,0)



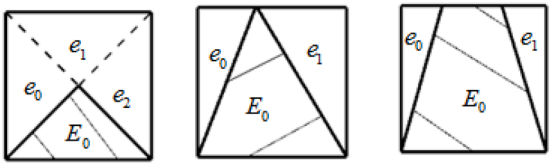
(c) (2,2,0)



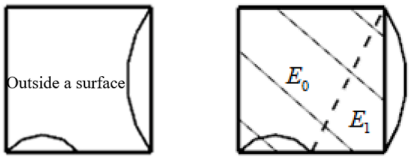
(d) (3,0,1) and (0,3,1)



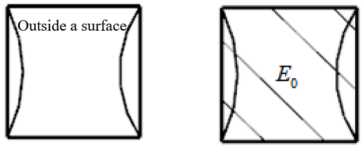
(e) (2,1,1) and (1,2,1)



(f) (2,0,2) and (0,2,2)



(g) (1,0,3) (h) (0,1,3)



(i) (0,0,4)

Fig. 10. Classification of templates.

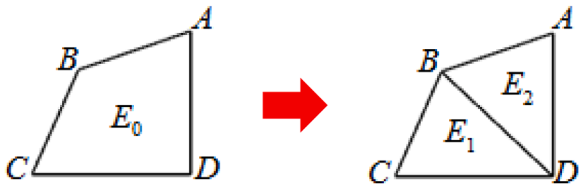


Fig. 11. Splitting a cell owning large corners.

the change of curvature of the boundary curve, we first discretize the boundary curve based on a dichotomous method (see Section 4.1), and get some discrete points. The distribution of discrete points reflects the change in curve curvature. Thus, we further subdivide the leaf cells according to the distribution of these discrete points. And ensure that the distribution of the leaf cells can better fit the curvature change of the boundary curve.

The discrete points obtained in the previous step will be located. The discrete points may fall on the vertices, edge or inside of a cell. Finally, the cells containing discrete points are subdivided if they satisfy a subdivision rule. This subdivision rule is as follows: (1) Ensure that there are no more than two discrete points within each leaf cell; (2) The size of the leaf cell is no larger than the average length of the discrete curve segments connected to all discrete points within that cell.

Fig. 3(a) illustrates a leaf cell containing three discrete points of a boundary curve. According to the subdivision rule (1), the leaf cell needs to be subdivided, and the subdivision results are shown in Fig. 3(b). According to the subdivision rule (2), the dimension of the leaf cell is larger than the average length of the discrete curve segments connected to all discrete points within the cell, so the leaf cell continues to be subdivided until the subdivision rule is not satisfied, as shown in Fig. 3 (c) and Fig. 3(d).

3.3. Balancing adjacent cells

The purpose of balancing adjacent cells is to achieve a smooth size transition between adjacent meshes. The BTM ensures the smooth size transition by controlling the level difference between adjacent cell edges does not exceed 1.

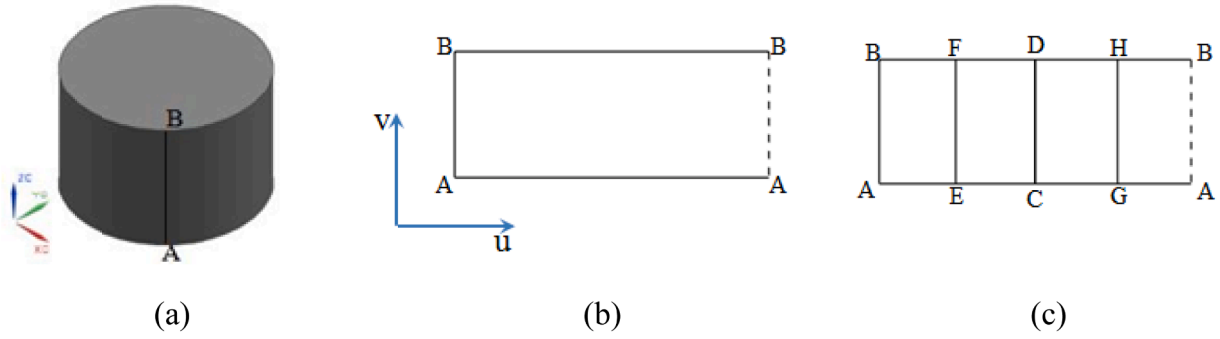


Fig. 12. Root cell of a periodic surface in the U-direction: (a) A periodic surface in the U-direction (side face of cylinder); (b)Initial root cell in parametric space; (c) Initial subdivision for root cell.

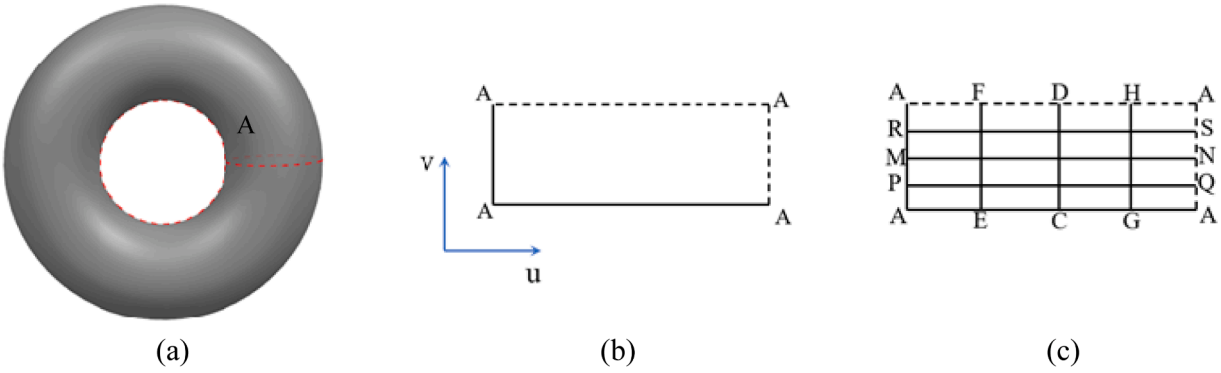


Fig. 13. Root cell of a periodic surface in the UV-direction: (a) A periodic surface in the UV-direction; (b)Root cell in parametric space; (c)Initial subdivision for root cell.

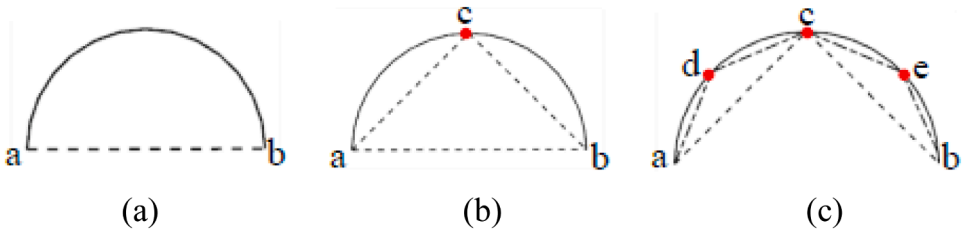


Fig. 14. Process of a dichotomous method: (a)Initial curve; (b) Discretize results for once; (c)Discretize results for twice.

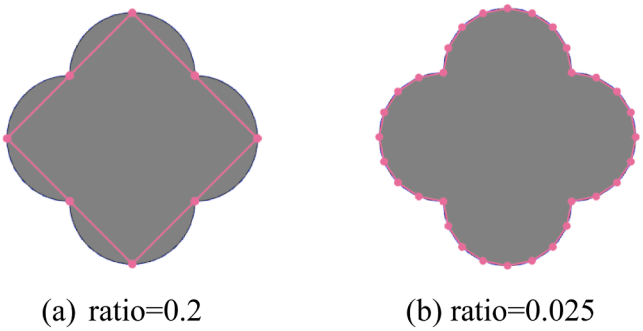


Fig. 15. Discretized results of curve according to different given ratios.

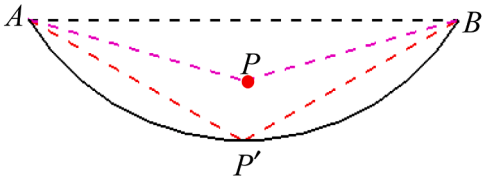


Fig. 16. An arbitrary point lies in the area between a curve and its chord.

As shown in Fig. 4(a), the edge of cell E_0 differs from the edge of cell E_3 by one level. However, the edge of cell E_0 differs from the edge of cell $E_1(E_2)$ by two levels. Therefore, cell E_0needs to be subdivided into cell E_4 and cell E_5, as shown in Fig. 4(b). The hierarchy of edges shared by all neighboring cells after balancing operation does not differ by more than one.

3.4. Finding the intersection points between boundary curves and cell edges

It is necessary to calculate the intersection points between boundary curves and cell edges for two reasons: First, the number of intersection points on a cell edge can be used to determine the position of the two endpoints of the cell edge with respect to the meshed surface. Second, the intersection points on the cell edge can be applied to treat the boundary cells according to a set of templates.

Finding the intersections between a boundary curve and a cell edge is divided into the following steps: firstly, find the leaf cell containing the discrete points of the boundary curve; secondly, calculate the intersection points between cell edges and curve segments connected by these discrete points, and lastly compute the intersection points between the boundary curve and the leaf cells that do not contain the discrete points.

As shown in Fig. 5(a), the discrete points P_0 and P_1 are marked as red points on a boundary curve, and they are located in the leaf cellsE_0andE_3respectively. To get the intersections between the boundary curve and the cell containing discrete points, we calculate the intersections between the cell edge and the curve segments connected by discrete points. The process of calculating intersections is performed in parametric space. The curve segment can be determined as a quadratic curve based on its two endpoints and its midpoint in the parameter space. Moreover, the two endpoints of a cell edge can determine a straight line. The intersection between the quadratic curve and the straight line is calculated to obtain its parametric coordinates. The parametric coordinate can then be mapped via the parametric surface equation into the three-dimensional coordinates in physical space. The intersection points are shown in Fig. 5(b). I_0, I_1, I_2, I_3 are the intersection points marked as crimson crosses.

For getting the intersections between boundary curves and the cells which do not contain discrete points, we must rely on a principle of entry and exit. As shown in Fig. 5(a), cellE_1,cell$E_2$and cellE_4 do not contain discrete points. After getting the intersections between boundary curves and cells containing discrete points, the intersection points already exist on the edges of these three cells, as shown in Fig. 5(b). The existence of an intersection point on a cell edge means that a curve segment penetrates into the cell, and so there must be another intersection point between the curve segment and the cell. This is because the curve segment should penetrate out through the cell. That is the principle of entry and exit. According to this principle, all intersections between a boundary curve and the cells can be obtained, as shown in Fig. 5(c).

3.5. Determining the position of mesh nodes

It is crucial to determine the position of the mesh nodes relative to a

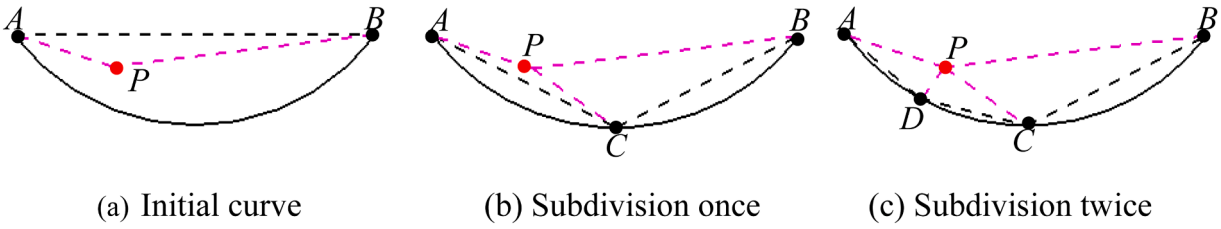


Fig. 17. Subdivision process of curve while a point lies in the area between a curve and its chord.

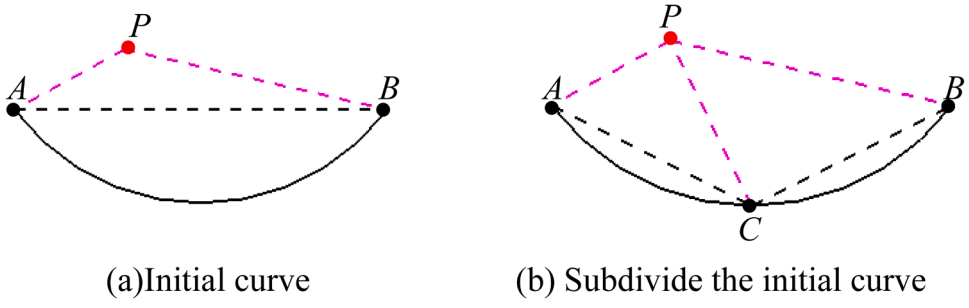


Fig. 18. Subdivision process of curve while a point on the same side of a curve and its chord.

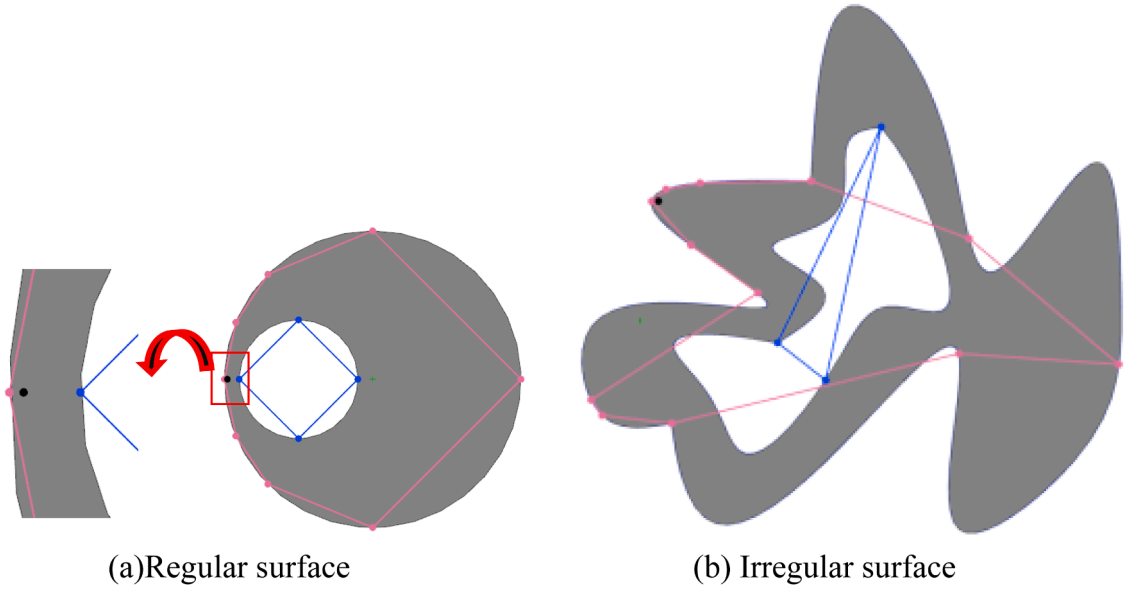


Fig. 19. Subdivision results of Loops on a surface.

Table 1
Mesh statistic of periodic surface.

Surface Type	Sphere				Cone				Ring			
	AFM		BTM		AFM		BTM		AFM		BTM	
Mesh Number	120	16978	112	19936	214	23610	212	21271	278	12356	288	12288
Time(s)	0.047	17.766	0.016	2.639	0.031	3.749	0.015	2.575	0.188	11.477	0.031	1.412

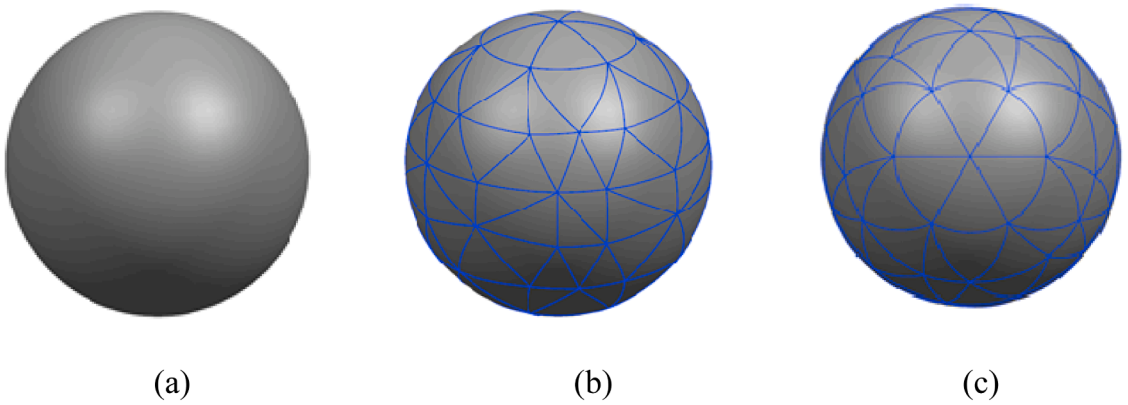


Fig. 20. Results of sphere meshed by AFM(Mesh Number:120): (a) Sphere; (b) Front view; (c)Top view.

surface. The purpose of this process is to prepare for the subsequent processing of treating the boundary cells by a set of templates. What's more, based on the information about the position of the mesh nodes, the cell can be classified as the following: inner mesh, outer mesh and boundary mesh. The inner meshes are retained, and the boundary meshes are treated to obtain the final mesh.

The steps to determine the position information of a cell node are as follows: The first step is to determine the positional relationship of the object node relative to the surface. Based on the position of the object node, the boundary loop of the face is discretized by another dichotomy method (see Section 4.2), and turned into a corresponding polygon. Combining the position of point relative to polygon algorithm, the position relationship between the object node and surface can be obtained. In the second step, according to the position of the object node, the position of other cell nodes which connected to the object node by cell edges can be determined, and special details are shown in Section 3.5.1. The third step is to repeat the second step for all cell nodes whose location information has just been judged justly, recursively until all location information of cell nodes has been determined (see Section 3.5.2 for details).

3.5.1. *Determining the location of a cell node*

If the position of one endpoint of a cell edge is known, the position of another endpoint can be obtained from the number of intersection points on the cell edge. As shown in Fig. 6, if the number of intersection points on a cell edge is an odd number, the position state of one endpoint is opposite to that of another endpoint on the cell edge. If the number of intersection points on a cell edge is an even number, the position state of

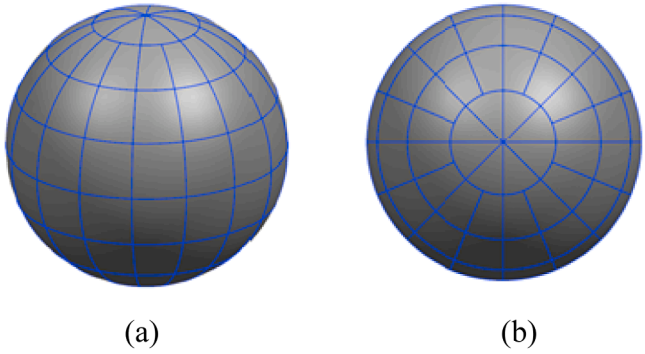


Fig. 21. Results of sphere meshed by BTM (Mesh Number:112): Front view; (b) Top view.

one endpoint is the same as that of another endpoint on the cell edge.

As shown in Fig. 6, the blue line is a boundary curve of a face, and the brown-cross points are the intersections between cell edges and the boundary curve. The solid red points indicate the internal cell nodes, and the hollow red points represent the external cell nodes. When the number of intersection points on a cell edge is an odd number, the two endpoints of the cell edge own opposite position states, such as the cell edge A_0B_0 and the cell edgeA_2B_2.When the number of intersections on a cell edge is an even number, the two endpoints of the cell edge have the same position status, such as edge cellA_1B_1, cell edgeA_3B_3,cel1 edgeA_4B_4 and cell edgeA_5B_5.

3.5.2. *Traversing cell nodes to determine their location information*

As shown in Fig. 7, the square node is the initial cell node whose position state is known. The position state of the triangular nodes around the initial node is determined by judging the number of intersection points of a corresponding cell edge. After judging the position state of the triangular nodes, the position state of the star nodes around each triangular node is then judged. This process proceeds recursively until the position state of all cell nodes is obtained.

3.6. *Moving points*

Based on the distance between an intersection point and two endpoints of a cell edge, it is determined whether the two endpoints need to be moved further to the intersection point respectively. If an endpoint of the cell edge is close to an intersection point, the endpoint of the cell edge is moved to the intersection point (see Fig. 8(a)(b)). If an endpoint of the cell edge is close to two intersection points at the same time, the endpoint is moved to the position between two intersection points on a boundary curve, as shown in Fig. 9(a)(b).

The processing of moving points avoids the formation of the sub-cells with small short edges in the subsequent processing of treating boundary cells. Fig. 8(d) and Fig. 9(d) show the new cells generated by treating the boundary cells after an operation of moving points. Fig. 8(c) and Fig. 9

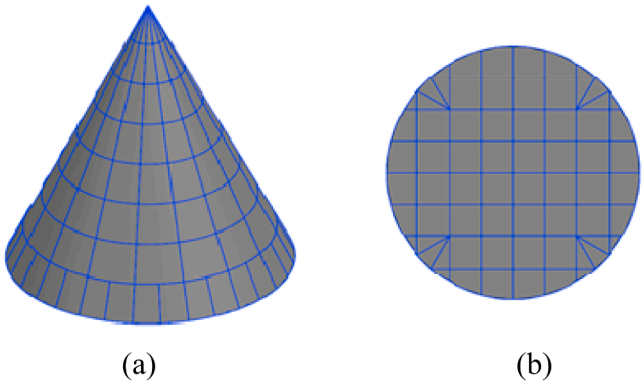


Fig. 23. Results of Cone meshed by BTM (Mesh Number:212): (a) Front view; (b) Bottom view.

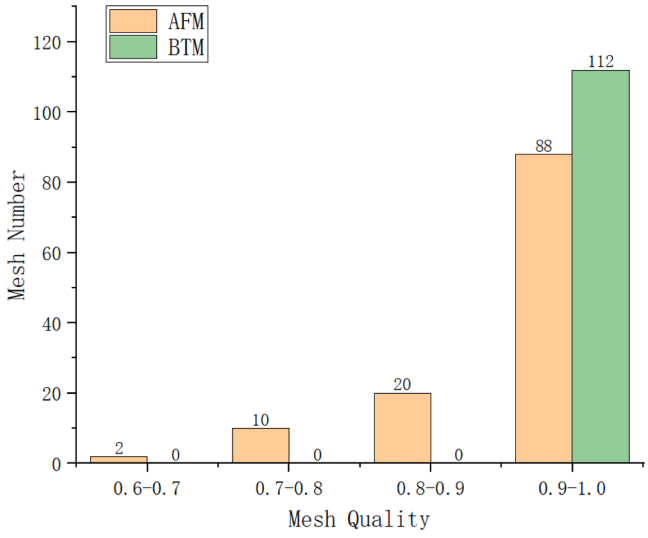


Fig. 24. Mesh quality statistics of sphere (mesh number of AFM:120, mesh number of BTM 112).

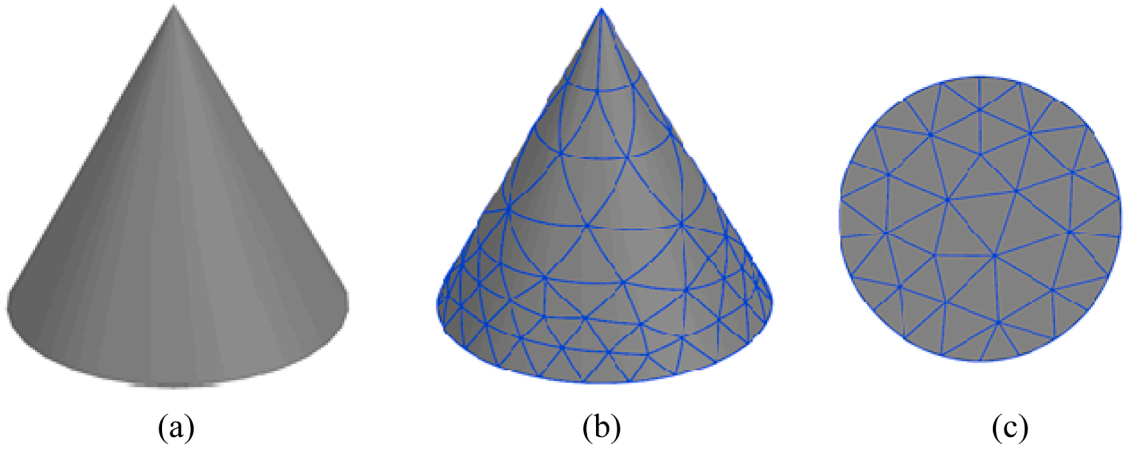


Fig. 22. Results of cone meshed by AFM(Mesh Number:214): (a) Cone; (b) Front view; (c)Bottom view.

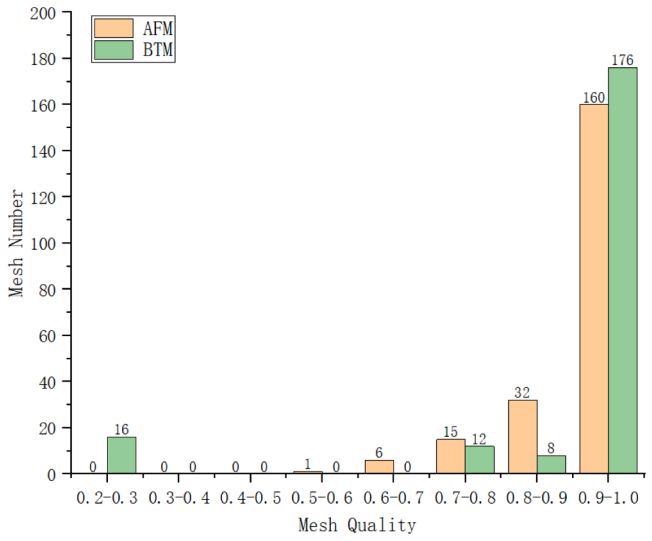


Fig. 25. Mesh quality statistics for cone (mesh number of AFM: 214, mesh number of BTM 212).

(c) show the new cells generated by processing the boundary cells without moving points, which have small short edges and own poor shape.

3.7. Treating boundary cells by a set of templates

After determining whether a cell is a boundary cell, the cell needs to be classified by the position states of its four nodes. And then the boundary cell is treated by the corresponding template. The templates are broadly classified as shown in Fig. 10.

Fig. 10 shows the templates corresponding to the different types of boundary cells. The (a, b, c) refers to the type of a boundary cell. The letter "a" denotes the number of cell nodes which are outside the surface, the letter "b" is the number of cell nodes located inside the surface, and the letter "c" represents the number of cell nodes on the boundary curves of the surface. The letters "E" and "e" denote the new cells generated after treating the boundary cell with the complementary templates. For example, (1,3,0) denotes that the number of cell nodes located outside the surface is 1, the number of cell nodes located inside the surface is 3, and the number of cell nodes located on the boundary curves is 0. The letter "E" represents the new sub-cell generated by the template (1,3,0), while "e" is the new sub-cell generated by the template (3,1,0). It is worth noting that the cell in Fig. 10(g) and Fig. 10(i) is outside the face, so it is deleted and not remained as a final cell.

3.8. Optimizing mesh quality

After treating boundary cells by templates, the cells with large corners will be subdivided. The purpose of this process is mainly to further improve the quality of a final generated mesh.

As shown in Fig. 11, the degree of $\angle ABC$ in cell E_0 is greater than 120, which we define as a large-angle. Therefore, we subdivide the cell E_0 to eliminate large angles, and generate sub-cell E_1 and E_2.

3.9. Periodic surface

Compared with 3D volume mesh generation, one of the difficulties in surface mesh generation is the mesh generation of periodic surfaces. The mesh generation process of periodic surfaces encounters a special problem that does not exist in the mesh generation of non-periodic surfaces, namely the missing boundary information of periodic surfaces with respect to their parameter domains [22]. The boundary of a non-periodic surface is complete with respect to its parametric domain. That is, the mapping of boundary curves of a non-periodic surface can constitute a complete 2D parameter domain on the parameter space. For

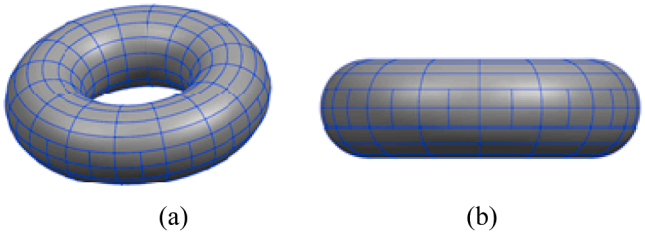


Fig. 27. Results of ring meshed by BTM(Mesh Number:288): (a) Front view; (b) Side view.

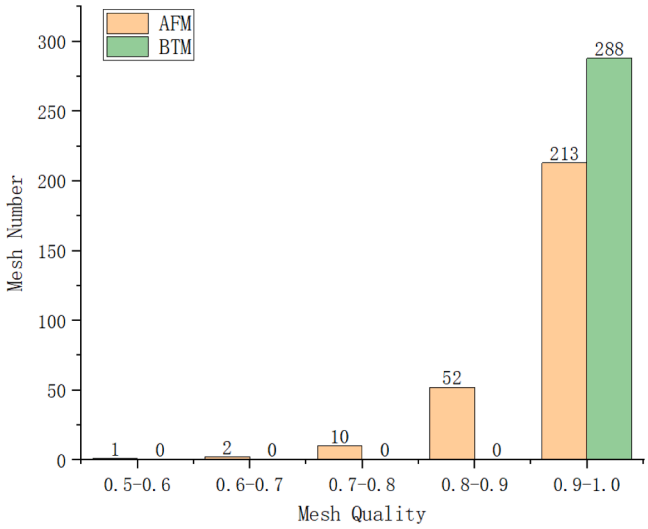


Fig. 28. Mesh quality statistics for ring (mesh number of AFM: 278, mesh number of BTM: 288).

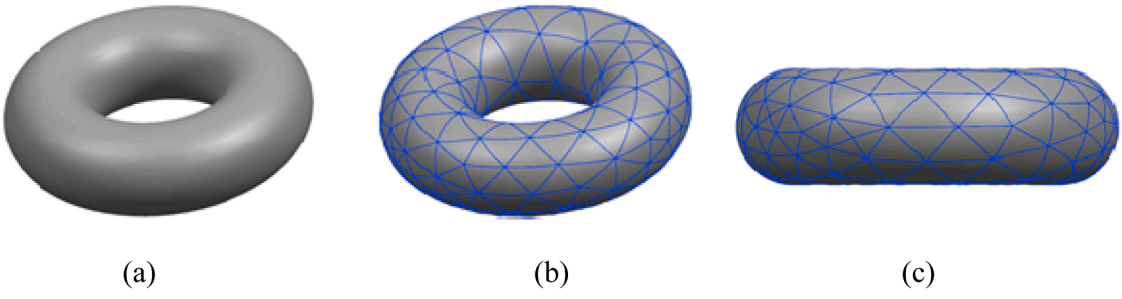


Fig. 26. Results of ring meshed by AFM(mesh number:278): (a) Ring; (b) Front view; (c)Side view.

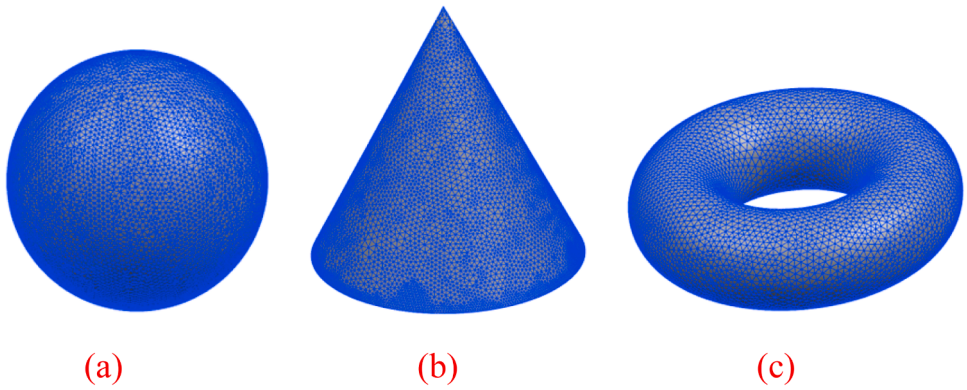


Fig. 29. Mesh results of AFM:(a) Sphere (mesh number: 16978); (b) Cone (mesh number: 23610); (c) Ring (mesh number: 12356).

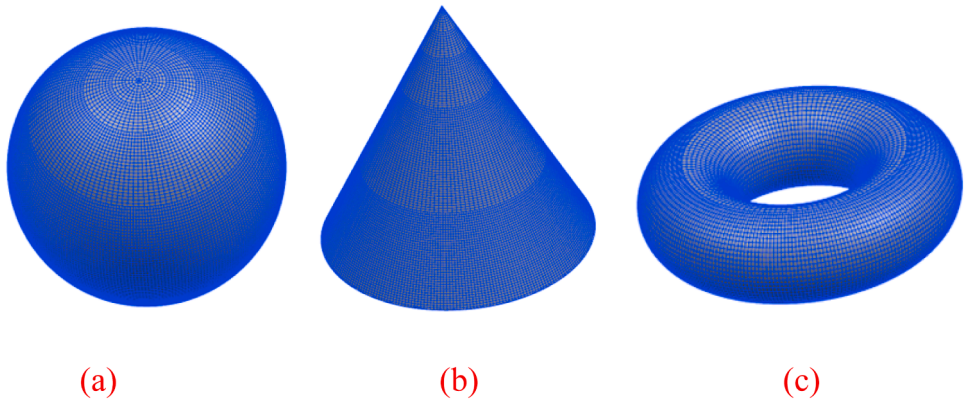


Fig. 30. Mesh results of BTM:(a) Sphere (mesh number:19936); (b) Cone (mesh number:21271); (c) Ring (mesh number:12288).

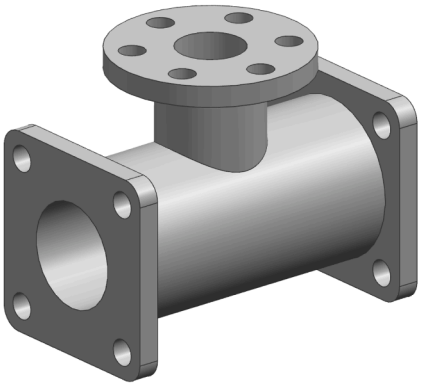


Fig. 31. Three-way pipe.

the periodic surface constructed by the geometric modeling system, the boundary of its parameter domain is incomplete, that is, the mapping of the 3D boundary curve cannot form a complete 2D parameter domain on the parameter space.

To solve the above problem, the virtual boundary adjustment algorithm and the shifting-AFT algorithm are proposed in the literature [22, 23]. Employing these methods, the Advance front method (AFM), one of the mainstream methods of mesh generation, can be directly applied to the mesh generation of closed surfaces. However, adding virtual boundaries is difficult and needs a set of shifting operators.

In the implementation of the BTM, the initial root cell in the parametric domain is continuously subdivided. For the periodic surface, we still maintain the topological relationship of the initial root cell, which consists of four cell edges. However, the opposite edges in the periodic direction can be mapped to the same location in 3D space. For example, taking a cylindrical surface (a periodic surface in the U-direction) as an example, its parameter equation is as follows:

$$\begin{cases} X = r\cos u \\ Y = r\sin u \\ Z = v \end{cases} \quad \begin{pmatrix} 0 \le u \le 2\pi \\ a \le v \le b \end{pmatrix} \tag{3}$$

Where *a*, *b*, *r*are known quantities. As can be seen from the

Table 3
Statistic of meshes generated by BTM for three-way pipe.

Mesh Size	0.8	0.6	0.45	0.35	0.25	0.15	0.1
Mesh Number	3945	6474	9019	13949	22041	80172	137533
Time(s)	0.485	0.691	0.922	1.36	2.359	10.469	16.735

Table 2
Statistic of meshes generated by AFM for three-way pipe.

Mesh Size		0.75	0.6	0.45	0.35	0.25	0.15	0.1
Mesh Number		4122	6690	8980	16670	29180	82432	150122
Time(s)	Oct-Tree	0.281	0.329	0.578	1.313	1.844	6.078	9.672
	AFM	0.231	0.296	0.438	0.625	1.078	3.063	7.141

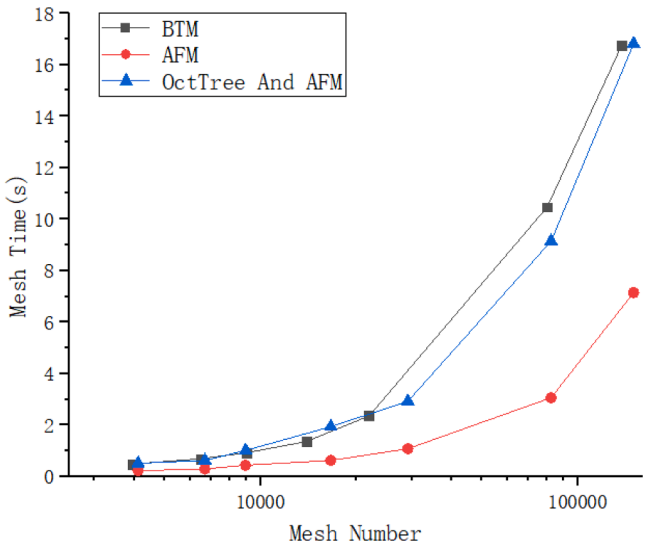


Fig. 32. Comparison of CPU time between BTM and AFM process.

Table 4
Mesh statistic for three-way pipe.

Method		Time (s)	Mesh Number	Average value of mesh quality
Total Process of AFM	Oct AFM	0.281 0.231	4122	0.925
BTM		0.485	3945	0.977
ANSYS workbench			3920	0.884

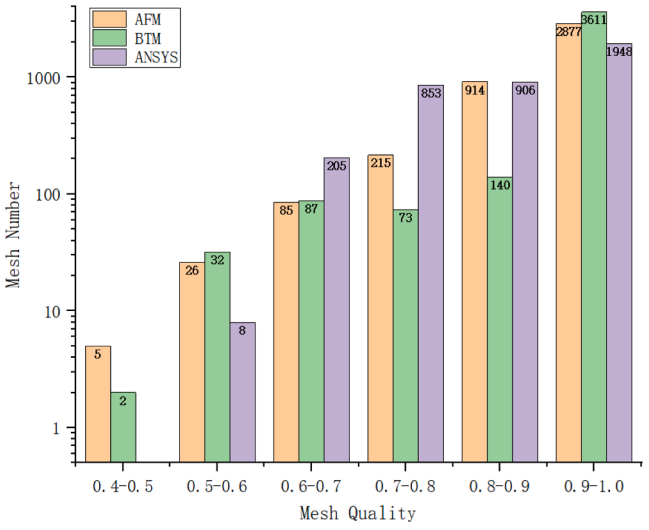


Fig. 33. Mesh quality statistics for tree-way pipe (mesh number of AFM: 4122, mesh number of BTM: 3945).

parametric equation, the points(0, v)and(2π, v) in the parametric domain are the same location in 3D space. Therefore, considering the periodicity of periodic surfaces and the pointer characteristics of C++, when the two sides of the parameter domain of periodic surfaces are taken as the opposite edges of the root cell, the opposite edges are represented by the same pointer variable.

As an example, the root cell of a periodic surface in the U-direction is shown in Fig. 12. Though the left and right edges are still the same edge in 3D physical space, the topology of the root cell in parametric space remains unchanged, and the root cell still consists of four edges. Implied by the C++ program, the left and right edges are represented by the same pointer variable. Aim to break the periodicity of the surface, the root cell needs to be first subdivided into four sub-cells along the periodic direction (see Fig. 12(b)). Then the subsequent subdivision process of the sub-cell is the same as that of a cell on a non-periodic surface. Similarly, the construction and subdivision of the root cell of a periodic surface in the UV-direction are shown in Fig. 13.

4. Two dichotomy methods for curve discretization

4.1. A dichotomy method based on curve curvature

The main idea of the dichotomy method is to discretize a boundary curve of a surface into two parts continuously until a given subdivision condition is not satisfied, and then stop the subdivision procession. The given subdivision condition will determine the final number of discrete segments of the curve. The process is shown in Fig. 14.

Arc *ab* is a spatial curve segment, and the degree of its curvature can be calculated by Eq. (2) in Section 2.2.1. In this section, $l_{\overset{\frown}{ab}}$ (Eq. (2)) is the length between point a and point b along the curve segment, $l_{\overline{ab}}$ (Eq. (2)) is the actual distance between point a and point b in physical space.

If the *Ratio* in Eq. (2) is larger, it reflects that the curvature of the arc is greater, and the string *ab* cannot approximate the arc *ab*. If *Ratio* is larger than a given ratio, the arc is divided into arc *ac* and arc *bc*, this process continues recursively until the *Ratio* of all arcs is less than the given ratio. This method is simple and stable, and can adjust the given ratio to get different discretization results, as shown in Fig. 15.

4.2. A dichotomy method based on a point

According to the position of a point relative to a curve, the algorithm discretizes the object curve. The key of this algorithm is to ensure that the position relationship between the point and the discretized curve remains unchanged.

How to judge whether a curve should be discrete based on a point? It is necessary to judge whether the curve arc and its chord are still on the same side of the point. We find that if the point is located between the curve arc and its chord, the sum of the length of the connecting lines between the point and the two ends of the curve must be less than the length of the curve.

As shown in Fig. 16, an arbitrary point P lies in the area between the curve AB and the string AB. Now, we should prove that the sum of the length of line segment AP (L_{AP}) and the length of line segment BP (L_{BP})is always less than the arc length of curve AB ($L_{\overset{\frown}{BP}}$).

As can be clearly seen from Fig. 16, the area of triangle ΔABP is less than the area of triangle $\Delta ABP'$, and thus the perimeter of triangle ΔABP is less than the perimeter of triangle$\Delta ABP'$. However, these two triangles share the same edge AB, and thus $L_{AP} + L_{BP} < L_{AP'} + L_{BP'}$. The length of the curve segment $L_{\overset{\frown}{AP'}}$ is larger than the length of the line segment $L_{AP'}$, and the length of the curve segment $L_{\overset{\frown}{BP'}}$ is larger than the length of the line segment $L_{BP'}$. Thus, the sum of the lengths of line segment AP and line segment BP is less than the length of the curve segment AB, *that is*$L_{AP} + L_{BP} < L_{\overset{\frown}{AP'}} + L_{\overset{\frown}{BP'}}$.

Finally, we get the conclusion that as long as the point is located between a known curve and its chord, the sum of the length of the connecting lines between the point and the two ends of the curve is less than the length of the curve. The converse negative proposition of this conclusion is that if the sum of the length of the connecting line between a point and the two ends of a curve is greater than the length of the curve, the point must not lie in the area between the known curve and its chord.

According to the above conclusions, we propose a dichotomy method based on a point. If the sum of the lengths of the connecting lines between the target point and the two ends of a curve is less than the length

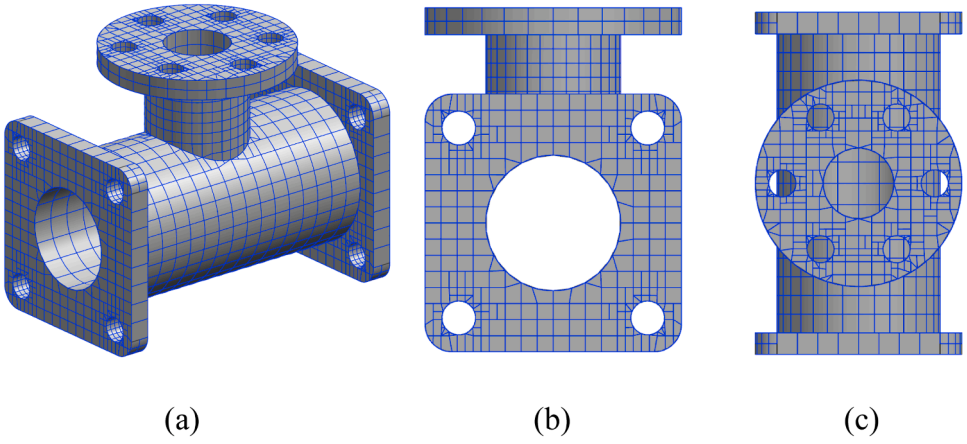


Fig. 34. Results of three-way pipe meshed by BTM:(a) Front view; (b)Side view; (c)Top view.

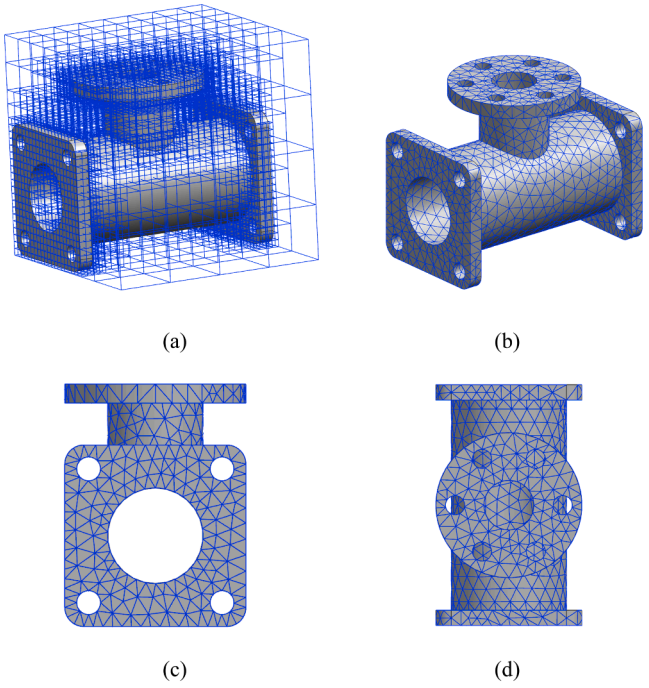


Fig. 35. Results of three-way pipe meshed by AFM: (a) Sizing field constructed by octree refinement; (b) Front view; (c)Side view; (d)Top view.

of the curve, the curve is divided into two curve segments, and then continue to judge whether to subdivide the two sub curve segments, until the sum of the lengths of the connecting lines between the target point and the two ends of all sub curve segments is greater than the length of the sub curve segment, then stop subdivision. According to the different positions of target points, the curve subdivision process is shown in Fig. 17 and Fig. 18.

The dichotomy method can be applied to judge the position relationship between a point and a loop on a plane. The loop is discretized by the dichotomy method to get a polygon. The position relationship between the point and the polygon is consistent with the positional relationship between a point and the loop. By judging the position relationship between the point and the polygon, the position relationship between the point and the loop can be determined. Fig. 19 shows

Fig. 37. Cup.

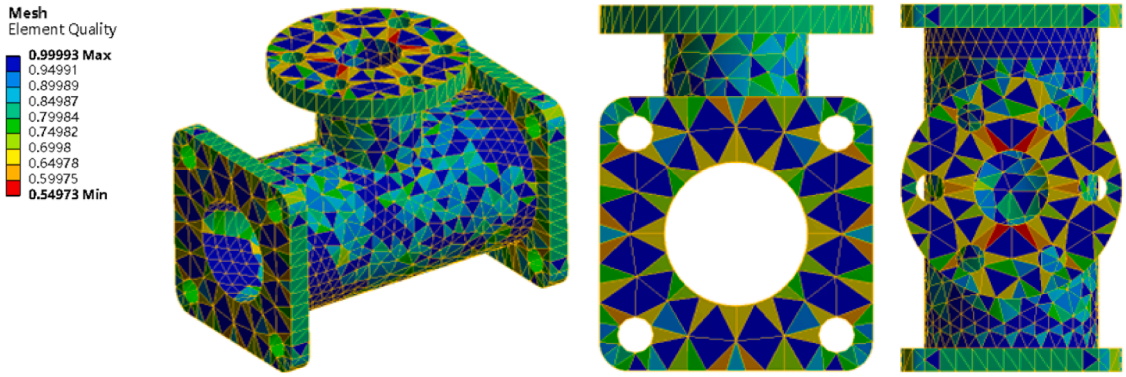


Fig. 36. Results of three-way pipe meshed by ANSYS workbench.

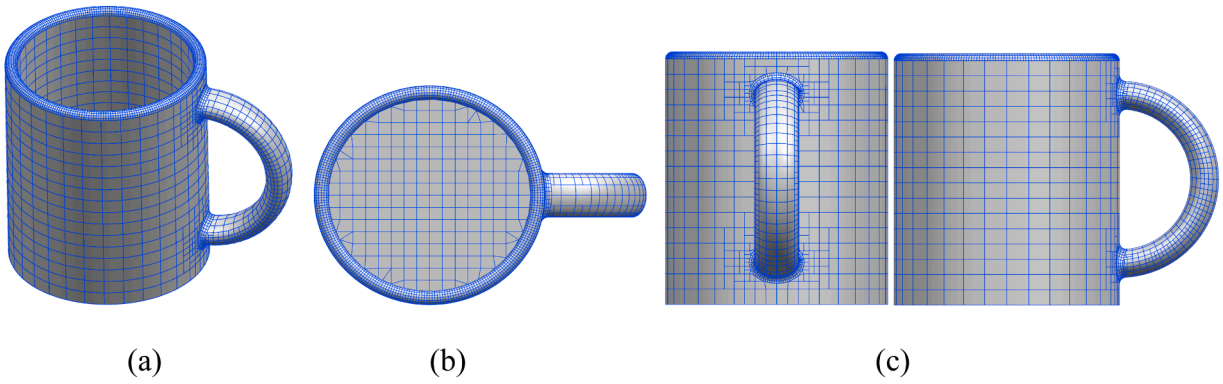


Fig. 38. Results of the cup meshed by BTM:(a) Front view; (b) Top view; (c) Side view.

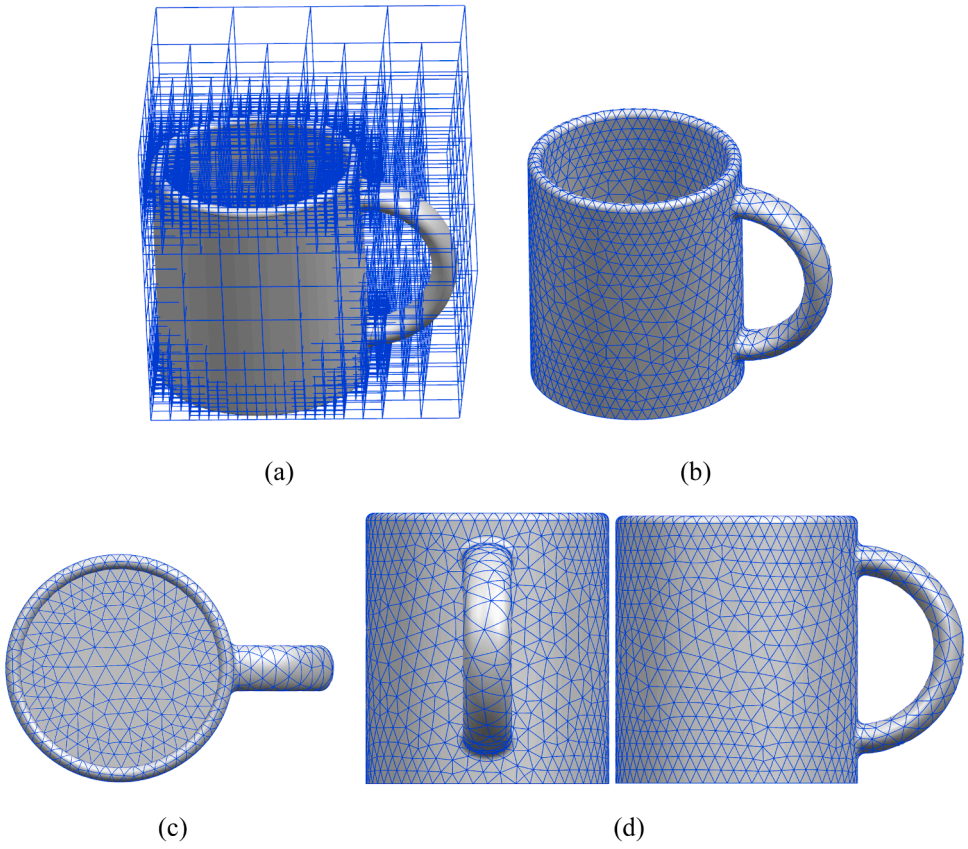


Fig. 39. Results of the cup meshed by AFM: (a) Sizing field constructed by octree refinement; (b) Front view; (c) Top view; (d) Side view.

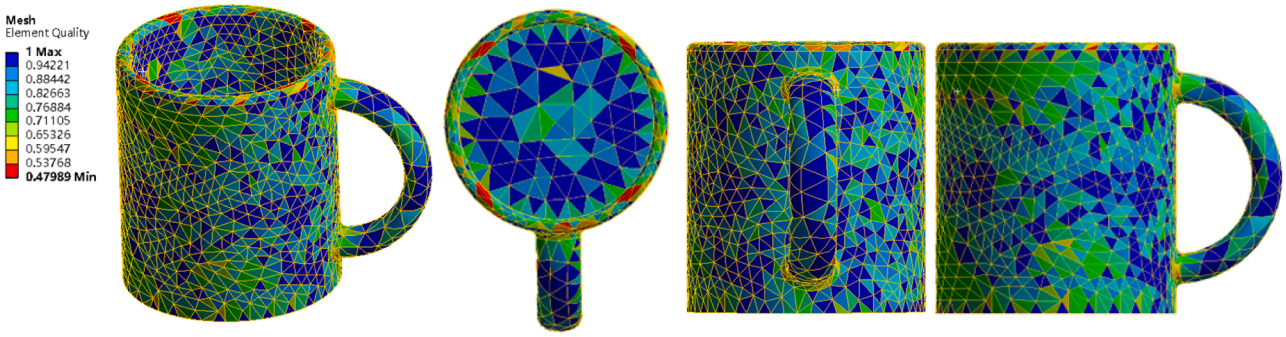


Fig. 40. Results of the cup meshed by ANSYS.

Table 5
Mesh statistic for cup.

Method		Time (s)	Mesh Number	Average value of mesh quality
Total Process of AFM	Oct	0.218	4586	0.985
	AFM	4.751		
BTM		6.205	4622	0.997
ANSYS workbench			4436	0.882

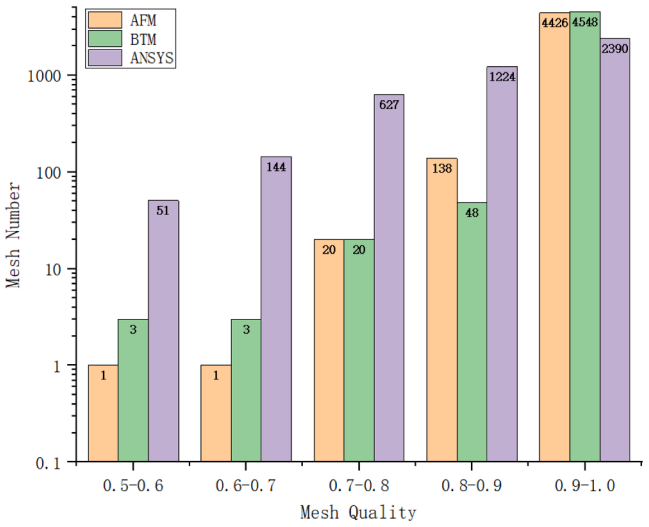


Fig. 41. Mesh quality statistics for the cup.

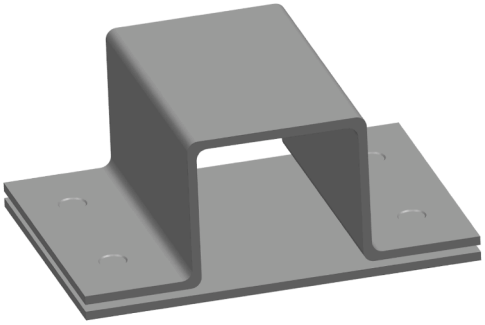


Fig. 42. Spot-welding model.

the discrete result of loops on a surface with the dichotomy method.

5. Numerical examples

In this section, meshes results of typical geometric models are given to demonstrate the effectiveness and robustness of the BTM. These examples contain periodic and non-periodic surface models. These models are meshed by BTM and AFM, respectively. The effectiveness of BTM is illustrated by comparing it with the AFM. Based on some classical literature on AFM, our team implemented the AFM in C++. The AFM program has been integrated into a CAE software developed by our team, and a large number of geometric models are meshed by this software to verify the stability of the AFM program. Since the AFM and BTM programs are both implemented by our team, thus the comparison between them is relatively fair. The CPU time consumed by BTM and AFM is counted on an AMD Ryzen 5 processor 3.0 GHz. In addition, commercial software (ANSYS workbench) is introduced for comparison to further verify the ability of BTM to generate high quality meshes. The mesh quality is evaluated by the criterion proposed by Lo [24]. The mesh whose quality factor is larger than 0.9 is defined as an excellent mesh, and mesh quality factors are calculated by Eq. (4) and Eq. (5).

For a triangle ABC, the quality factor α can be calculated by the following equation:

$$\alpha = 2\sqrt{3}\ \frac{\| CA \times CB \|}{\| CA \|^2 + \| AB \|^2 + \| BC \|^2} \tag{4}$$

Where *AB, BC, CA, CB* are directed vectors. The quality factor α is in the range from 0.0 to 1.0. The bigger quality factor of one element, the more regular its shape is. The quality factor of equilateral triangle is 1.0.

For a quadrilateral ABCD, it contains four triangles, such as ΔABC,ΔABD,ΔCDBandΔCDA respectively. Their quality factors are-α_1,α_2,$\alpha_3$andα_4 respectively. Assume that $\alpha_1$is the largest and the α_4 is the smallest. The$\alpha_2$is larger than α_3. The quality factor β can be defined as follows:

$$\beta = \frac{\alpha_3 \alpha_4}{\alpha_1 \alpha_2} \tag{5}$$

The quality factor β is in the range from 0.0 to 1.0. The bigger quality factor of one element, the more regular its shape is.

5.1. Periodic surface models

In this part, some periodic surfaces are taken as examples for verifying the effectiveness of BTM. The consumed CPU time is compared between BTM and AFM for similar mesh densities, and it is shown in Table 1. The mesh results of periodic surfaces meshed by AFM and BTM are shown in this section.

As can be seen from Table 1, the effectiveness of BTM is higher than AFM. This is because the faces are relatively regular, and thus the levels subdivided from the initial cells are relatively small in the implementation of BTM. Most cell edges are obtained by subdividing their parent edges, and their length is directly taken as half of the length of the parent edge. So, there is no need to calculate the length again, which saves time and improves efficiency. In addition, the time consumed by AFM for the cone is less than that of the sphere and the ring. This is because the cone owns a flat face. The Meshing results and mesh quality are shown in Figs. 20-28.

The mesh results for the periodic surfaces in U-direction are shown in Figs. 20-23. The shape of the mesh generated by BTM and AFM is regular. However, as can be seen from Fig. 24 and Fig. 25, the quality of meshes generated by BTM is better than that of meshes generated by AFM. For the sphere, the number of excellent meshes generated by AFM accounts for 73.33% of the total number of meshes in AFM. However, the percentage of excellent meshes generated by BTM is 100%. For the cone, compared with AFM, the shape of meshes generated by BTM around the tip of the cone is poor (see Fig. 23(b)), and thus the quality of these meshes is less than 0.3(see Fig. 25). But the percentage of excellent meshes generated by BTM is greater than that of AFM, as shown in Fig. 24 and Fig. 25.

Compared with the meshes result generated by AFM (see Fig. 26), the BTM can naturally obtain anisotropic meshes. Fig. 27 shows the subdivision results of the BTM for a periodic surface in UV-direction, and it can be seen from the figure that the mesh dimension in the V-direction is longer than that in the U-direction. The binary-tree subdivision generates quadrilateral meshes with regular shapes, the percentage of excellent meshes generated by BTM is 100%, as shown in Fig. 28. The results

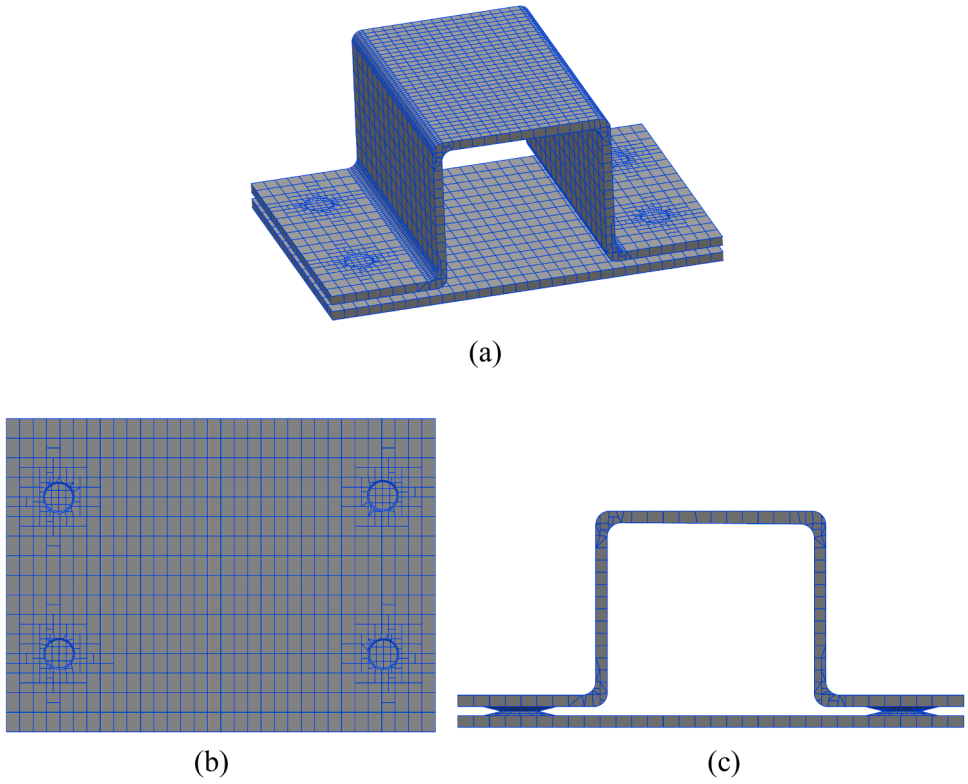


Fig. 43. Results of the spot-welding model meshed by BTM:(a) Front view; (b) Bottom view; (c) Side view.

verify that the algorithm can be applied to the periodic surfaces in UV-direction. In addition, the mesh results of the periodic surfaces divided by small mesh sizes are shown in Fig. 29 and Fig. 30.

5.2. Realistic geometric models

In this section, BTM is applied for some non-periodic surfaces, and compared with AFM to further verify the effectiveness of the proposed method. It is worth noting that the ANSYS workbench is employed to verify the practicality of the BTM. Though the software does not provide the CPU time statistics for mesh generation, the grid quality statistics of the software are complete. And mesh quality evaluation metric used by ANSYS workbench is also calculated by Eq.(4). The ANSYS workbench is referred to as ANSYS for short.

5.2.1. Example 1: three-way pipe

A three-way pipe (see Fig. 31) is meshed by BTM and AFM with different mesh sizes. The consumed CPU time is compared between the BTM and AFM for different mesh densities and shown in Tables 2-3 and Fig. 32.

As can be seen from the statistics, the CPU time consumed by AFM is significantly less than that of BTM. However, in the process of AFM, all the generated meshes rely on a so-called mesh size field constructed by Oct-tree. Thus, as shown in Fig. 32, the CPU time consumed by BTM is similar to the total time consumed by AFM and Oct-tree.

In order to compare the quality of meshes generated by AFM, BTM and ANSYS, the three-way pipe is meshed by these methods, and mesh statistics are shown in Table 4 and Fig. 33. As can be seen from Table 4, the CPU time consumed by BTM is more than that of AFM, but the CPU time consumed by BTM is less than the total running time of AFM and oct-tree. What's more, the Average quality of meshes generated by BTM is significantly higher than that of other methods. Compared with AFM, the percentage of excellent meshes is improved from 69.80% to 91.53% in result of BTM, as shown in Fig. 33. And the excellent rate of meshes meshed by ANSYS is only 49.69%. The quality of meshes generated by BTM is better than that of AFM and ANSYS. Thus, the effectiveness of the proposed method is clearly demonstrated. The corresponding meshes results are shown in Figs. 34-36.

5.2.2. Example 2: cup

Fig. 37 shows a cup model, and mesh results (BTM, AFM and ANSYS) are shown in Figs. 38-40. As can be seen from Fig. 37, at the arc chamfer of the connection between the cup handle and the cup body, the surface curvature is relatively large. Therefore, the distribution density of the grid is significantly greater than that of the grid inside the cup handle and the cup body. In order to ensure the calculation accuracy, it is necessary to distribute relatively dense meshes at the place where the physical field parameters or the geometric curvature changes greatly. However, the performance of AFM and ANSYS is not so excellent as BTM, as shown in Fig. 38-40.

The consumed CPU time and mesh quality of BTM and AFM are shown in Table 5 and Fig. 41. As can be seen from the statistics, the running time required by BTM is a little more than the total running time consumed by AFM and Octree. However, the quality average of meshes generated by BTM is better than that of AFM and ANSYS. The percentage of excellent grids generated by AFM and BTM are 96.51% and 98.40%, respectively. And the excellent rate of meshes meshed by ANSYS is only53.88%.

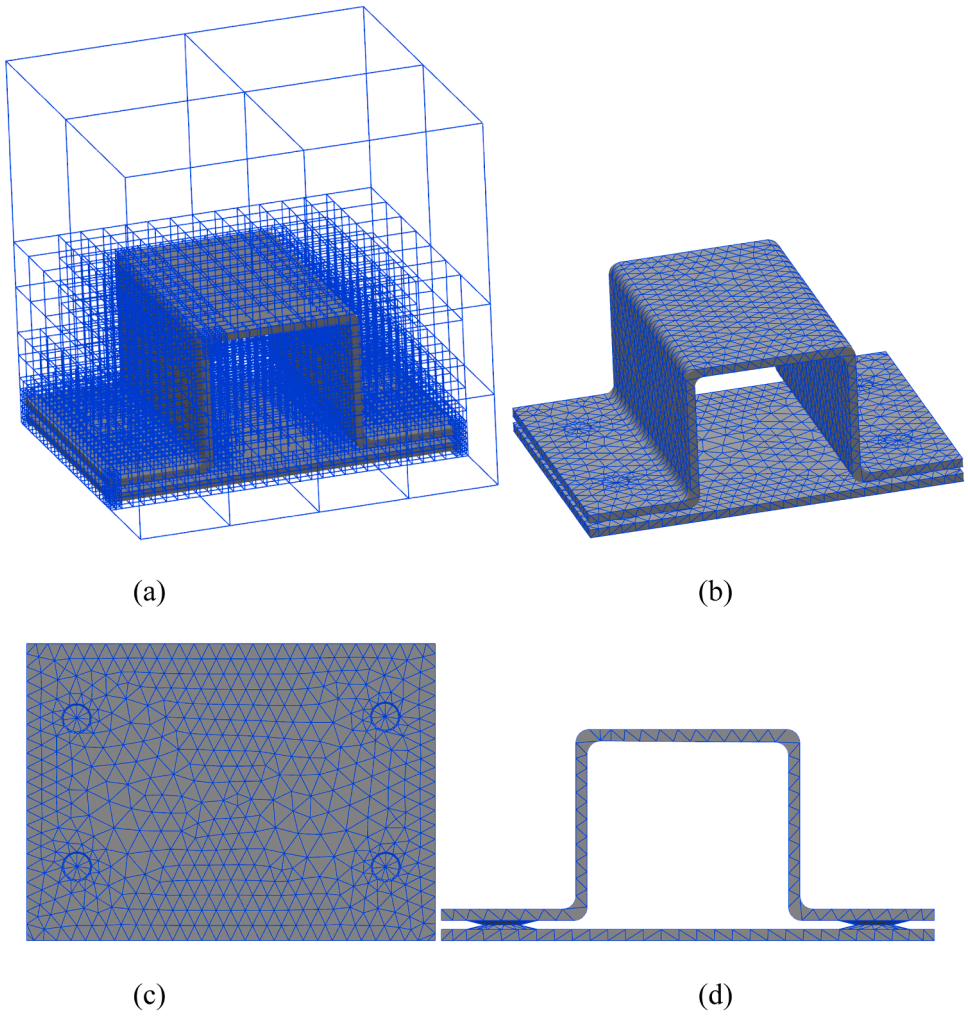


Fig. 44. Results of the spot-welding model meshed by AFM: (a) Sizing field constructed by octree refinement; (b) Front view; (c) Bottom view; (d) Side view.

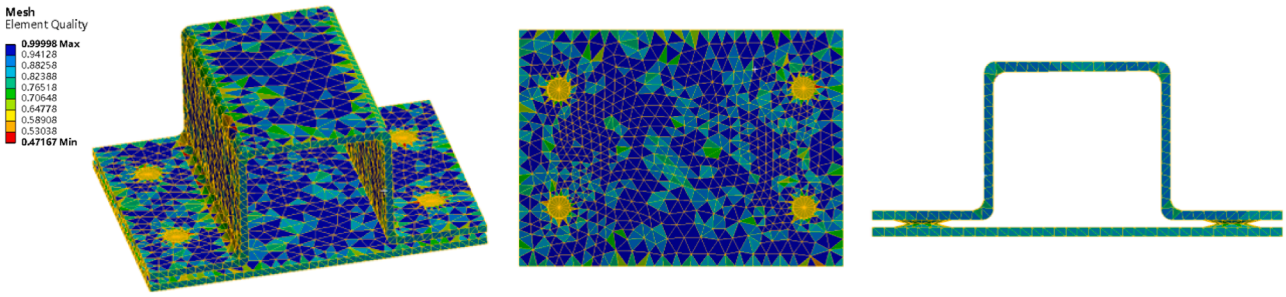


Fig. 45. Results of the spot-welding model meshed by ANSYS workbench.

Table 6
Mesh statistic for the spot-welding model.

Method		Time (s)	Mesh Number	Average value of mesh quality
Total Process of AFM	Oct AFM	0.516 0.078	6334	0.934
BTM		0.719	6392	0.990
ANSYS workbench			6404	0.868

5.2.3. Example 3: spot-welding model

Fig. 42 shows a spot-welding model which owns slender surfaces. This model is employed to demonstrate the ability of BTM to mesh thin structures. As can be seen from Fig. 43, the BTM is able to generate quadrilateral cells with regular shapes on the thin surface of the spot-welding model.

Mesh distribution affects the simulation accuracy in regions where the physical field parameters or the geometric curvature changes greatly. Compared with the meshes result of AFM (see Fig. 44) and ANSYS (see Fig. 45), the meshes generated by the BTM are more densely distributed around the welding spot and on fillet surfaces, while the mesh distribution in other areas is sparse.

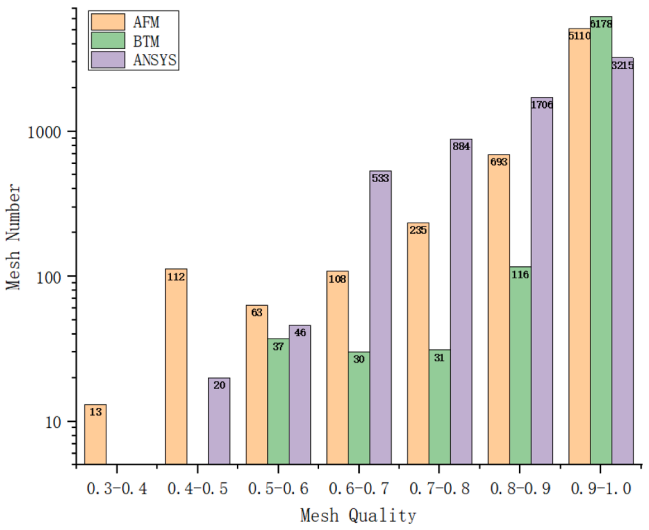


Fig. 46. Mesh quality statistics for spot-welding model.

The consumed CPU time and mesh quality of BTM, AFM and ANSYS are shown in Table 6 and Fig. 46. As can be seen from the statistics, the running time required by BTM is a little more than the total running time consumed by AFM and Octree. However, the quality of meshes generated by BTM is significantly higher than that of AFM. The average quality of meshes generated by ANSYS is lower than that of AFM and BTM. Moreover, the percentage of excellent grids generated by AFM, BTM and ANSYS is 80.68%, 96.65% and 50.20%, respectively.

5.2.4. *Example 4: relative complex model*

The above three examples demonstrate the effectiveness of BTM. In this section, some relatively complex models are meshed by BTM to verify the robustness of BTM. Meshes results are shown in Fig. 47. In addition, more complex models will be the focus of future work.

5.3. *"dirty" model*

The"dirty"models exist geometric noise, such as extremely short edges(Fig. 48(a)) and narrow faces(Fig. 49(a)). When the boundary of a face has an extremely short edge, the subdivision process of the face is the same as that of other faces without an extremely short edge. It is worth noting that, at the fourth step of BTM (Fig. 1(d)), the boundary cell, which contains the extremely short edge, will not be subdivided, as

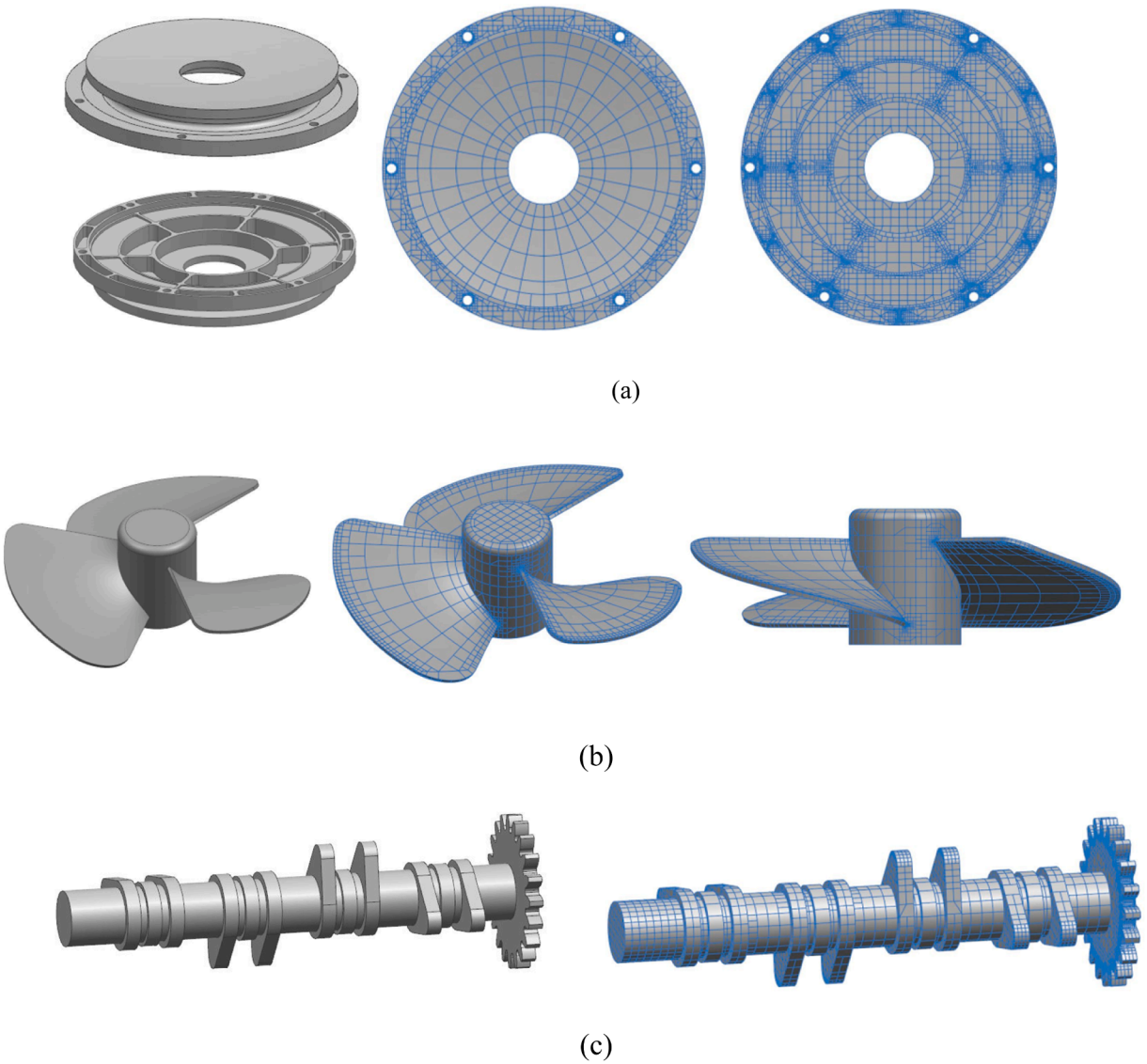


Fig. 47. Relatively complex models meshed by BTM:(a) End cap; (b)Fan; (c) Cam shaft.

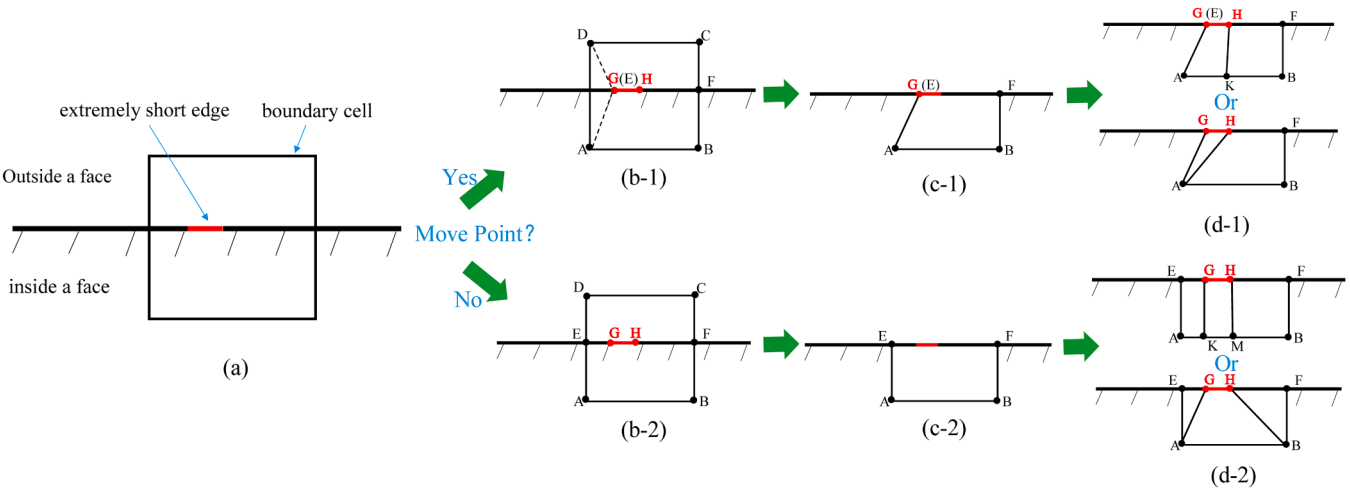


Fig. 48. BTM process for extremely short edges.

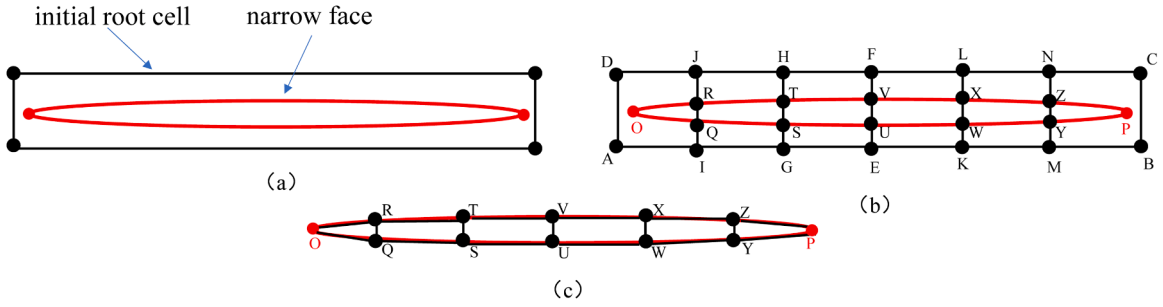


Fig. 49. BTM process for extremely narrow face.

shown in Fig. 48. If the endpoints of the extremely short edge are closer to the cell edge, the operation of moving points is performed, as shown in Fig. 48(b-1). Then the boundary cell can be treated by the template shown in Fig. 10(c), and the results are shown in Fig. 48 (c-1, c-2). The Cell ABEF can be treated by the extremely short edge shown in Fig. 48 (d-1, d-2).

If a “dirty” model has an extremely narrow face, the process of mesh generation for the narrow face can be performed step by step according to the BTM process shown in Fig. 1. For example, the initial root cell of the face is shown in Fig. 48(a). Fig. 48(b) shows the results before treating the boundary cell. The boundary cell can be treated by the templates shown in Fig. 10(a). The final mesh results are shown in Fig. 48(c).

As shown in Fig. 50(a), the model has very short edges, which are highlighted in yellow in the enlarged view. For the currently popular mesh methods (AFM, DTM, etc.), the geometric model with geometric "noise" requires geometric repair (merging edges, merging faces, etc.), otherwise it is difficult to mesh this model. Moreover, the quality of the generated grid is poor, and the local grid is too dense, which is not conducive to subsequent simulation and analysis process. Fig. 50(b) shows the meshing results of the AFM on a cube with geometric "noise". It can be seen that the mesh generated by this method is very dense on a local area of the "dirty" geometry model without geometric repair. However, the BTM allows for the existence of hanging points, making the meshing more flexible and freer. Fig. 50(c) shows the meshes result generated by the BTM. The BTM is convenient to mesh the "dirty" geometry model. In addition, a weld bead with an extremely short edge (see Fig. 51(a)) is meshed by BTM, as shown in Fig. 51(b). Complex geometric models with geometric "noise" will be the focus of future work.

5.4. DiBFM based on BTM

The meshes generated by the binary-tree method (BTM) have been applied for the dual interpolation boundary face method (DiBFM) to solve the 3D potential problems [18] and elastic problems [20]. In the reference [18], a steady-state heat conduction problem of a goblet is solved by the DiBFM. The goblet is discretized with binary tree grids shown in Fig. 52(a). Fig. 52(b) and Fig. 52(c) show simulation results calculated by the DiBFM and FEM, respectively. It can be seen that the contours given by our method using fewer nodes are consistent with FEM using much more nodes. Their specific details can be found in reference [18], and demonstrate the BTM can be effectively applied in numerical simulations.

6. Conclusions

This paper proposes an adaptive binary-tree mesh generation method, which automatically generates discontinuous meshes. Since the existence of overhanging points is allowed, the connection relationship of adjacent meshes is more flexible and more suitable for the automatic meshing of arbitrary complex models. Numerical examples illustrate

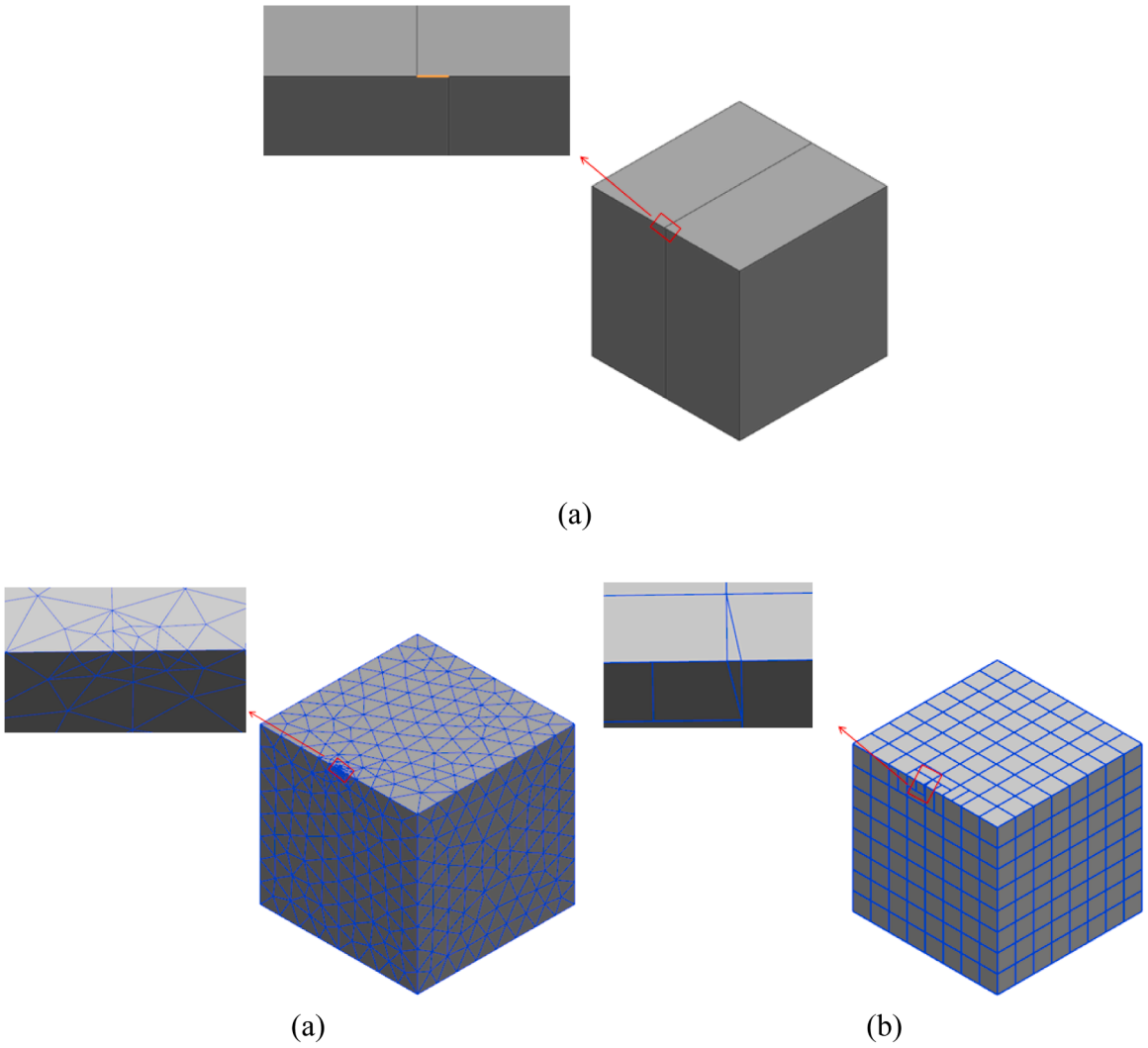


Fig. 50. Cube model with geometric "noise": (a) geometric "noise"; (b)Mesh results of AFM; (c) Mesh results of BTM.

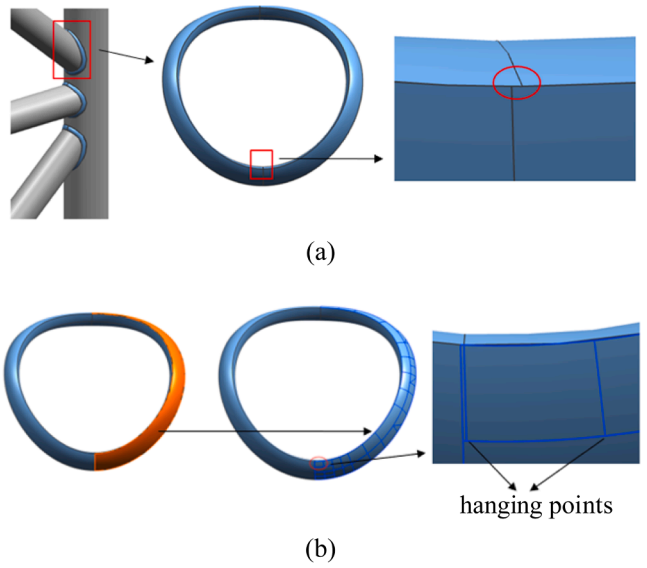


Fig. 51. Steel frame weld model:(a) a weld bead with extremely short edge; (b) mesh results.

that the BTM generates mostly rectangular cells with regular shapes on various geometric surfaces. Moreover, the method can flexibly control the mesh size distribution without a sizing field. It owns the ability to generate anisotropic meshes and mesh periodic surfaces. Compared with the AFM process, the running time required by BTM is a little more than the total running time consumed by AFM and Octree, but the quality of meshes generated by BTM is significantly higher than that of AFM. In addition, compared with commercial software (ANSYS workbench), the mesh quality obtained by BTM is better than that obtained by ANSYS workbench. Coupled with the dual interpolation boundary face method, the BTM has been applied to the process of solving 3D problems. These results verify the effectivity and practicality of the proposed method. The application of this method for meshing complex CAD models with geometric "noise" will be a major task 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. Zhang Jianming reports financial support was provided by National Natural Science Foundation of China. Ju Chuanming reports financial support was provided by Shandong Provincial Natural Science Foundation of China. Chi Baotao reports financial support was provided by China Postdoctoral Science Foundation.

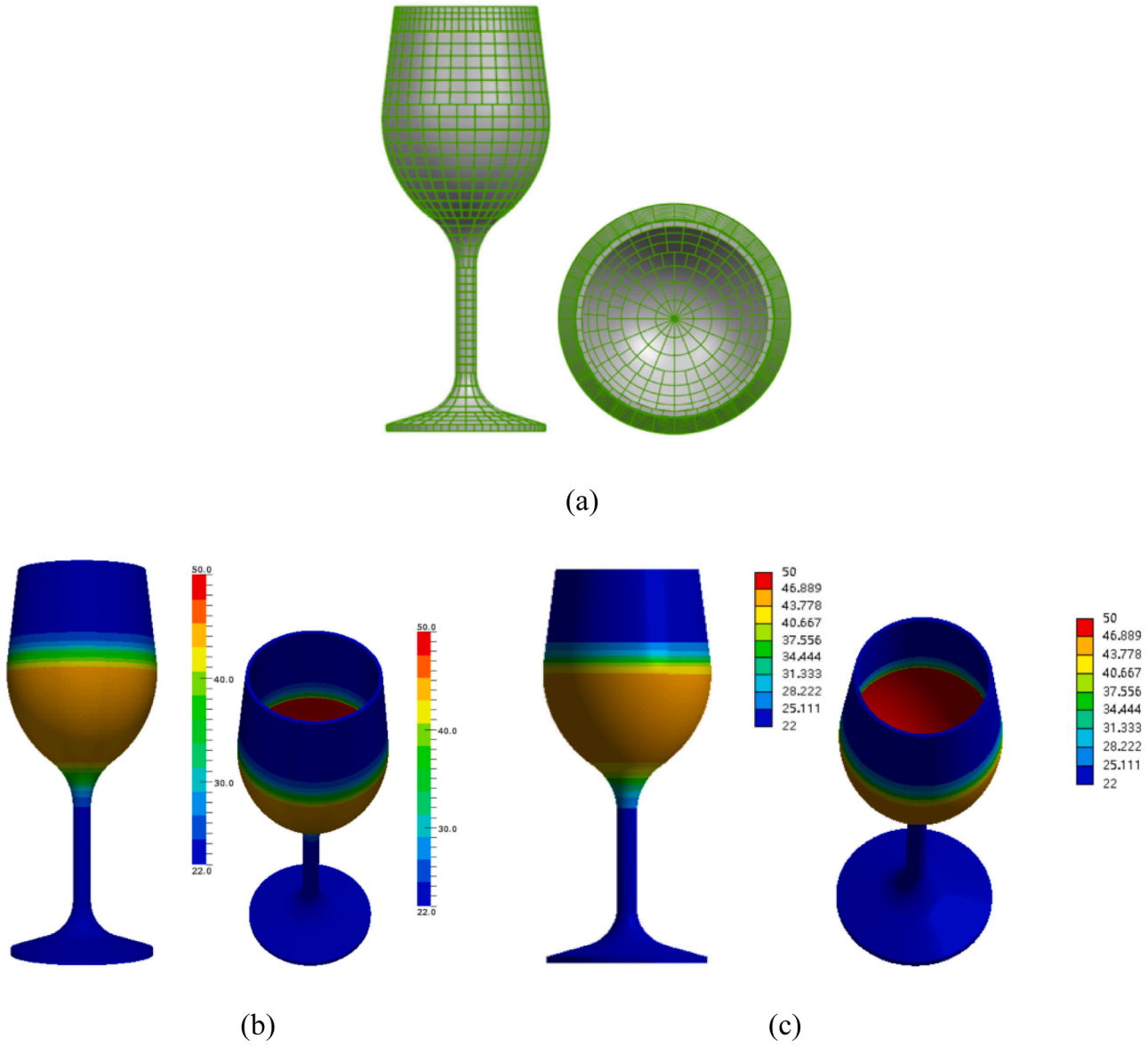


Fig. 52. Simulation results: (a)mesh results of BTM; (b)Contours of temperature: DiBFM with 1880 source nodes; (c)Contours of temperature: FEM with 222,335 nodes.

Data availability

The data that has been used is confidential.

Acknowledgments

This work was support by National Natural Science Foundation of China (11772125 and 11972010), Shandong Provincial Natural Science Foundation of China (ZR2022QA072), China Postdoctoral Science Foundation (2021M702024).

References

[1] Frey PJ, George PL. Mesh generation: application to finite elements. London: Iste Ltd; 2008.
[2] Thompson JF, Soni BK, Weatherill NP. Handbook of grid generation. CRC press; 1998.
[3] Tristano JR, Owen SJ, Canann SA. Advancing Front surface mesh generation in parametric space using a riemannian surface definition. IMR 1998:429–45.
[4] Borouchaki H, Patrick L. Parametric surface meshing using a combined advancing-front generalized Delaunay approach. Int J Numer Methods Eng 2015;49(1–2):233–59.
[5] Yu K, Chen J, Fu K, He J, Zheng J, Zheng Y. On the efficiency of the advancing-front surface mesh generation algorithm. Comput-Aided Design 2022;153:103403.
[6] Foucault G, Cuillière JC, François V, Léon JC, Maranzana R. Generalizing the advancing front method to composite surfaces in the context of meshing constraints topology. Comput-Aided Design 2013;45(11):1408–25.
[7] Baehmann PL, Wittchen SL, Shephard MS, Grice KR, Yerry MA. Robust, geometrically based, automatic two-dimensional mesh generation. Int J Numer Methods Eng 1987;24(6):1043–78.
[8] Fang TP, Piegl LA. Delaunay triangulation using a uniform grid. Comput Graph Appl 1993;13(3):36–47.
[9] Borouchaki H, George PL, Mohammadi B. Delaunay mesh generation governed by metric specifications. Part I. Algorithms. Finite Elem Anal Des 1997;25(1–2):85–109.
[10] Du Q, Wang D. Recent progress in robust and quality Delaunay mesh generation. J Comput Appl Math 2006;195(1–2):8–23.
[11] Baldwin KH, Schreyer HL. Automatic generation of quadrilateral elements by conformal mapping. Eng Comput 1985;2(3):187–94.
[12] Ruiz-Gironés E, Sarrate J. Generation of structured meshes in multiply connected surfaces using submapping. Adv Eng Softw 2010;41(2):379–87.
[13] Yerry MA, Shephard MS. A modified quadtree approach to finite element mesh generation. IEEE Comput Graph Appl 1983;3(1):39–46.
[14] Chen J, Xiao Z, Zheng Y. Automatic sizing functions for unstructured surface mesh generation. Int J Numer Methods Eng 2017;109(4):577–608.

[15] Xiao Z, He S, Xu G, Chen J, Wu Q. A boundary element-based automatic domain partitioning approach for semi-structured quad mesh generation. Eng Anal Bound Elem 2020;113:133–44.
[16] Chong CS, Kumar AS, Lee HP. Automatic mesh-healing technique for model repair and finite element model generation. Finite Elem Anal Design 2007;43(15):1109–19.
[17] Zhang J, Lin W, Dong Y, Ju C. A double-layer interpolation method for implementation of BEM analysis of problems in potential theory. Appl Math Model 2017;51:250–69.
[18] Zhang J, Xiao R, Wen P, Ju C, Lin W, He R. Dual interpolation boundary face method for 3-D potential problem based on binary tree grids. Comput Methods Appl Mech Eng 2022;390:114432.
[19] Zhang J, Ju C, Wen P, Shu X, Lin W, Chi B. A dual interpolation boundary face method for 3D elasticity. Eng Anal Bound Elem 2021;122:102–16.
[20] Zhang J, Chai P, He R, Lin W, Ju C, Chi B. Implementation of a dual interpolation boundary face method by discontinuous meshes. Eng Anal Bound Elem 2022;139:152–68.
[21] De Miguel Barcena R. Curvilinear mesh generation for CAD models involving trimmed, periodic and degenerated surfaces. 2020. http://hdl.handle.net/10251/148714.
[22] Guan Z, Shan J, Yao Z, Gu Y. An extended advancing front technique for closed surfaces mesh generation. Int J Numer Methods Eng 2010;74(4):642–67.
[23] Guan Z, Sui X, Gu Y, Li Y. Automatic finite element mesh generation over 3D combined surfaces. Chin J Comput Mech 2003;20(4):409–16.
[24] Lee C, Lo S. A new scheme for the generation of a graded quadrilateral mesh. Comput Struct 1994;52(5):847–57.