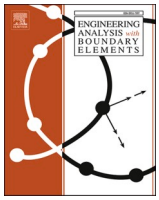
Engineering Analysis with Boundary Elements 166 (2024) 105846

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

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



A nonconforming surface mesh generation method by binary tree

Jianming Zhang [a,*], Chong Zhang [a], Rongxiong Xiao [a], Baotao Chi [b]

[a] State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China

[b] School of Mechanical Engineering, Shandong University of Technology, Zibo 255000, China

ARTICLE INFO

Keywords:
Automatic surface subdivision
Binary tree
Nonconforming mesh
Geometry repair

ABSTRACT

Computer-aided engineering (CAE) has emerged as an indispensable tool for facilitating engineering practices and driving industrial innovation. However, the insufficient quality and efficiency of discretizing complex computer-aided design (CAD) models significantly impede the advancement of CAE calculation accuracy and automation. The presence of "dirty" geometry leads to the fact that it is almost impossible to generate a traditional conforming mesh without geometry repair. In most instances, automating geometry repair proves to be even more challenging than meshing. To cope with intricate CAD model with "dirty" geometry, a novel binary tree surface subdivision method (BSSM) is proposed for automatically generated conforming and nonconforming meshes directly on CAD models. In contrast to the conventional conforming mesh, the nonconforming mesh allows for hanging nodes, thereby eliminating the limitations of the mesh conformance. This facilitates rapid transitions in mesh size for automatic mesh generation when dealing with "dirty" geometry. A series of numerical models employing BSSM are presented in this study. Results reveal that BSSM can automatically, efficiently and reliably generate high level quality mesh for arbitrary structures.

1. Introduction

Modern industrial design and manufacturing requires the utilization of CAE analysis to address a wide range of fields of issues, including structure, fluid, heat transfer, electro-magnetic, acoustics, vibration aspects, and more. The expanding scope of engineering applications mandates that CAE confronts nonlinearity, multiphysics, as well as large-scale and multi-scale problems, leading to a substantial escalation in the computational time and space demands [1]. In contrast to the CAE solution and post-processing phases, the pre-processing stage falls short of achieving complete automation due to the absence of systematic theoretical support. This deficiency considerable impact speed and accuracy of analysis and solution when confronted with intricate problems. Being a critical procedure of pre-processing, mesh generation has emerged as a primary research focal point for both CAE software developers and researchers [2].

Surface mesh is a main research focus in CAE mesh division and computer graphics. Of all current mesh generation algorithms, advancing front technique (AFT) [3–8] and Delaunay triangulation (DT) [9–11] are widely recognized as the most mature methods. While AFT exhibits strong adaptability to geometric boundaries and ability to produce high-quality mesh, its efficiency and stability are hindered by the heavy use of judgment statements and distance calculations. DT benefits from a mature mathematical theory and superior algorithmic efficiency, but it is significantly influenced by the surface shape and mesh mapping process. Moreover, the tree-structure [12–14] reduce the complexity of automatic mesh division. However, it yields the inferior meshes near the surface boundary and the self-intersections inevitable occurrence, seriously affecting the calculation exactitude. Generally, the aforementioned algorithms necessitate initial input CAD model [15] without "dirty" geometry, such as short edges, narrow surfaces and sharp angles. Otherwise, the solid model must undergo geometry repair [16–18]. Unfortunately, achieving satisfactory computational accuracy through geometry repair typically involves extensive manual operation [1], which is not able to desire the demands posed by complex engineering problems.

Besides geometry repair, various methods including virtual topology [6], three-dimensional (3D) reconstruction [19,20], parallel aligned surface meshing [21] and idealization operators [22] have been implemented to mitigate or avoid the negative effects of "dirty" geometry on mesh division. Moreover, Liu et al. propose a novel unified framework for rapidly generating high-quality mesh on dirty composite

* Corresponding author.
E-mail address: zhangjm@hnu.edu.cn (J. Zhang).

https://doi.org/10.1016/j.enganabound.2024.105846
Received 19 March 2024; Received in revised form 9 June 2024; Accepted 11 June 2024
Available online 22 June 2024

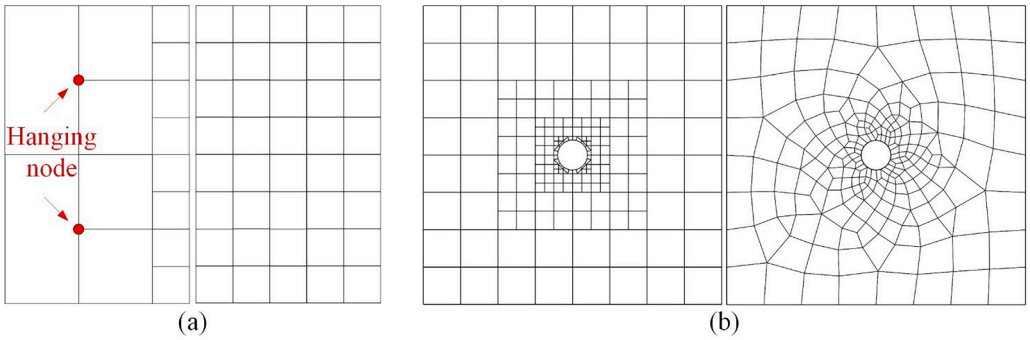


Fig. 1. Mesh transition with equal size of nonconforming and conforming (a) rectangular mesh, (b) quadrilateral-dominated mesh.

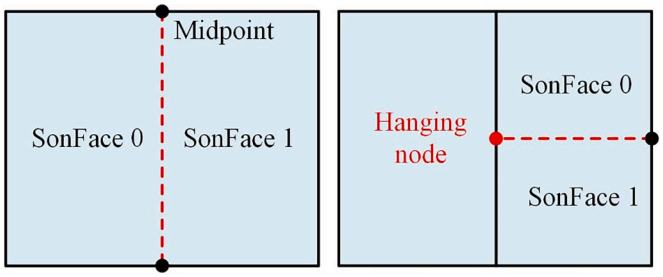


Fig. 2. Binary-tree subdivision method.

surfaces [23], which has been successfully utilized with CAD models of industry-level complexity. Nevertheless, these methods modify the original CAD model, bring about the loss of part essential geometric or topological information in the mesh result. Therefore, current methods are still incapable to realize automatic high-quality discretization of complex structures with "dirty" geometry. In this paper, BSSM is proposed to fully restoring the initial CAD model by discrete mesh results. Contrary to traditional conforming mesh, the nonconforming mesh incorporates hanging nodes (Fig. 1), where the conforming mesh is generated by business software ANSYS Workbench (ANSYS). Obviously, nonconforming mesh can allow for flexible control of mesh density and regular dimensional transitions. Furthermore, it facilitates parallel

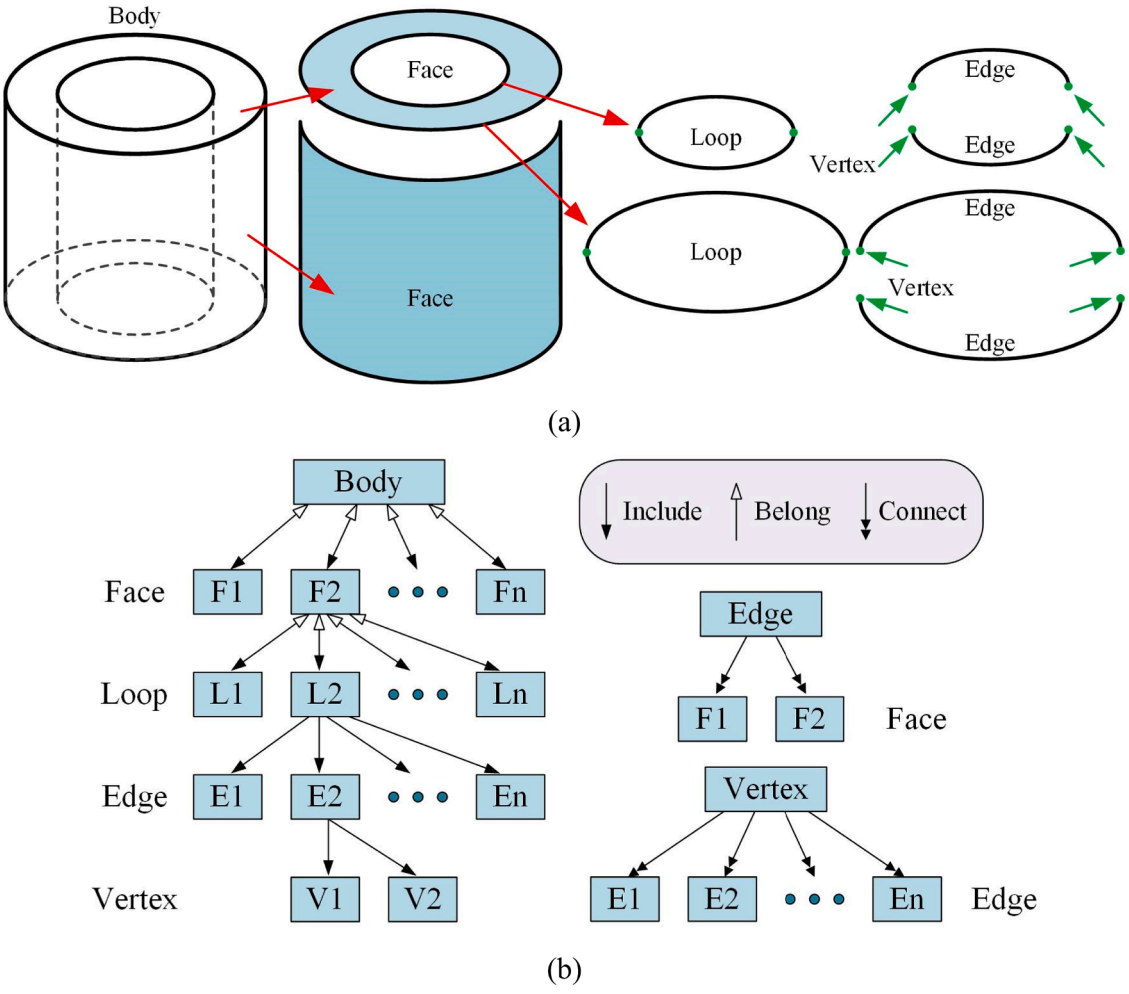


Fig. 3. Topological structure for a 3D geometry model.

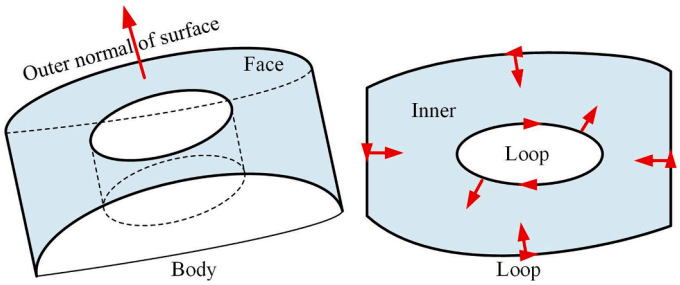


Fig. 4. Loop orientation, inner and outer of the surface.

dividing for nonconforming mesh. The data structure for parallel computing is unificated, eliminating unnecessary matching between different regional meshes and allowing complete independency in each regional mesh division. On the other hand, the binary tree, as an important type of tree structure, exhibits a low vacancy rate and high space utilization rate compared to the quadtree algorithm. The binary tree subdivision process is shown in Fig. 2. During subdivision, the midpoints of the opposite cell edge (single mesh) are connected to divide a cell into two parts. Therefore, the binary tree subdivision method is more suitable for generating anisotropic mesh and offers easier control to local mesh refinement. BSSM has been applied to the boundary element method (BEM) for solving solid mechanics problems such as elasticity, heat transfer, acoustics, etc. Moreover, BSSM is able to serve as an intermediary process in numerical methods such as finite element method (FEM) and finite volume method (FVM).

Building upon previous work by our research group [24–28], this paper extensively exploits topological information of CAD model to enhance the relevant algorithms for instance boundary restoration, "dirty" geometry processing, etc. The rest of this paper is organized as follows: Firstly, the boundary representation (B-Rep) structure necessary for mesh generation is introduced. Then, the nonconforming mesh discrete method based on BSSM is described in detail for a simple 3D CAD model. Finally, a series of numerical results are given to illustrate the advantages and generality of the BSSM.

2. Preparing topological structure for mesh generation

The mesh quality determines the accuracy, efficiency, and authenticity of numerical calculation. An optimal mesh should closely align with the geometric domain within the tolerance range, necessitating precise capture the needful information for mesh generation. B-Rep, which utilizes surfaces to represent 3D geometry, is the predominant method for presenting CAD models, providing superior control and error exactitude. Fig. 3(a) illustrates the hierarchical structure of the 3D geometry, progressing from "Body" → "Face" → "Loop" → "Edge" → "Vertex", with each level encompassing the subsequent level. Among these elements, a Body is an independent structure surrounded by multiple Faces, a Face is formed by multiple Loops of negligible thickness, a Loop consists of Edges arranged in sequential order. The Vertex serves as the fundamental element of B-Rep, an Edge represents a curve connecting two Vertices. Conversely, each Face belongs to a single Body, and each Loop belongs to a single Face. Furthermore, the connection relationship between Face and Edge is such that each Edge connects two Faces, and each Vertex can connect several Edges. The overall topology relationship among the B-Rep elements is shown in Fig. 3(b), with distinct styles of arrows denoting various topology relationships.

Aside from the B-Rep topology, the orientation of Face and Loop must also be specified to ensure that the left side of all Loops corresponds to the interior of Face (see Fig. 4). The normal direction of a Face is defined as from the outward pointing to inward of Body, as well as the outermost Loop of the Face follows the right-hand rule. Accordingly, the outer Loop is oriented counterclockwise, while all other Loops are oriented clockwise.

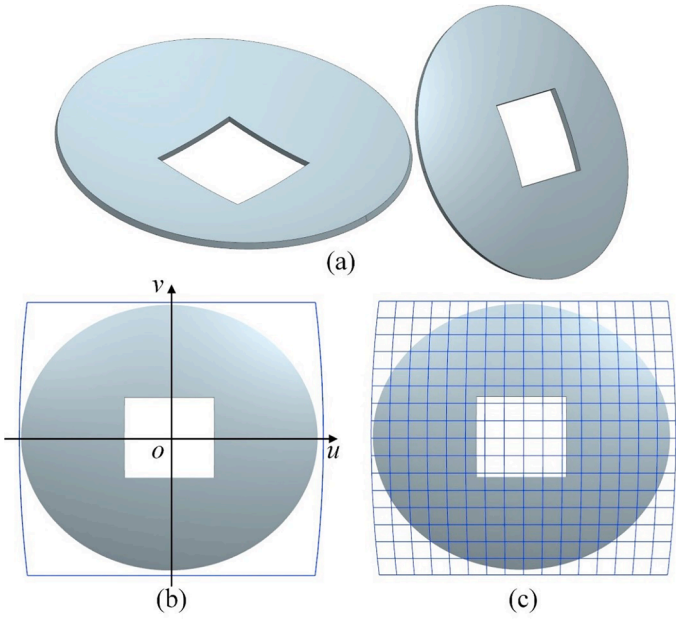


Fig. 5. (a) Example of a 3D CAD model, (b) root cell, (c) result for root cell subdivide.

3. Outline of BSSM algorithm

BSSM is briefly outlined in the following steps:

Step 1. Construct background mesh
Step 1.1 Projection of the 3D surface onto the 2D parameter space and subdivision based on curvature (Fig. 5).
Step 1.2 Discretize the Loop into Segments according to the curvature of the geometric Edges (Fig. 7(a)).
Step 1.3 Calculate the intersection points of the background mesh with the Loop (Fig. 7(b)).
Step 1.4 Refinement of the local mesh, ensuring that a maximum of one geometry vertex and one complete Segment between two intersections (Fig. 10(b)).
Step 1.5 Balancing the background mesh size, reducing the ratio of neighboring cell sizes, and ensuring at most one hanging point on each cell edge (Fig. 10(c)).
Step 2. Constrained boundary recovery
Step 2.1 Categorization of all cells into rim cells, inner cells, and outer cells (Fig. 13(d)).
Step 2.2 Set the degradation points based on the distance from the inner cell nodes to the Loop on the rim cells (Fig. 14). Degenerate points will be eliminated during the boundary recovery process.
Step 2.3 Categorized into the rim cells into common boundary templates, adhere boundary templates and concave boundary templates for boundary recovery according to the number and location of degradation points (Fig. 19).

4. Construct background mesh base on binary tree

The BSSM is performed within the parametric space of entire Face, requiring the mapping of 3D parametric surfaces to 2D faces. Eq. (1) demonstrates the parameter mapping method for surfaces. In this section, we employ the CAD model depicted in Fig. 5(a) as an explanatory example.

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

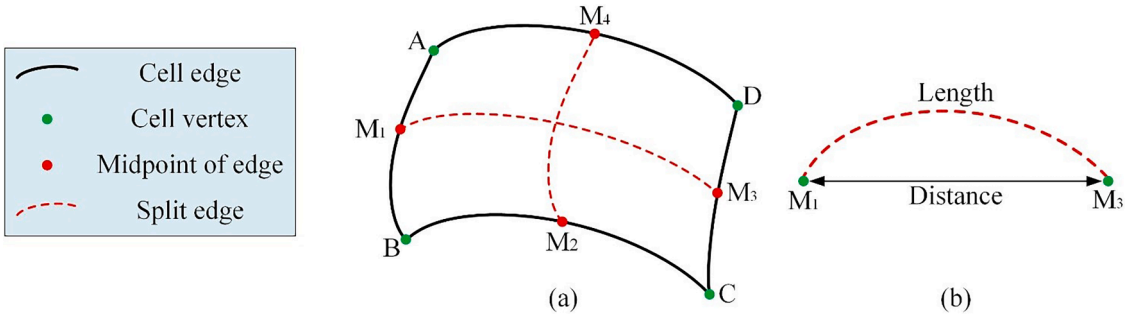


Fig. 6. Curvature calculation, (a) cell, (b) edge.

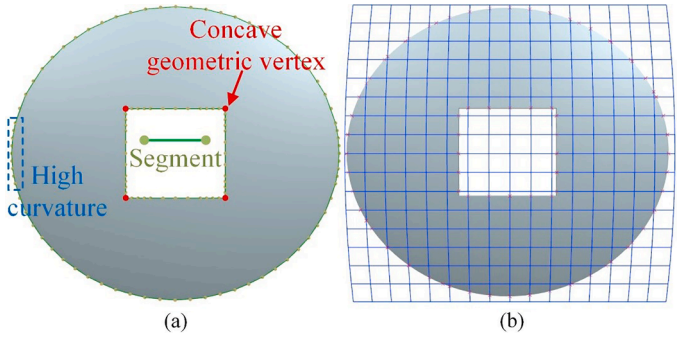


Fig. 7. (a) Result for loop subdivide, (b) result for intersection points calculation.

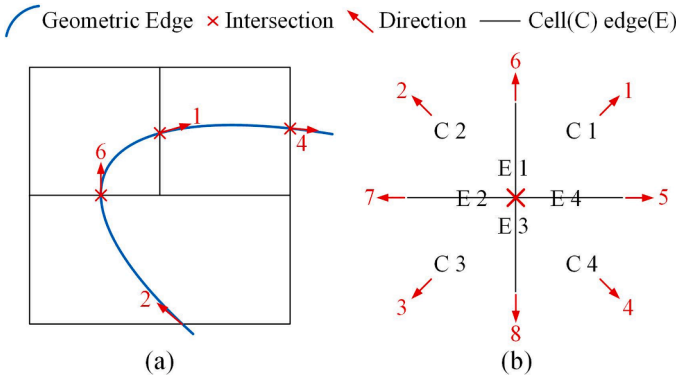


Fig. 8. (a) Intersection location and tangent direction, (b) cell and cell edge location.

4.1. Initial background mesh

4.1.1. Root cell and subdivide

The initial root cell is the smallest rectangular box that covers the face parameter domain in 2D parameter face. Fig. 5(b) displays the range of this box denoted as $u_0 \le u \le u_1, v_0 \le v \le v_1$. Initially, the root cell is subdivided based on the mesh size, which is defined in this study as 1/24 of the maximum length projected by the CAD model onto 3D Cartesian coordinates. Subsequently, the mesh is further subdivided according to the curvature of cell. The curvature is determined by the length of the split edge (Fig. 6) and the distance between the endpoints of the split edge. The curvature of cell (*Ratio*) can be calculated by

$$Ratio = \frac{l_{\overset{\frown}{M_1M_3}} - d_{\overline{M_1M_3}}}{d_{\overline{M_1M_3}}}, \tag{2}$$

where $l_{\overset{\frown}{M_1M_3}}$ is the length of split edge of point M_1 to M_3, and $d_{\overline{M_1M_3}}$ and distance of point M_1 to M_3 in Fig. 6(b), respectively. Subdivision is required when the curvature of cell exceeds the predetermined threshold value (default value in this study is 0.0125). The resulting subdivisions of root cell are shown in Fig. 5(c).

4.1.2. Subdivide loop

The loops of faces are subdivided into segments based on the curvature of the geometric edges for local mesh refinement. The subdivision criterion is identical to that described in Section 4.1.1. As depicted in Fig. 7(a), a segment is present between every pair of green points and geometric vertex obligatory serving as the endpoint of the segment. Evidently, segments are denser in areas with higher curvature. The region surrounding the concave geometric vertex (indicated in Fig. 7(a)) is also encrypted to facilitate handling of the concave boundary template. Moreover, it is crucial to note that the difference in length of consecutive segments should not exceed double. To deal with "dirty" geometry, the tiny geometric edges (1/23 length of the connected geometric edges) are exempt from this condition.

4.1.3. Calculating the intersection points between loop and mesh

Fig. 8(a) displays the intersections that connect the background mesh to loop. The calculation of intersections coordinate emerges as the most time-consuming procedure in Section 3 when the geometric boundary is composed of complex spline curves. Therefore, reducing the frequency of intersections calculation significantly improves calculation efficiency. For this purpose, the position of intersections on the background mesh and the tangent direction of intersections on the loop are recorded (Fig. 8(a). In the parameter space, all cell edges of background mesh are oriented in the positive direction. The tangent lines are categorized into eight directions according to the quadrant in the 2D Cartesian coordinate system. The numbers adjacent to the intersections in Fig. 8(a) indicate the sequence number of directions in Fig. 8(b). Fig. 8(b) illustrates the position of the cell (C) and the cell edge (E) within the coordinate system. It is sufficient to calculate the intersection of the cell corresponding to the tangent direction of current intersection with the loop to determine the location of subsequent intersection. This approach effectively diminishes the number of intersection calculations. The result of the intersections calculation for the example are displayed in Fig. 7(b).

4.1.4. Local mesh refine in inflection feature

To achieve accurate calculation in non-smooth locations, local mesh refinement is imperative for areas within the loop that exhibit significant curvature and numerous geometric vertices. The mesh requires refinement in two scenarios (Fig. 9(a)): (1) There are two or more geometric vertices between two intersections (geometric vertices connected by a tiny edge in loop are excluded). (2) There is a complete segment between two intersections (except segment within tiny edge). In addition, concave geometric vertices frequently lead to numerical computation singularities, necessitating further refinement, as depicted in Fig. 9(b). The results of the local mesh refinement are illustrated in Fig. 10(a) and (b). Following each subdivision, new intersections are inserted to loop to guarantee that all locations where loop crosses the

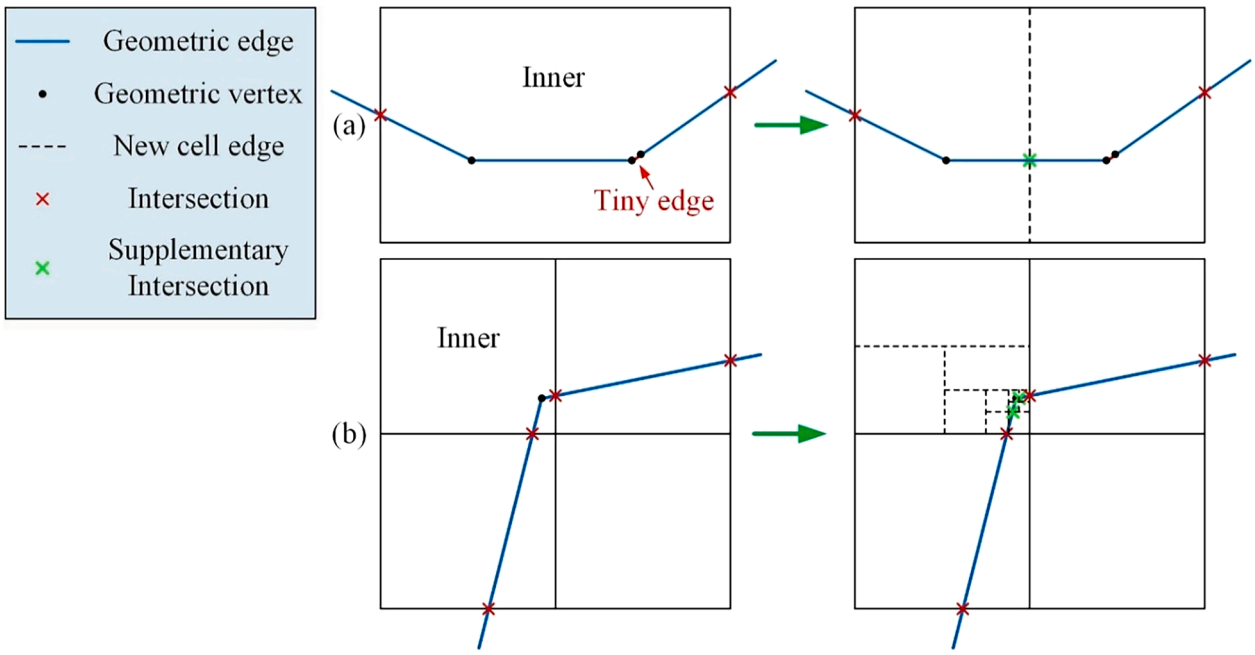


Fig. 9. (a) Refine by geometric vertices or segments, (b) Refine by concave geometric vertex.

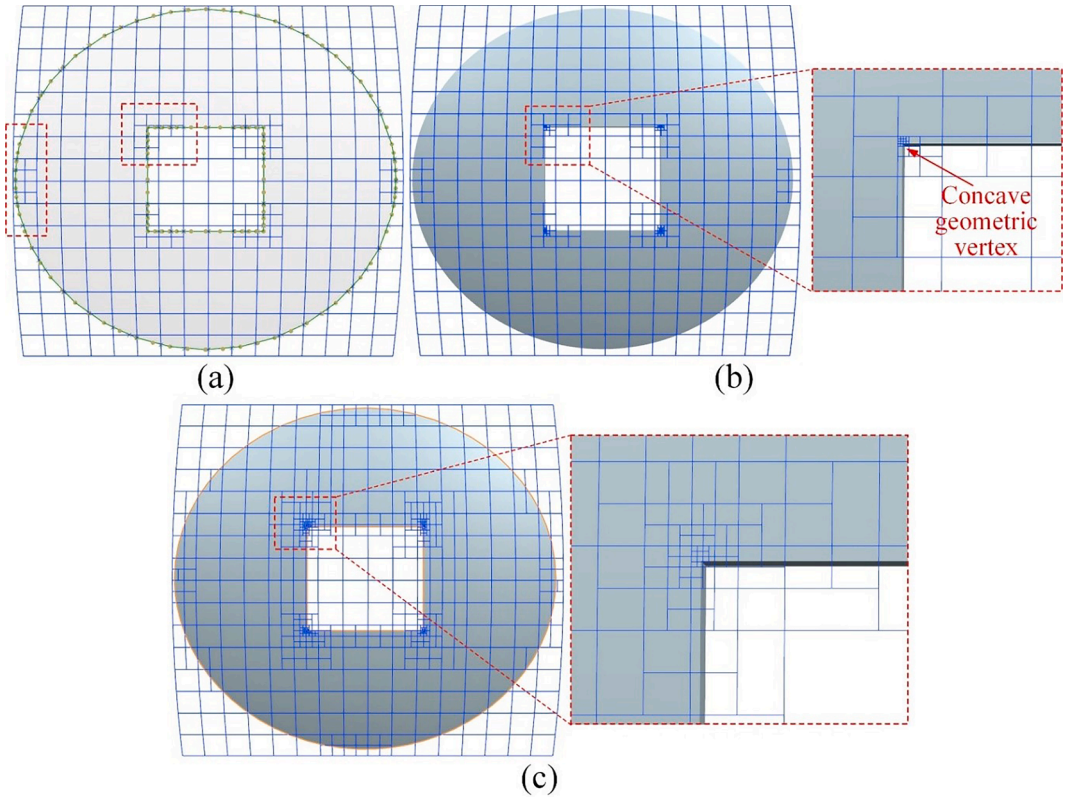


Fig. 10. Refine result for example (a) curvature refine, (b) concave vertex refine, (c) balancing background mesh result.

background mesh have intersection.

4.2. Balancing background mesh

Excessive adjacent mesh size or an unfavorable aspect ratio of the cell can significantly diminish the precision of numerical calculations. In order to facilitate a seamless transition in mesh size, it is crucial to balancing mesh size through mesh subdivision.

4.2.1. Mesh balanced in adjacent face

Within the topological structure, each geometric edge connects two faces, enabling the identification of adjacent cells on neighboring faces through edges. Mesh balanced in adjacent face requires that the ratio of tangential length and normal lengths of neighboring cells are both less

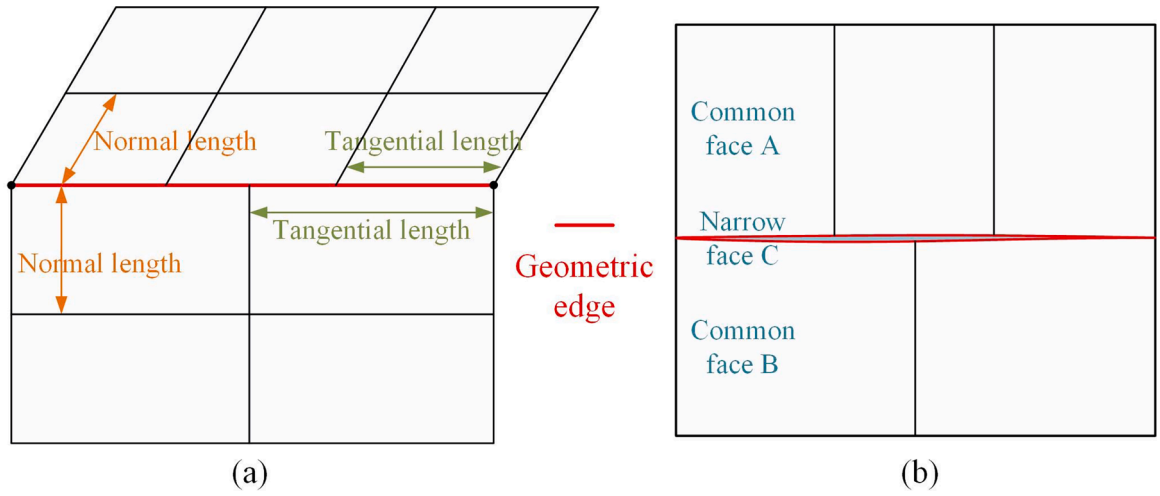


Fig. 11. (a) Normal and tangential length, (b) balance narrow face.

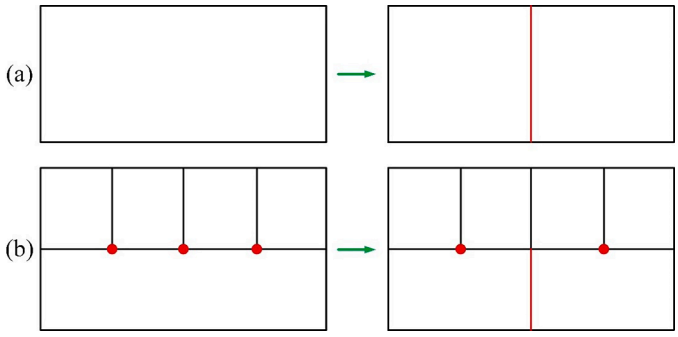


Fig. 12. (a) Balance cell aspect ratio, (b) eliminate excess hanging nodes.

than 2, as depicted in Fig. 11(a). In the case of one face being excessively narrow (with an aspect ratio greater than 50), it is essential to balance solely the tangential length of the cells to prevent excessive subdivision of adjacent faces, while still permitting the normal length to traverse the narrow face. Specifically, when face A or face B aligns with a narrow face C in Fig. 11(b), it becomes imperative to balance face A and B along the normal direction.

4.2.2. *Mesh balanced in individual face*

The process of balancing individual face serves dual objectives. The first objective is analogous to that of conventional conforming mesh, which is to restrict the cell aspect ratio under 2, as shown in Fig. 12(a). In the case of an extremely narrow face, the aspect ratio of the cell is relaxed to face aspect ratio/10. Furthermore, each cell edge has maximum of one hanging node, and any additional hanging nodes should be subdivided along hanging nodes, as depicted in Fig. 12(b). The results following the completion of mesh balancing are presented in Fig. 10(c). Obviously, the transition from dense cells to coarse cells is distinct and seamless.

5. Constrained boundary recovery

Confirming the absolute consistency of the recovered boundary with the original geometric boundary and topology is a crucial demand. The primary drawback of the tree-structured method lies in the inadequate quality of the meshes along the geometric boundary. The constrained boundary recovery algorithm proposed in this paper effectively retains all the geometric characteristics of the loop while preserving the favorable shape of cells. Since the nonconforming mesh on the geometric edges do not need to be continuous, the discrete of each face can

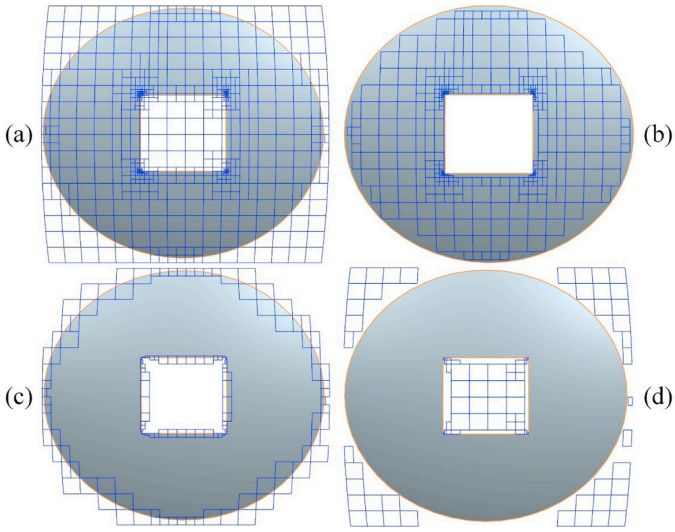


Fig. 13. Classify all cells (a) total cells, (b) inner cells, (c) rim cells, (d) outer cells.

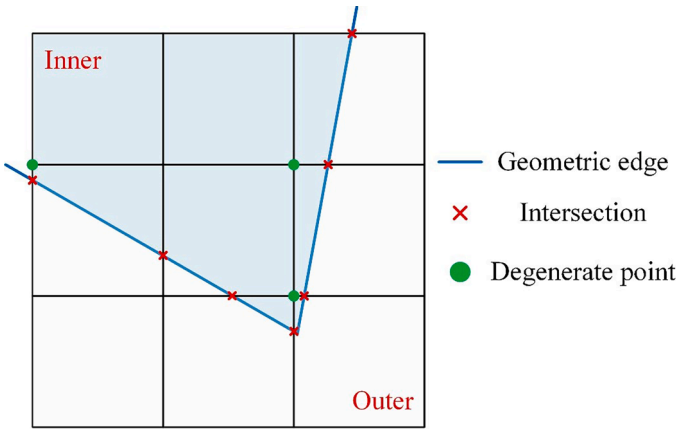


Fig. 14. Identification degenerate point.

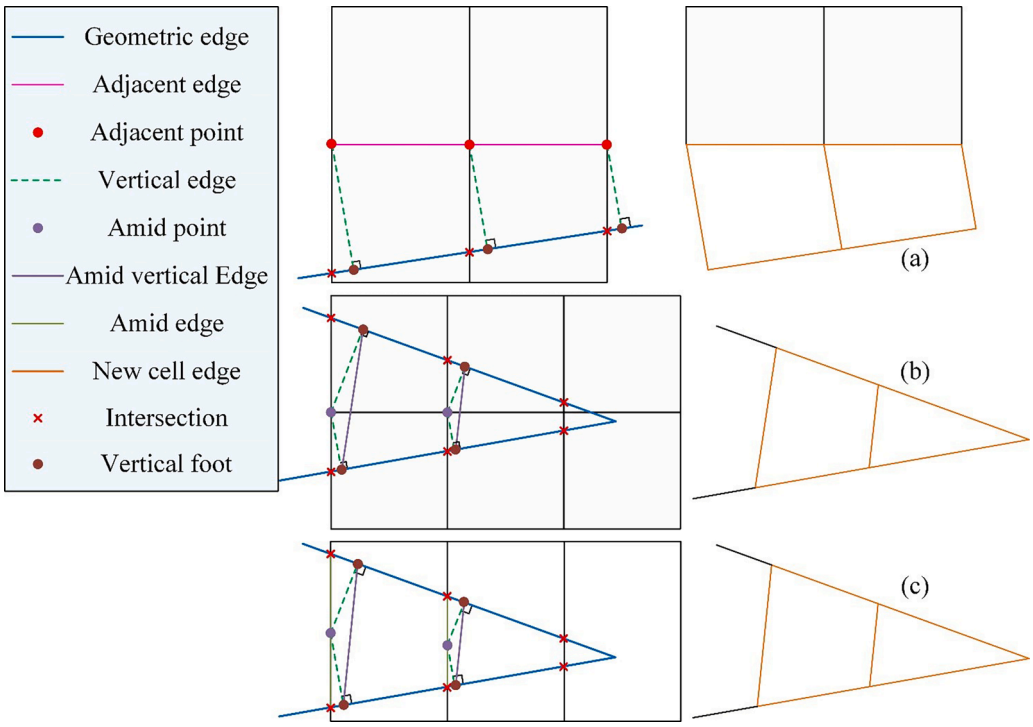


Fig. 15. Common boundary template.

be carried out independently.

5.1. Classification of cells according to the position relative to loop

Categorization of cells into three types, namely external (invalid), internal (valid), and boundary (requiring boundary recovery), is determined based on the presence or absence of intersections on each cell, as shown in Fig. 13. The outer cells refer to the mesh located outside the loop. Since these cells do not contribute to the numerical computation, they are considered invalid. Cells that intersect with the loop are termed rim cells. These cells possess partial presence within and outside the loop, and the rim cells will be split by the boundary recovery algorithm. The inner cells are inside the loop that can directly engage in the calculation without involving any adjustments.

5.2. Boundary template

Every intersection on the cell edge can seek out a corresponding cell node inside the loop, and calculate the distance between the cell node and the boundary. The cell node is designated as a degenerate point when the distance is below 0.3 times the length of the cell edge, as illustrated in Fig. 14. Degenerate points will be eliminated during the boundary recovery process. Rim cells are categorized into three boundary templates by mean of degenerate points and concave geometry vertices. Empirical evidence indicates that cell edges with a 90° to the geometric boundary are most favorable for numerical calculation. Therefore, it is preferable to align the cell edge vertical to the loop whenever feasible in boundary recovery.

5.2.1. Common boundary template

Discretization of the rim cell follows the common boundary template in the absence of degenerate points. The cell edge of intersects the loop in rim cell contains two cell nodes, with one positioned inside the loop and the other outside the loop. The cell node situated within the loop is labeled as an adjacent point, while the connection of contiguous adjacent points gives rise to adjacent edges, depicted as in Fig. 15(a). The green dashed line in Fig. 15 represents the vertical edge, connecting the

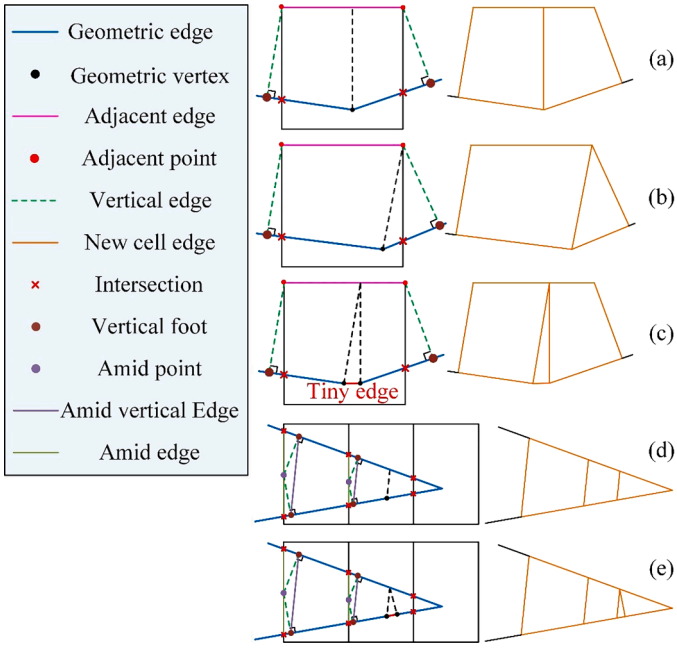


Fig. 16. Geometric vertices processing.

adjacent point to the loop at a perpendicular angle. The vertical foot in the vertical edge serves as a cell node on the loop. Connecting consecutive vertical feet on the loop generates a new cell edge. A quadrilateral cell is formed by adjacent edge, vertical edges, and new cell edge.

In the case where a single cell node is present between the opposite geometric edges, as illustrated in Fig. 15(b), it is labeled as an amid point. Here the vertical edge is connected to the amid point and loop. At least two vertical edges are connected to each amid point. Connecting these two vertical feet located on different geometric edges creates the amid vertical edge, while the loop enclosed by these vertical feet forms a

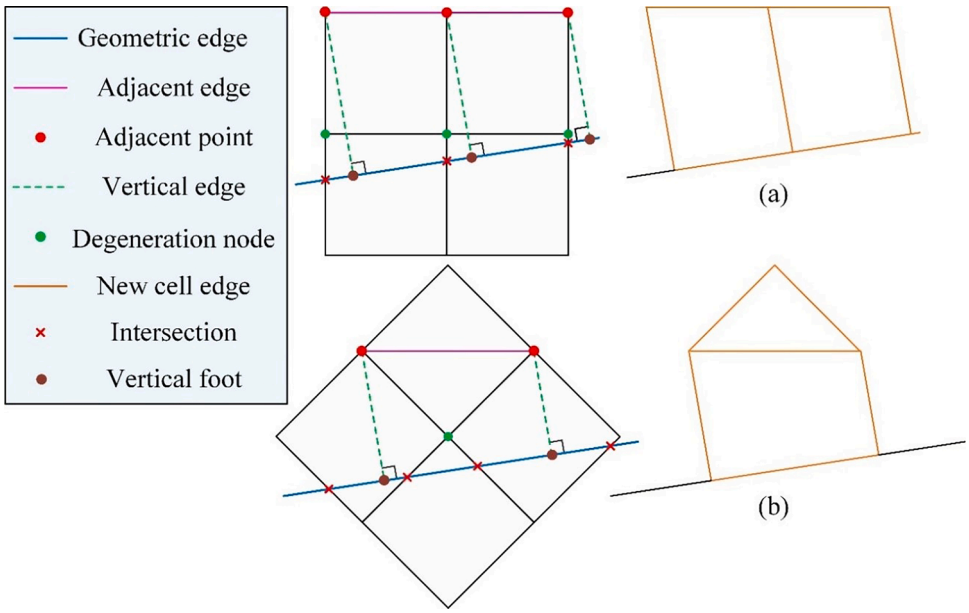


Fig. 17. Adhere boundary template.

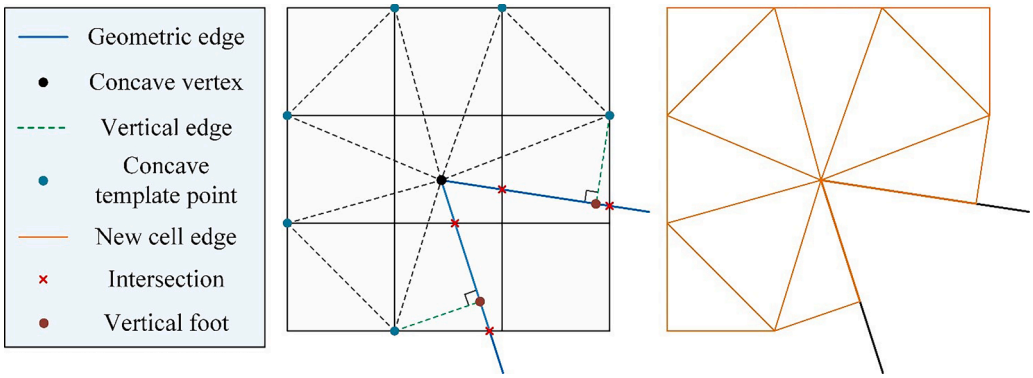


Fig. 18. Concave boundary template.

quadrilateral or triangular cell. Furthermore, in the absence of cell nodes between the opposite geometric edges, specifically when there are two intersections on the cell edge, as depicted in Fig. 15(c), the edge connecting these two intersections is referred to as the amid edge. The middle point of the amid edge is designated as the amid point. A quadrilateral or triangular cell, resembling the configuration shown in

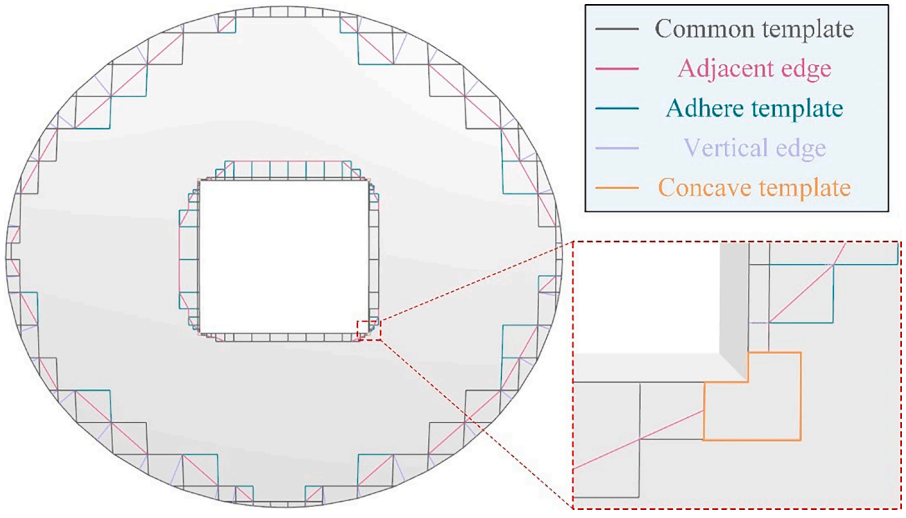


Fig. 19. All boundary template for example.

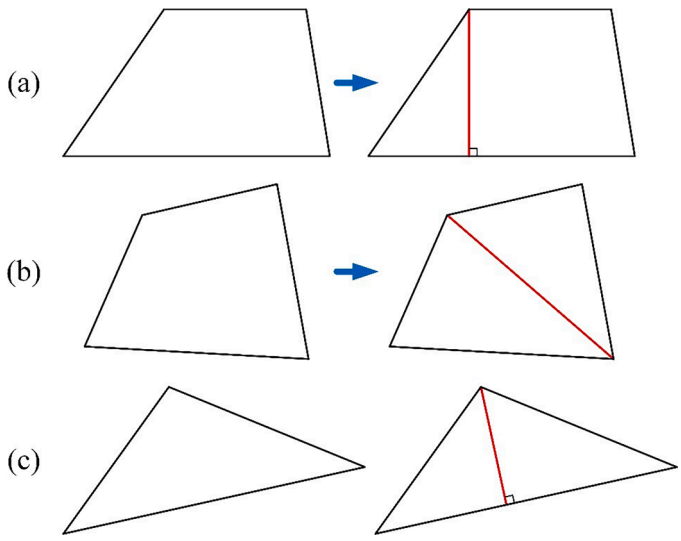


Fig. 20. Eliminate large angles.

Fig. 15(b), is formed in Fig. 15(c).

In the presence of geometric vertex between two vertical feet, a vertical edge is necessary to create along geometric vertex toward the adjacent edge, resulting in discretization of two quadrilateral cells, as shown in Fig. 16(a). When the vertical foot is positioned close to the adjacent point, it leads to a configuration comprising of a quadrilateral cell and a triangular cell, as illustrated in Fig. 16(b). Furthermore, in the presence of tiny edge between the two vertical feet as depicted in Fig. 16 (c). Any vertex of tiny edge can be selected to create a vertical edge towards the adjacent edge, resulting in discretization of two quadrilateral cells and one triangular cell. A comparable methodology was employed for the amid edge and amid point, as illustrated in Fig. 16(d) and (e).

5.2.2. Adhere boundary template

The rim cells containing the degenerate points is processed by Adhere boundary template. The adjacent point in the Adhere boundary template is the closest cell node to the degenerate point within the loop. Fig. 17(a) shows the cell generation process for a degenerate point connecting an adjacent point. The methodology for constructing quadrilateral cells follows the same principles as the common boundary template. In the case where a degenerate point connects two adjacent points (Fig. 17(b)), connect two adjacent points to obtain an adjacent edge. A quadrilateral cell and a triangular cell are generated along the adjacent edge in this case.

5.2.3. Concave boundary template

Creating the vertical edges for the adjacent point near the concave vertices is challenging, and thus requires the use of a dedicated discrete method to split rim cells. The cell node near to the cell where the concave vertex is located is labeled as a concave template point (CTP), as shown in Fig. 18. The two nearest CTPs to the loop are connected to the loop by the vertical edge. Each CTP is connected to concave vertex to form a new cell edge. The remaining new cell edge is obtained by connecting the adjacent CTPs. The vertical edge and new cell edge to formation of multiple triangular cells that encompass concave vertex. This treatment of the concave boundary template yields cells with desirable shape characteristics. The outcome of the all boundary template construction is depicted in Fig. 19.

5.3. Eliminate large angle

To this stage, some cells with large angle exist though most of the cells maintain a high level of mesh quality. Optimization is performed necessarily by a straightforward process to eliminating large angles that surpass a predetermined angle threshold (120° in this study). At the initial stage, vertical edge is considered to split the large angle to form a quadrilateral and a triangular cell, as shown in Fig. 20(a). This approach contributes to avoid the creation of larger angles during the split process. In case of vertical foot of the vertical edge lies near the cell node, the cell can be subdivided two triangular cells along the diagonal line formed by large angle, as depicted in Fig. 20(b). The triangular cell shown in Fig. 20(c) also uses a vertical edge to eliminate large angles. Fig. 21 shows the final discrete results of the presented example.

6. Numerical examples

This section presents a series of numerical examples with and

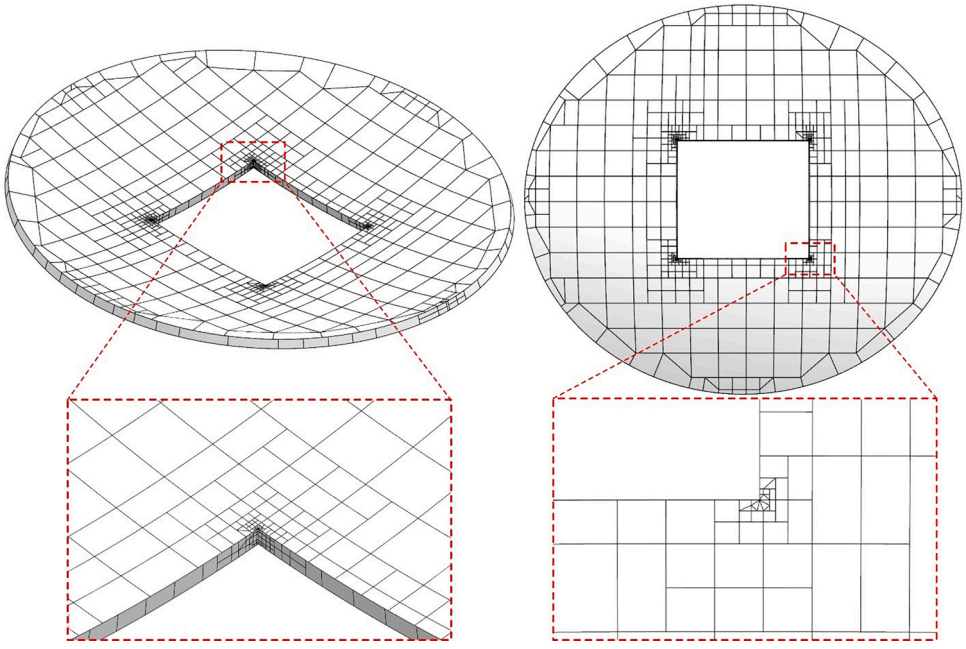


Fig. 21. Mesh result for example.

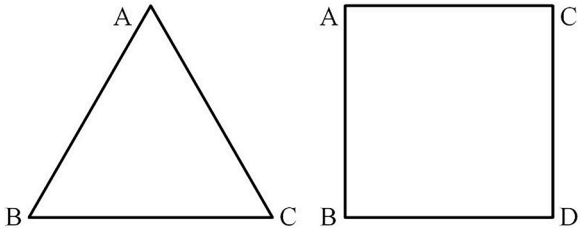


Fig. 22. Triangular cell and quadrangular cell.

without "dirty" geometry to showcase the effectiveness and stability of the BSSM. All the programs are implemented in the C++ and executed on a computing device equipped with an Intel(R) Core(TM) i5-8250U CPU operating at a clock speed of 1.8GHz. Both BSSM and ANSYS are executed in single-threaded mode. The figures in this section display the final mesh results required by the boundary face method (BFM) [29], wherein the curved mesh is obtained through the interpolation along each cell edge.

6.1. Mesh quality

The precision and accuracy of subsequent numerical calculations are determined by the quality of the mesh. A widely method for evaluating mesh quality is the mesh quality factor (*MQF*) [30]. In the case of the $\triangle ABC$ (Fig. 22 left), the *MQF* is defined as

$$MQF_3 = 2\sqrt{3}\ \frac{\| CA \times CB \|}{\| CA \|^2 + \| AB \|^2 + \| BC \|^2} \tag{3}$$

where *AB, CB, AB, BC* represent the directional vectors of the three edges of the triangle separately. The *MQF* ranges from 0 to 1, where a value of 1 indicates the highest mesh quality, while a value of 0 represents the lowest. For quadrilateral *ABCD* (Fig. 22 right), its *MQF* can be determined by the *MQF* of its constituent $\triangle ABD$, $\triangle ACD$, $\triangle BAC$, and $\triangle BDC$. The calculation procedure is as follows

$$MQF_4 = \frac{MQF_{3\min1} MQF_{3\min2}}{MQF_{3\max1} MQF_{3\max2}} \tag{4}$$

where $MQF_{3\min1}$ and $MQF_{3\min2}$ represent the smaller values, while $MQF_{3\max1}$ and $MQF_{3\max2}$ represent the larger values. In addition, the maximum and minimum angles of the cell can provide insights into its quality. Due to the presence of tiny geometric angles in "dirty" geometry, this section employs the maximum angle as a key metric to evaluate the mesh quality. An optimal cell is indicated by a maximum angle value approaching 90°

Section 6.1 presents the mesh results of CAD models with "dirty" geometry that were discretized utilizing the proposed BSSM, as well as the results obtained using ANSYS (Version 2020 R2). ANSYS employs a quad-dominant discretization approach, and controls the number of cells in ANSYS to be comparable to BSSM.

Fig. 23 displays the CAD model of the clock hand, accompanied by the mesh results obtained through BSSM and ANSYS. The tiny geometric edge is indicated by a red dashed line, measuring only 1/100th the length of the neighboring geometric edges. It is apparent that BSSM effectively preserves the tiny edge and employs it as the cell edge on the adjacent faces. The quadrilateral cells demonstrate nearly rectangular shapes and exhibit high level of mesh quality. The triangular cells retain a tiny angle at the geometric tip. Conversely, the ANSYS disregard the tiny edge in the geometric repair. Suboptimal mesh shape in areas of "dirty" geometry is caused by imperfect geometry repair. According to the mesh quality statistics presented in Table 1, BSSM successfully preserves all "dirty" geometry, with an *MQF* value that is merely 1.9 % lower than that of ANSYS. Through its method of eliminating large angles, BSSM achieves a maximum mesh angle that is 24.1 % lower than that obtained by ANSYS. Consequently, the mesh generated by BSSM demonstrates greater suitability for numerical calculations compared to ANSYS.

Table 1
Statistic of cells generated for CAD model.

CAD model	Mesh generating method	Total cell number	Average value of mesh quality	Maximum angle (°)	Mesh generation time(s)
Clock hand	BSSM	458	0.885	108.2	0.017
	ANSYS	457	0.902	142.4	0.688
Engine shell	BSSM	43313	0.906	118.6	2.989
	ANSYS	42566	0.948	165.9	54.620

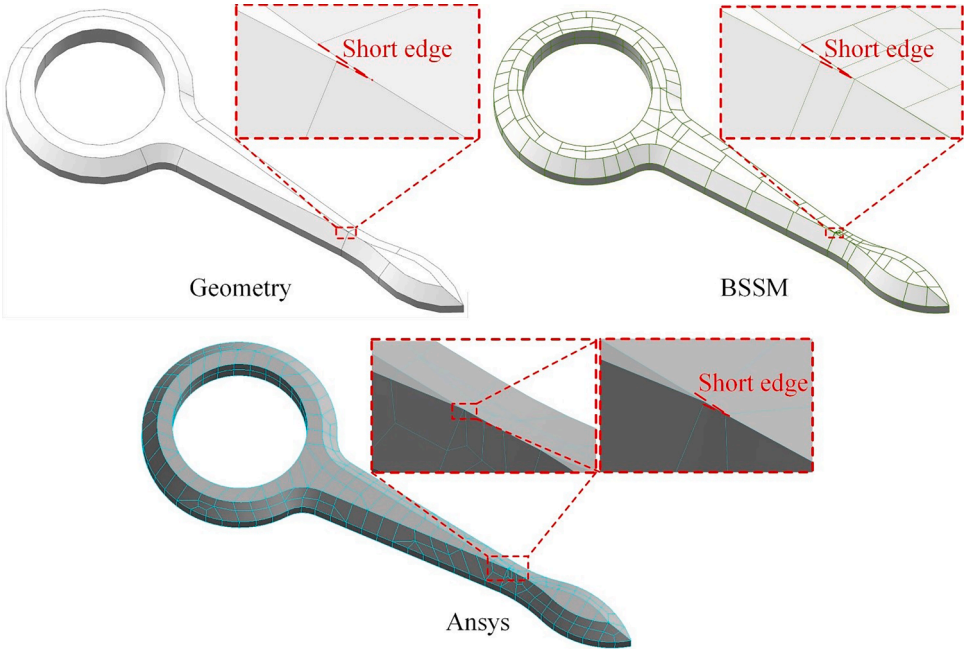


Fig. 23. Mesh result for clock hand.

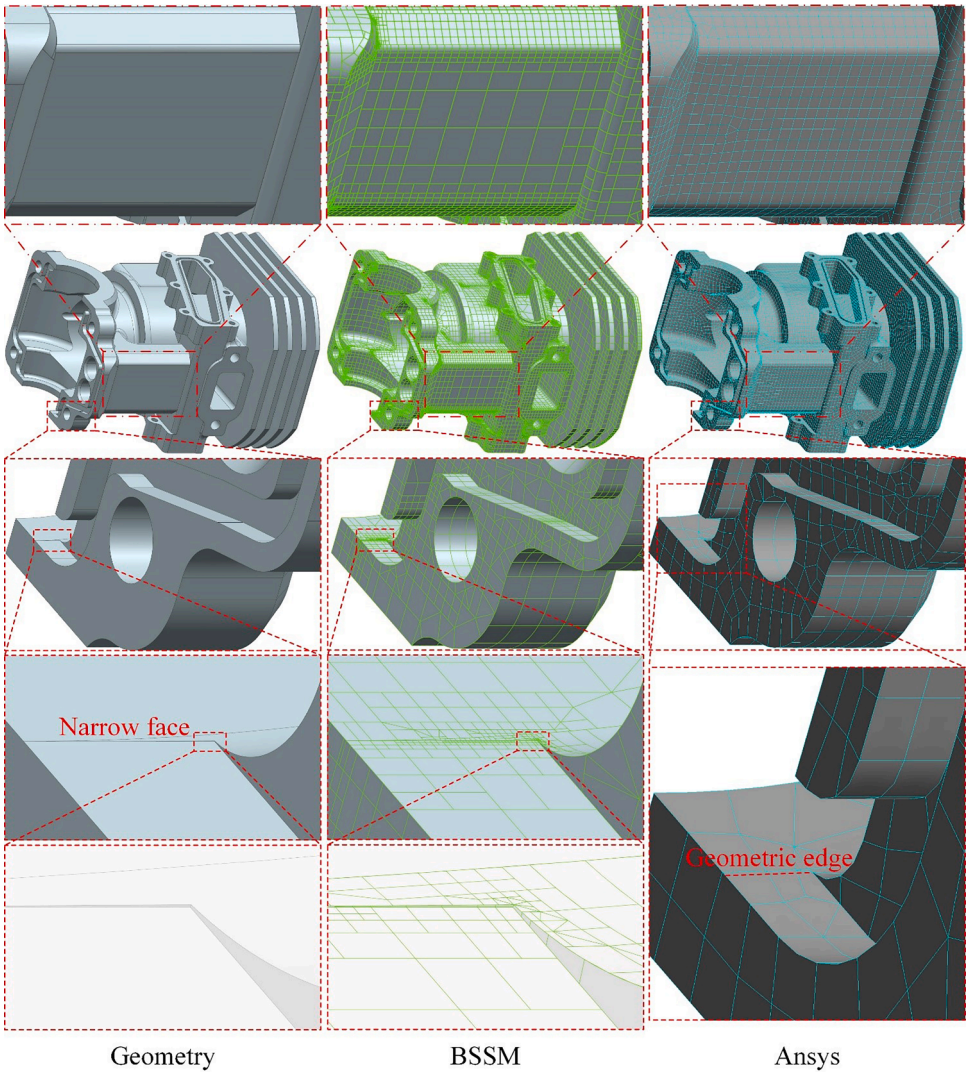


Fig. 24. Mesh result for engine shell.

To demonstrate the advantages of BSSM for narrow faces, an engine shell was discretized into over 42000 cells using both BSSM and ANSYS methods. Fig. 24 displays the narrow face with aspect ratios exceeding 500, and the BSSM mesh results preserve the narrow face intact. There is little effect on the surrounding face mesh as only the tangential direction of the narrow face is balanced. ANSYS entirely abandons all attributes of the narrow face, resulting in its transformation into a geometric edge (red dashed line in Fig. 24 bottom). However, cell edges are not preserved on this geometric edge, which means that ANSYS cannot implement constrain boundary recovery. Moreover, on a larger regular face (Fig. 24 top), BSSM achieves the same computational accuracy as ANSYS with only one-third of the number of cells. Both methods accomplish higher *MQF*s (Table 1), with ANSYS reaching an *MQF* of 0.948 through disregards the narrow face. But ANSYS has a maximum angle measures 165°, which is 28 % larger than the BSSM and have a significant negative impact on the corresponding region.

The mesh generation speeds of BSSM and ANSYS are also shown in Table 1. It is evident that BSSM exhibits considerably faster performance compared to ANSYS in both examples. This speed advantage stems from BSSM's capability to drastically reduce the number of intersection calculations between the background mesh and the geometric boundaries, as well as the boundary templates effectively minimize the frequency at which the cell nodes are mapped to each other in 3D space and to the parameter space.

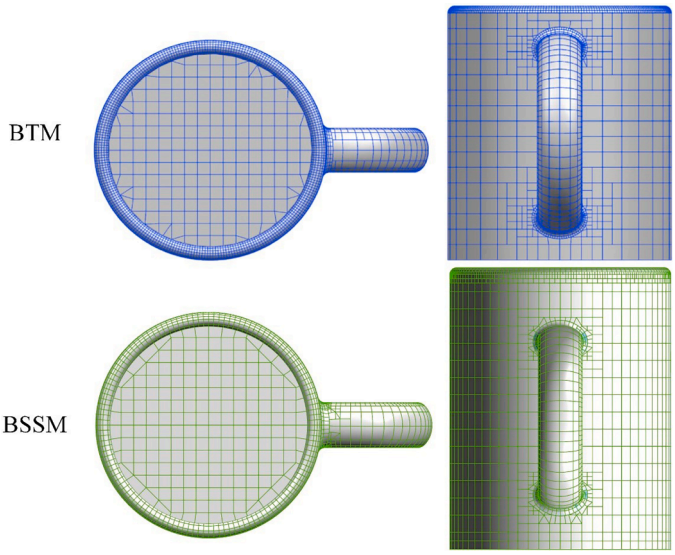


Fig. 25. Comparison of mesh results between BTM (Fig. 38 in literature [24]) and BSSM.

Table 2
Statistic of cells generated for BTM and BSSM.

Mesh generating method	Total cell number	Average value of mesh quality	Mesh generation time (s)
BTM	4586	0.985	8.634
BSSM	3633	0.978	0.482

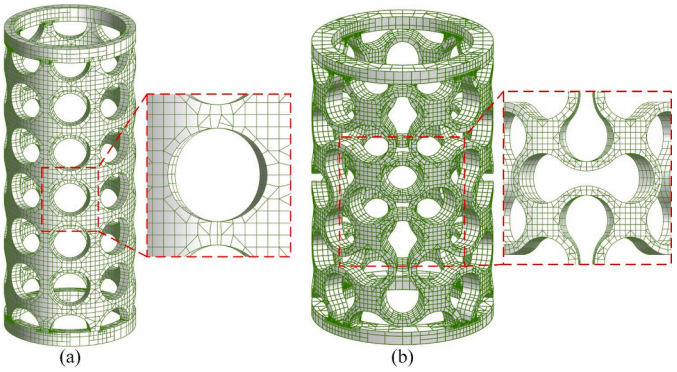


Fig. 26. Mesh result for porous tubes.

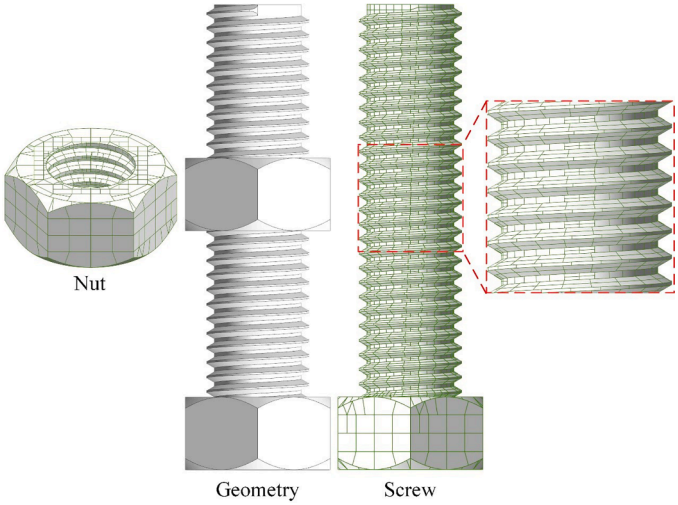


Fig. 27. Mesh result for screw and nut.

6.2. Compare with BTM

Fig. 25 illustrates two nonconforming mesh results for a cup, where BTM [24] is our previously utilized algorithm. In contrast to BTM, BSSM employs topological information to delineate the internal and external boundaries of the geometric face, leading to markedly reducing the number of intersection calculations between the background mesh and loop. In order to mitigate the numerical computation errors between neighboring faces, a process of mesh size balancing on neighboring faces is introduced. Furthermore, based on our computational experience suggests that achieving an approximate 90° angle between the boundary cell edge and the loop yields more accurate numerical computation results. Therefore, the boundary template is redesigned to reduce the number of templates and ensure the quality of the boundary mesh. From Fig. 25 and Table 2, it is evident that BSSM not only markedly enhances the computational speed, but also facilitates smooth transition between adjacent geometrical faces, resulting in a comparable mesh quality with fewer cells.

Table 3
Statistic of cells generated for CAD model.

Fig	CAD model	Total cell number	Average value of mesh quality	Maximum angle (°)	Mesh generation time (s)
Fig. 26	(a)	19120	0.901	110.5	5.069
	(b)	12389	0.904	104.4	2.865
Fig. 27	Nut	1544	0.775	98.3	0.499
	Screw	4950	0.818	96.5	3.175
Fig. 28	(a)	53164	0.873	108.6	2.135
	(b)	11219	0.897	102.7	0.397
Fig. 29	Ring	1772	0.959	106.6	0.067

6.3. Mesh result

This section presents a series of representative mesh discretization results for engineering models without “dirty” geometry.

Firstly, two porous circular tubes [31] with intricate microstructures are shown in Fig. 26, discretized into 19120 and 12389 cells respectively. Both structures exhibit an *MQF* exceeding 0.9 and maximum angle below 111° Due to the presence of numerous narrow boundaries within the porous structure, the dimensions of the structure are relatively uniform, resulting in a predominantly conforming mesh. This demonstrates that BSSM can generate conforming mesh with guaranteed high quality.

Fig. 27 displays the mesh results for a screw and nut. The presence of spiral narrow faces generation significant challenges in mesh discretization of screws and nuts, often requiring a substantial reduction in mesh size to ensure accurate division. Likewise, as BSSM relaxes the aspect ratio of narrow face cells, the *MQF* of screws and nuts (Fig. 27 in Table 3) is considerably lower compared to other instances. Nevertheless, the screw mesh consists of regular rectangular cells, leading to maximum angle close to 90°

The mesh results for the commonly employed gears in mechanical equipment are depicted in Fig. 28. Gear (a) generates 53164 cells, with an average *MQF* of 0.873 and a maximum angle of 108.6° The teeth of gear (a) are composed of multiple narrow faces, resulting in numerous small cells of similar size on tooth of gear. Therefore, the utilization of nonconforming meshing is unable to significantly reduce the number of cells for this gear. Gear (b) generates 11219 cells with an *MQF* of 0.897 and a maximum angle of 102.7° Both gears exhibit excellent mesh quality according to the mesh results.

As illustrated in Table 3, the gear with the highest cell counts exhibits faster mesh generation compared to the screw with fewer than one-tenth of the number of cells. This discrepancy arises from the inherent characteristics of the threaded faces in the screw, which are composed of complex spline faces. Consequently, a substantial number of iterations are required to calculate intersection positions and cell node space mappings. In contrast, gears primarily comprise faces with minimal curvature, thereby simplifying the most time-consuming process.

For periodic faces such as the sphere and cylindrical face, the periodic face topology analysis methodology is thoroughly explained in Section 3.9 of the literature [24]. A periodic face is highlighted in yellow in Fig. 29. Mesh statistics are presented in Table 3, indicating 1772 cells and a mesh generation time of 0.067s. The result demonstrates that the BSSM is capable of rapidly discretizing periodic faces resulting in the generation of high-quality mesh. Furthermore, in the case of non-watertight structures such as gaps, it is necessary to create a new face at the gap to form a watertight structure and utilize BSSM for mesh generation. Since the repair of non-watertight structures is not currently performed, numerical examples are provided.

7. Conclusions

This paper presents a novel and comprehensive approach BSSM for

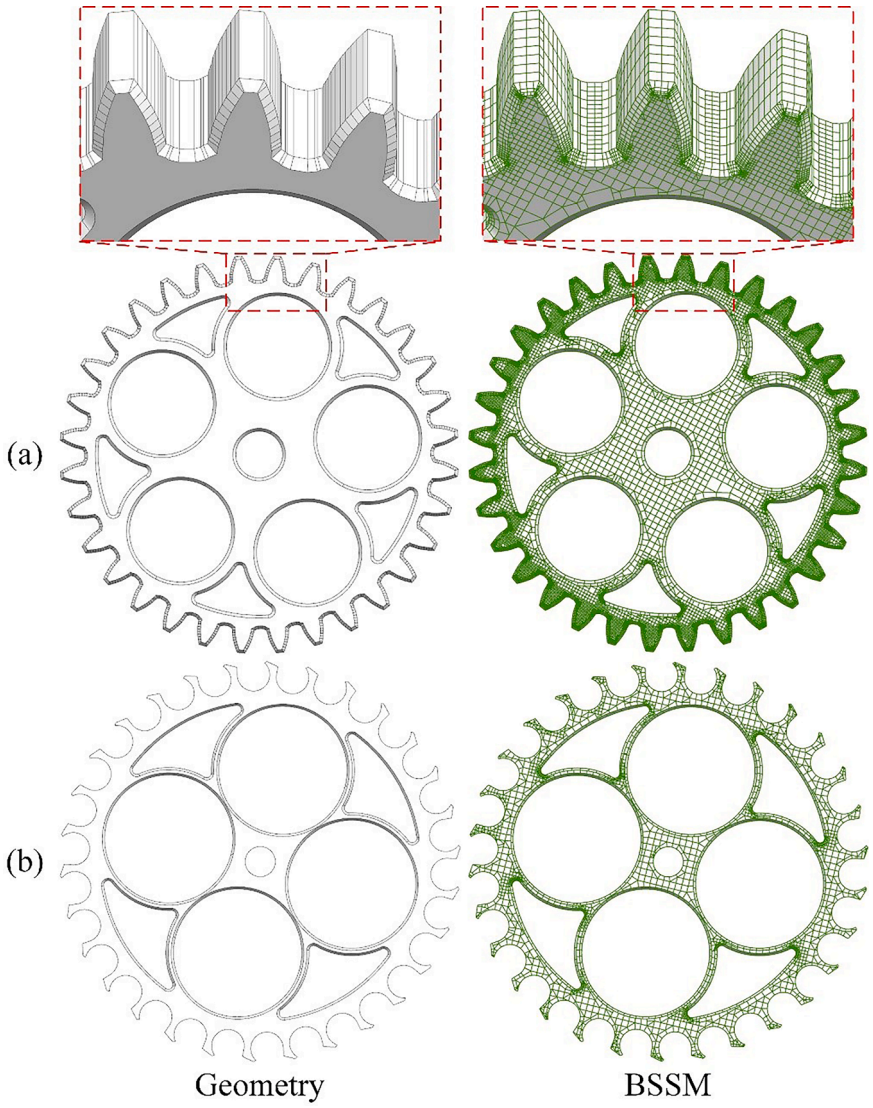


Fig. 28. Mesh result for gears.

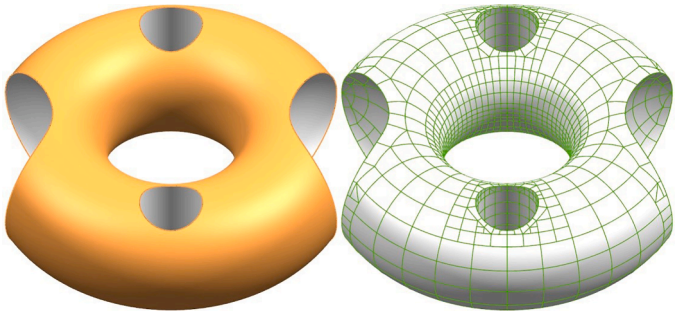


Fig. 29. Mesh result for periodic face.

adaptive surface mesh discretization. BSSM leverages the topological information and geometric parameters of CAD models, providing a novel approach to deal with "dirty" geometry and generate high quality mesh while preserving all geometric features. The main conclusions of this research are drawn as follows:

(1) The BSSM eliminates the need for mesh mapping in most steps, thereby accelerating the solution speed. This is achieved by efficiently calculating the intersection between the background mesh and the geometric boundary, as well as generating the vertical edge within the boundary template.
(2) By permitting hanging nodes, BSSM demonstrates the ability to generate a mesh of superior quality while preserving all geometric features of the CAD model. Notably, although the commercial software ANSYS entirely discards small features, the mesh quality factor achieved by BSSM remains comparable. Besides, it is important to highlight that the maximum cell angle in ANSYS is substantially larger than BSSM.
(3) BSSM showcases its capability to produce high-quality conforming mesh when the geometric features within the structure are of similar sizes.
(4) BSSM exhibits strong universality and robustness in handling complex structures, enabling fully automatic discretization of arbitrary complex structures independent of manual operation.

Finally, it should be emphasized that the application of BSSM is currently limited to the boundary element method. While the conversion of nonconforming mesh to conforming mesh is not excessively arduous, it does necessitate non-standard treatment. This is exemplified by the difficulty of local mesh refinement due to the elimination of hanging points.

CRediT authorship contribution statement

Jianming Zhang: Writing – review & editing, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition. **Chong Zhang:** Writing – review & editing, Writing – original draft, Visualization, Validation, Formal analysis, Data curation. **Rongxiong Xiao:** Writing – review & editing, Validation, Supervision, Data curation. **Baotao Chi:** Writing – review & editing, Validation, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Jianming Zhang reports financial support was provided by National Natural Science Foundation of China. Baotao Chi reports financial support was provided by National Natural Science Foundation of China. Baotao Chi reports financial support was provided by China Postdoctoral Science Foundation. Baotao Chi reports financial support was provided by Ministry of Education Industry-school Cooperative Education Project. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported by National Natural Science Foundation of China under grant numbers, 12172126 and 12202251; China Postdoctoral Science Foundation, China (2021M702024); Ministry of Education Industry-school Cooperative Education Project, China (220606517023742).

Sincere thanks to Prof. Lubomir Ondris from Jantárová 8, 040 01 Košice, Slovak Republic for his valuable comments on the paper.

References

[1] Shimada K. Current issues and trends in meshing and geometric processing for computational engineering analyses. J Comput Inf Sci Eng 2011;11:021008.
[2] Thompson JFSB, Weatherill NP. Handbook of grid generation. CRC press; 1998.
[3] Tristano JROSJ, Canann SA. Advancing front surface mesh generation in parametric space using a riemannian surface definition. In: Proceedings of the 7th international meshing roundtable; 1998. p. 429–45.
[4] Lan TS, Lo SH. Finite element mesh generation over analytical curved surfaces. Comput Struct 1996;59:301–9.
[5] Guan ZQ, Shan JL, Zheng Y, Gu YX. An extended advancing front technique for close d surfaces mesh generation. Int J Numer Methods Eng 2008;74:642–67.
[6] Wang W, Fan HZ, Xi G. An extension of advancing front technique on new target surface after virtual topology operations. Adv Eng Softw 2018;124:42–52.
[7] 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 Des 2013;45:1408–25.
[8] 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 Des 2022;153:103403.
[9] Baehmann P, Wittchen S, Shephard M, Grice K, Yerry M. Robust, geometrically based, automatic two-dimensional mesh generation. Int J Numer Methods Eng 1987;24:1043–78.
[10] Guo JW, Ding F, Jia XH, Yan DM. Automatic and high-quality surface mesh generation for CAD models. Comput Aided Des 2019;109:49–59.
[11] Fang TP, Piegl LA. Delaunay triangulation using a uniform grid. IEEE Comput Graph Appl 1993;13:36–47.
[12] Shephard M, Yerry M. A modified quadtree approach to finite element mesh generation. IEEE Comput Graph Appl 1983;3:39–46.
[13] Legrain G, Allais R, Cartraud P. On the use of the extended finite element method with quadtree/octree meshes. Int J Numer Methods Eng 2011;86:717–43.
[14] Liang XZ. 2D/3D mesh generation via different tree structures. Carnegie Mellon University; 2013.
[15] Beall M, Walsh J, Shephard M. Accessing CAD geometry for mesh generation. In: Proceedings of the 12th international meshing roundtable; 2003.
[16] Butlin G, Stops C. CAD data repair. In: Proceedings of the 5th international meshing roundtable; 1996.
[17] Dey S, Shephard MS, Georges MK. Elimination of the adverse effects of small model features by the local modification of automatically generated meshes. Eng Comput 1997;13:134–52.
[18] Foucault G, Cuillière JC, François V, Léon JC, Maranzana R. Adaptation of CAD model topology for finite element analysis. Comput Aided Des 2008;40:176–96.
[19] Khan D, Plopski A, Fujimoto Y, Kanbara M, Jabeen G, Zhang YJ, Zhang XP, Kato H. Surface remeshing: a systematic literature review of methods and research directions. IEEE Trans Vis Comput Graph 2022;28:1680–713.
[20] Wang J, Huang J, Wang FL, Wei MQ, Xie HR, Qin J. Data-driven geometry-recovering mesh denoising. Comput Aided Des 2019;114:133–42.
[21] Yu F, Cao J, Shan JL, Lo SH, Guan ZQ. PASM: Parallel aligned surface meshing. Int J Numer Methods Eng 2021;122:3705–32.
[22] Fine L, Remondini L, Leon JC. Automated generation of FEA models through idealization operators. Int J Numer Methods Eng 2000;49:83–108.
[23] Liu TR, Ye HF, Zheng JJ, Zheng Y, Chen JJ. Advancing front mesh generation on dirty composite surfaces. Comput Aided Des 2024;49:103683.
[24] Ju CM, Zhang JM, Xiao RX, Chi BT. Automatic surface mesh generation by a binary-tree method. Eng Anal Bound Elem 2023;152:473–95.
[25] Xiao RX, Zhang JM, Chai PF, Ju CM, Lin WC, He R. Dual interpolation boundary face method for 3-D acoustic problems based on binary tree grids. Eng Anal Bound Elem 2023;150:7–19.
[26] Chai PF, Zhang JM, Xiao RX, He R, Lin WC. A multi-domain BEM based on dual interpolation boundary face method for 3D elasticity problem. Eng Anal Bound Elem 2022;143:568–78.
[27] Zhang JM, Xiao RX, Wen PH, Ju CM, Lin WC, 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.
[28] Zhang JM, Chai PF, He R, Lin WC, Ju CM, Chi BT. Implementation of a dual interpolation boundary face method by discontinuous meshes. Eng Anal Bound Elem 2022;139:152–68.
[29] Zhang JM, Qin XY. A boundary face method for potential problems in three dimensions. Int J Numer Methods Eng 2009;80:320–37.
[30] Lee CK, Lo S. A new scheme for the generation of a graded quadrilateral mesh. Comput Struct 1994;52:847–57.
[31] Zhang C, Xiao SH, Qin QH, Wang H. Tunable compressive properties of a novel auxetic tubular material with low stress level. Thin Walled Struct 2021;164:107882.