www.cadanda.com
Computer-Aided Design AND APPLICATIONS

Conforming Hybrid Mesh Generation for Solids with Tubular Holes

Cheng Huang[1], Jianming Zhang[2], Guizhong Xie[3], Chenjun Lu[4] and Guangyao Li[5]

[1]Hunan University, huangcheng@hnu.edu.cn
[2]Hunan University, zhangjm@hnu.edu.cn
[3]Hunan University, agui11712@163.com
[4]Hunan University, birdfly8888@hnu.edu.cn
[5]Hunan University, gyli@hnu.edu.cn

ABSTRACT

A new conforming hybrid mesh generation method is proposed and applied to generate mesh for three-dimensional solids with open-ended tubular holes. A pyramidal element generation method is presented to solve the non-conforming problem. The procedure consists of five steps. At first, surfaces of tubular holes are recognized and discretized. Secondly, the regions around the tubular holes are presented by hexahedral elements. Then, the surface triangular mesh is generated. After that, the pyramid elements are constructed. Finally, the tetrahedral mesh is generated to tessellate the rest of the domain. By the proposed method, high-quality hybrid mesh can be automatically created. And the number of the obtained elements is far less than the all-tetrahedral mesh. Several examples are presented to demonstrate the robustness and efficiency of our method.

Keywords: hybrid mesh, pyramid, advancing front method, sweeping, Delaunay method.

1. INTRODUCTION

Structures containing many small tubular holes find important application in engineering, such as cooling passages, electrical wiring passages, etc. For instance, in construction process of concrete dams many slender holes are designed as cooling passages. These pipes have significant influences on the resulting temperature and flux distribution in a body. Thus the design engineers draw much attention to properly arrange these pipes.

The analysis of solids with small tubular holes is still a difficult task by finite element method (FEM), as the FEM requires a very fine mesh to make sure the accuracy of the result. Usually, very large amount of modeling efforts are necessary for mesh generation in a body with many small holes, especially when the holes are very small in size. The boundary face method (BFM) [16] has been introduced to analyses steady-state heat conduction problem [13] and elastic problem [7] of three-dimensional solids with small tubular shaped holes of free shape. Usually, the boundary element method only needs to mesh the surface of the domain. The volume mesh is generated for boundary face method. The volume mesh is used to apply volume integration for the boundary face method and also be used to obtain and display the results of the arbitrary internal points.

The all-hexahedral mesh is preferred for numerical analysis when available [1,3]. Although the tetrahedral mesh can be generated automatically for the solids with small features, it requires more elements than an all-hexahedral mesh to obtain an equally accurate numerical solution. The number of elements has a large impact on the data and time costs for numerical analysis. However, all-hexahedral mesh automatic generation has proven to be more challenging. The hybrid mesh is suitable for meshing based on complex geometry, because the hybrid mesh generation method can generate appropriate elements for different regions. The hybrid mesh has the high mesh quality strategy and contains fewer elements than all-tetrahedral mesh. A number of hybrid mesh generation methods have been developed and implemented for complex geometries [2,9,11].

This paper presents a three-dimensional hybrid mesh generation method to discretize three-dimensional solids with open-ended tubular holes. The hybrid mesh is consists of hexahedral, tetrahedral and pyramidal elements. The region around these tubular holes is represented by hexahedral elements and the rest of the domain with complex geometry

Taylor & Francis

is described by tetrahedral elements. In this case, two tetrahedral elements are required to interface with one hexahedral element. Thus, discontinuities will arise at the boundary between the two kinds of meshes. In order to overcome the problem of non-conforming meshes, a pyramidal mesh generation method is proposed in this study. One layer of pyramidal elements is formed at the interface between hexahedral and tetrahedral elements. Three methods have been presented for insertion of pyramidal elements in an existing hybrid mesh at interface between hexahedral and tetrahedral elements in reference [12]. In our method, the pyramidal elements are generated after the hexahedral meshing and surface triangular meshing. Pyramids are constructed from each of the quadrilateral element of the interface hexahedron. Once the pyramidal elements have been generated, the tetrahedral mesh is formed to tessellate the rest of the domain. The proposed method is similar to the advancing front method for tetrahedral mesh generation. High quality pyramidal elements can be generated by our method within a reasonable time limit. The final hybrid meshes have the ease of tetrahedral meshing on complex geometries with the advantages of hexahedral cells in regions around the holes. Compared to all-tetrahedral meshes, our method can generate far fewer elements within a reasonable time limit, resulting in substantial savings in both data preparation and computing costs. And our method can generate mesh for structures containing holes of arbitrary shape.

In the following sections, section 2 describes the general scheme of the mesh generation algorithm. The identification and discretization of the tubular holes' surfaces and hexahedral mesh generation process is described in section 3. Pyramidal mesh generation process is described in section 4. In section 5, the tetrahedral mesh generation method is discussed. Several examples are provided in section 6. And the paper ends with conclusions in Section 7.

2. SCHEME OF THE METHOD

Our method consists of five steps. Firstly, the surfaces of tubular holes are recognized and discretized. Secondly, the hexahedral mesh is generated around tubular holes. Then, the surface triangular mesh is generated. After that, the pyramid elements are constructed. Finally, the tetrahedral mesh is generated in the rest of the domain. The scheme of the hybrid mesh generation method is described in Fig. 1. Let's take a cube with a tubular hole for example. The mesh generation process starts from the recognition of surfaces of tubular holes. Then the central line of the tubular hole is discretized. The hexahedral mesh is generated by sweep method along the central line of the tubular hole, as shown in Fig. 1(a). After hexahedral mesh generation, edges of the domain are discretized, and new fronts are added into the initial front array. Then surface triangular mesh is generated by advancing front method, as shown in Fig. 1(b). The pyramidal elements are constructed by the proposed method, as shown in Fig. 1(c). Finally, the rest of the domain is discretized into tetrahedral elements by the Delaunay coupled with advancing front method. Then, we get the final hybrid mesh, as shown in Fig. 1(d).

An overall mesh size and a mesh size for sweep meshing are defined by user. The Sweep mesh size is approximates to the minimum distance between the outside quadrangle of the hexahedral mesh and the tube face. The overall mesh size and mesh size for sweeping have to be modified by the minimum radius of curvature of the tube, the minimum distance between cylinders and the minimum distance between cylinders and the boundary of the domain.

S is the overall mesh size, St is the mesh size for sweeping, Dc is the minimum distance between cylinders, Dcb is the minimum distance between cylinders and the boundary of the domain and Rc is the minimum radius of curvature of the tubes.

If $St > 1.4 \times Dcb$, the St changed to $1.4 \times Dcb$ and S is changed to $1.3 \times St$.

Then, if $St > 0.6 \times Dc$, the St is changed to $0.6 \times Dc$ and S is changed to $1.3 \times St$.

Then, if $St > 0.8 \times Rc$, the St is changed to $0.8 \times Rc$ and S is changed to $1.3 \times St$.

Each of these steps is briefly described in the following sections. First, identification and discretization of the tubular holes' surfaces and the hexahedral mesh generation is detailed step by step. Afterwards, the pyramid generation and the tetrahedral mesh generation are described.

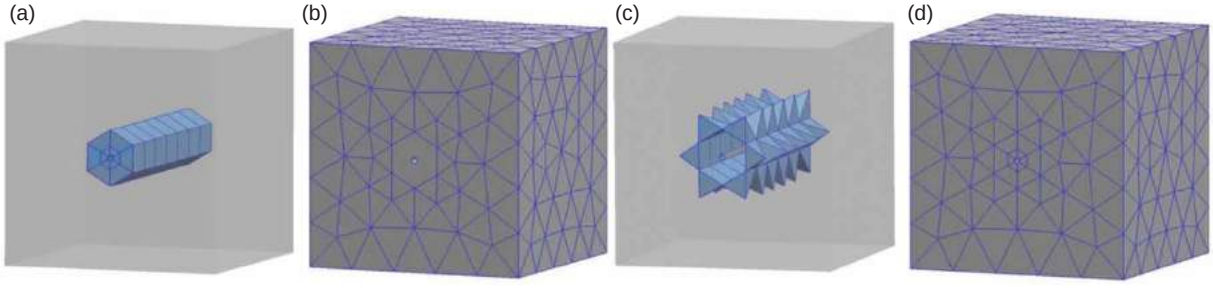

Fig. 1: The scheme of the hybrid mesh generation method.

3. HEXAHEDRAL MESH GENERATION

3.1. The Recognition of the Surface of the Tubular Hole

At first, surfaces of tubular holes need to be recognized from the geometry model. The BFM is implemented directly based on the boundary representation data structure (B-rep) that is used in most CAD packages for geometry modeling. In the B-rep scheme, a solid is modeled by its boundary faces which are bounded regions on planes or surfaces. Without concise mathematical formula for its explicit representation, such a boundary face is described by its bounding loops, edges and vertices [15]. In this work, the rule-based feature recognition method is used to recognize the surface of the tubular hole. In this approach, a set of heuristic rules are used to describe the operational definition for a class of features, which are similar in shape. A definition of a class of the surface of a tubular hole is quoted below.

- The surface of a tubular hole is a closed surface in the parametric space;
- The surface of a tubular hole contains two loops L_1 and L_2;
- L_1 contains only one edge E_1. E_1 is a closed curve;
- L_2contains only one edge E_2. E_2 is a closed curve;
- E_1 is an inner loop of another surface F_{n}. l_{avg} is the average length of edges contained in the outer loop of F_n. The length l_1 of $E_1 < 0.2\, l_{\mathrm{avg}}$;
- E_2 is an inner loop of another surface F_{m}. l_{avg} is the average length of edges contained in the outer loop of F_{m}. The length l_2 of $E_2 < 0.2\, l_{\mathrm{avg}}$. F_{n} and F_{m} may be the same surface.

Loop over surfaces of the domain. If a surface satisfies these above constraints, it is recognized as the surface of the tubular hole.

3.2. The Discretization of the Surface of the Tubular Hole

The discretization of the surface of the tubular hole contains two parts. One is the discretization along the circumferential direction of the tubular hole; the other is the discretization along the axial direction of the tubular hole. Assume that u is the axial direction and v is the circumferential direction of the surface $f(u, v)$ in the parametric space. The surface is periodic in the circumferential direction.

The discretization along the circumferential direction is the discretization of two edges of the surface of the tubular hole. The surface of the tubular hole is a closed surface. For a closed surface, the boundary polygon is not closed in the parametric space [6]. These two edges are periodic parametric curves. At first, one edge is discretized. Then, the first discretization node of the other edge needs to be located. In order to make sure these two first nodes of two edges are in the same period, and these discretization nodes of two edges are consistent.

The aim of the discretization along the axial direction is to obtain the discretization of the central line of the tubular hole. However, there is not a central line of the tubular hole in the geometry model. We select two parallel curves of the surface. Let $u = 0$ and $u = 0.5$ are two curves of the surface $f(u, v)$ in the parametric space. These two curves are discretized by the parametric curve discretization algorithm with the same rule. After the discretization of these two curves, the middle nodes of the corresponding two nodes of these two curves are calculated. These middle points are the discretization of the central line of the tubular hole. In this work, the algorithm according to the arc length rule for the discretization of 3D parametric curve is applied [4]. Appling this process, finally we can get the discretization of the central line of the tubular hole.

3.3. Sweep Method for Hexahedral Mesh Generation

Hexahedral elements are generated by the sweep method around the tubular holes. Sweeping is a method of meshing two and one half dimensional volumes with an all hexahedral mesh [14]. A two and one half dimensional volume is a volume that has a topologically constant cross section along a single axis known as the "sweep axis." These volumes can be defined by a source surface, a target surface and a series of link surface. The source and target surfaces may have different areas and curvatures. However, they must be topologically equivalent. Thus, they must have the same number of holes and logical sides. Furthermore, link surfaces have to be mapped. Hence, they must have only four logical sides.

In this work, a predigest sweep method is applied to generate hexahedral mesh. At first, a quadrilateral mesh is generated over the source surface. Then, the target surface mesh is generated. After that, the inner layers of nodes and elements and the structured quadrilateral mesh over the linking-sides are generated at the same time.

Firstly, a quadrilateral mesh is generated over the source surface. In a solid contains tubular holes, there is one or two surface intersected by a tubular hole. One surface is set as source surface; the other is set as target surface. If a tubular hole intersects one surface twice, the source surface and target surface will be the same surface. Not the whole source surface is discretized, only the region of the surface around the hole is discretized into quadrilateral mesh. It is a simple quadrangular mesh. According to the number of segments along circumferential direction of the hole and the number of layer of the quadrangular mesh, the quadrangular mesh around the hole is created. This quadrangular mesh is set as source

surface mesh, and the sweep method is applied to generate hexahedral mesh along the axial direction of the tubular hole.

The discretization of the central line of the tubular hole has been obtained in the previous step. The number of segments of the central line is the number of layers of the hexahedral elements. For every layer, there is one outer loop, and one inner loop for each hole in the sweep volume. The discretization of the out loop and inner loop of each layer can be easily gotten. Then, the following algorithm is used to generate hexahedral mesh along the central line of the tubular hole.

After the quadrangle mesh generation of the source surface, the quadrangle mesh is projected onto the target surface. The projection method between parametric surfaces and the least-squares approximation are applied to obtain the target surface mesh. The mapping is defined between the parametric spaces of these two surfaces. Thus, this sweep algorithm can be applied when the source surface and target surface have different curvatures. Our program is implemented directly based on the boundary representation data structure (B-rep). Each bounding surface of geometry model is represented as parametric form by the geometric map between the parametric space and the physical space. The parametric coordinates of nodes of the source and target surfaces can be easily obtained.

Then, the rest boundary nodes and inner nodes of the extrusion volume have to be generated. These nodes have to be placed, layer by layer, along the axial direction of the tubular hole. The details of the projection method can be found in references [14]. After the boundary and inner nodes of k-th level are generated, the respect hexahedral elements and the structured quadrilateral mesh over the link surfaces can be easily generated by joining the corresponding nodes between adjacent layers.

4. PYRAMIDAL MESH GENERATION

The mesh which contains tetrahedral and hexahedral elements is a non-conforming mesh. In order to solve this problem, a pyramidal mesh generation method is proposed. After the generation of hexahedral mesh, pyramidal elements can be generated by each quadrilateral element of the interface hexahedral elements. Our approach is similar to the advancing front method for tetrahedral mesh generation [8].

The quadrilateral elements of the interface hexahedral elements are set as fronts. These fronts which are used to generate new pyramidal elements are called active fronts. The front which is using to generate a pyramidal element is current front. If a new pyramidal element is generated by the current front, then the current front is set as an inactive front. Otherwise, the current front is set as a rejected front.

There are two kinds of approach to generate a new pyramidal element. One is generating a new node by the current front. The new node is linked to the four nodes of the current front to generate a pyramidal element. Another is generating a new pyramidal element by an existing node and the current front. The ideal node and existing nodes are called candidate nodes.

4.1. Ideal Node Creation

The ideal node is created by current front according to the local element size. At first, the average length of four edges of current front is calculated. Then, the position of the ideal node is calculated according to the average length. The lengths of the four new edges are desired to equal to the average length, so the new triangles are closed to equilateral triangle.

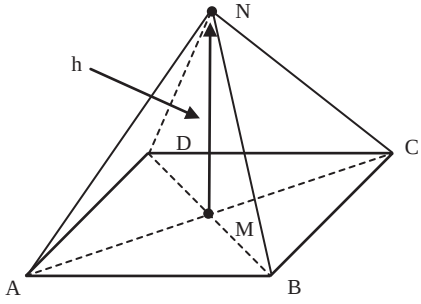

Fig. 2: Ideal node creation.

The creation process of the ideal node is shown in Fig. 2. Assume the current front is the quadrangle $ABCD$. The coordinates of its four nodes are $A(x_A, y_A, z_A)$, $B(x_B, y_B, z_B)$, $C(x_C, y_C, z_C)$ and $D(x_D, y_D, z_D)$. The average length of edges of the quadrangle is h.

1) Calculate the centroid M of the current front.

$$x_M = \frac{1}{4}(x_A + x_B + x_C + x_D),$$
$$y_M = \frac{1}{4}(y_A + y_B + y_C + y_D),$$
$$z_M = \frac{1}{4}(z_A + z_B + z_C + z_D) \tag{4.1}$$

2) Calculate the unit normal vector $\mathbf{n}$ of the current front.

$$\mathbf{n} = \frac{\mathbf{AB} \times \mathbf{A}D}{\|\mathbf{AB}\| \cdot \|\mathbf{A}D\|} \tag{4.2}$$

3) Calculate the coordinate of ideal node N. Assume O is the origin of coordinates. Then, $\mathbf{ON} = \mathbf{OM} + 0.98154h\mathbf{n}$. The coordinate of N can be calculated by

$$N_x = \mathbf{ON}.i, \quad N_y = \mathbf{ON}.j, \quad N_z = \mathbf{ON}.k. \tag{4.3}$$

A new pyramid must satisfy both the validity and the suitability criteria to be accepted as an element of the mesh.

4.2. New Element Creation with an Existing Node

Existing nodes in the neighborhood are considered as candidate nodes if they are positioned close to the current front. Set the ideal node as the center of the sphere, a searching sphere is created. Its radius is R, and $R = a^*h$. Usually, a is between 0.5 and 1.5. The nodes located in this sphere are set as candidate nodes. The nodes are ordered on increasing distance from the node N. During element generation, if a new element is constructed by a candidate node, then turn to the next active front.

4.3. Try Nodes

A new pyramid must satisfy the validity and the suitability criteria to be accepted. Situations might arise where neither the ideal node nor nay existing node satisfy both these criteria. In such case several try nodes [10] are constructed. The try nodes are usually located on the normal line through the node N, as shown in Fig. 3.

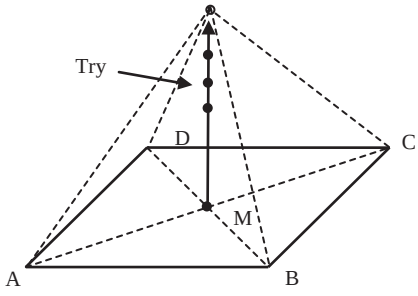

Fig. 3: Try nodes.

4.4. General Scheme

The main steps of the pyramidal mesh generation algorithm can be summarized as follows:

Step 1. The quadrilateral elements of the interface hexahedral elements are set as active fronts. Then the inactive front array is empty.

Step 2. Select an active front as current front. Calculate the coordinate of node M, and the element size h. Then determine the position of the ideal node N.

Step 3. Set the node N as the center of the sphere. A searching sphere is created. Its radius is R, and $R = 1.5^*h$. The nodes located in this sphere are set as visible nodes. The triangle and quadrangle linking to these visible nodes are set as visible triangular and visible quadrangle. The edges belongs to the visible elements are set as visible edges.

Step 4. The ideal node and visible nodes are set as candidate nodes. The candidate nodes are considered in turn to generate a new pyramid. When a new pyramid satisfies both the validity and the suitability criteria, the candidate node is used to generate a new pyramid.

- The vector **MN** is from the centroid to the candidate node. The angle between **MN** and **n** is less than 45 degrees;
- The new element dose not contain any exist node;
- The distance between the ideal node and visible triangles and visible quadrangles is more than 0.65^*h;
- The distance between the new edges and visible edges is more than 0.65^*h;
- The new edges do not intersect any visible element;
- The new triangles do not intersect any visible edges.

Step 5. If a new pyramid is created, update the front and the current mesh.

- Add the new pyramid into the current mesh;
- If the new triangle exists into the current mesh, remove it. If not, add the new triangle into the current mesh;
- Remove the current front from the active front array.

Step 6. If a new pyramid can not be created by current front, remove the current front from the active from array and set the current front as the rejected front.

Step 7. If the number of active fronts is not zero, the next active front is set as current front. And return to step 2.

Step 8. If the rejected front is empty, the mesh generation is complete. If not, set the rejected front as active front. Then, the criterion for element check is less strict. The try nodes are set as candidate nodes, instead of ideal node. And go to Step 1. Repeat the process until the rejected front is empty. Assume that the current candidate node is the k-th try node and the $Rate = 0.9k$ $(k = 1, 2, \ldots, 5)$. The validity and the suitability criteria are revised as follows.

- The angle between **MN** and **n** is less than 45 $/Rate$ degrees;
- The new element dose not contain any exist node;
- The distance between N and visible triangles and visible quadrangles is more than 0.65^*h^*Rate;
- The distance between the new edges and visible edges is more than 0.65^*h^*Rate;
- The new edges do not intersect any visible element;

- The new triangles do not intersect any visible edges.

If pyramids can not be generated with rejected fronts, an additional draw back process is applied to regenerate the pyramid around the reject fronts. Just like the AFM for tetrahedral mesh generation, the generated pyramids around the rejected front are deleted and the corresponding quadrilateral elements are set as active front. And the rejected fronts are arranged at the beginning of the active front array. Then, the pyramid generation process is applied. Repeat the process until the rejected front is empty.

Because the pyramids generation is before the tetrahedral mesh generation, the environment for pyramids generation is not geometry complex. Some examples have proved that the proposed algorithm can converge after sever repetition of the pyramid generation process with rejected front.

5. TETRAHEDRAL MESH GENERATION

In this section, we describe the scheme of the tetrahedral mesh generation. After the surface triangular mesh and the pyramidal mesh generation, the Delaunay method coupled with the advancing front method is applied to generate tetrahedral mesh [5]. The triangular elements of the pyramidal mesh and the surface triangular mesh are set as two-dimensional triangular mesh for tetrahedral meshing. We suggest using the advancing front method as an internal node creation tool, the Delaunay algorithm as an insertion tool and the Delaunay boundary tetrahedral mesh as a background mesh. Similar to the conventional advancing front method, the front is identified and internal nodes are created to form optimal tetrahedral elements in accordance with an element-size defined by the control space. A Delaunay kernel procedure is used to insert these nodes into the existing triangulation. Thus the convergence of the process is always ensured. The combined method has the high quality node placement strategy of the advancing front method and the high efficient and the mathematical properties of the Delaunay approach.

First step of the tetrahedral mesh generation algorithm is boundary tetrahedral mesh generation. At first, we construct a bounding box enclosing the domain. The box as a convex environment encloses all the boundary nodes. The box is defined by eight vertices and it is divided into five tetrahedrons. All the boundary nodes are inserted into the initial mesh using the Delaunay kernel. Then, the boundary is recovered, and we obtain the boundary tetrahedral mesh.

After the boundary tetrahedronization, the domain tetrahedral mesh is generated. The tetrahedral mesh is obtained by adding field nodes inside the existing mesh and then optimized so as to complete the final mesh of the domain. Similar to the conventional advancing front method, we define the suitable triangle generated in the last step as the front. And all the triangles of the boundary triangular discretization are defined as the initial fronts. We define the front which can be used to create new nodes as the active front. The internal nodes are iteratively created by the active front to form optimal tetrahedral elements and inserted into the previous mesh by Delaunay kernel. The quality of the generated mesh can be improved dramatically by several techniques of mesh optimization. So, after the mesh generation the Laplacian smoothing and topological optimization process are used to improve the quality of the mesh. This method can generate high quality tetrahedral mesh within a reasonable time limit.

6. EXAMPLES OF APPLICATION

In this section, several hybrid meshes are presented to demonstrate the robustness and efficiency of the proposed method. All the mesh generation process are carried out on the desktop computer with Intel (R) Core(TM)2 Duo CPU E7400 (2.80 GHz).

6.1. Block of a Practical Concrete Dam

The first example is one block of a concrete dam, as show in Fig. 4. It is a $15 \times 25 \times 4$ block. It contains a tubular cooling passage. The diameter of the cooling passage is 0.04, a very small value compared with the side length of the block. Usually, large quantities of elements are generated by conventional method. However, mesh generated by the present method just contains 1432 hexahedral elements, 716 pyramidal elements, 56446 tetrahedral elements, and 13607 mesh nodes. The CPU time needed for the mesh generation is listed in Tab. 1. A view of the hexahedral and pyramidal mesh is depicted in Fig. 5. Two layers hexahedral elements are generated. The final hybrid mesh is depicted in Fig. 6.

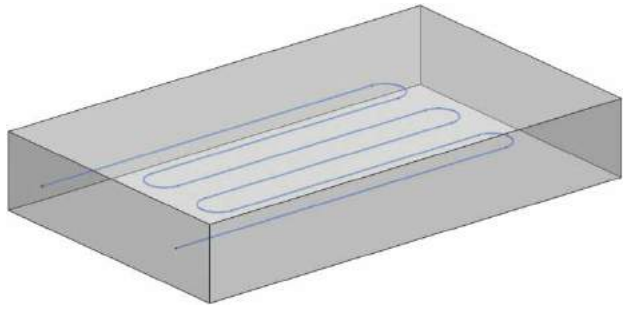

Fig. 4: A block of a concrete dam.

For tetrahedral mesh, the shape metric, Q_T, is the tetrahedral collapse. The tetrahedral collapse is cacu-lated by the following procedure. At each of the four nodes of the tetrahedron, the distance from the node

to the opposite side of the element is divided by the square root of the area of the opposite side. The minimum value found is normalized by dividing it by 1.24. As the tetrahedral collapses, this value approaches 0.0. For a perfect tetrahedron, this value is 1.0. For pyramid, the shape metric Q_p defined in reference [12] is used as the mesh quality estimate criteria. For tetrahedral and pyramidal elements, the element shape metric approaches 1.0 for a perfect element, and approaches 0.0 for degenerated element of zero volume. The Jacobian determinant J_m and condition number *Cond* are used as the mesh quality estimate criteria of hexahedral element [10]. When $0.5 \leq J_m \leq 1.0$ and $1 \leq Cond \leq 4$, the hexahedral element quality is acceptable for numerical analysis. Tab. 1 lists the element shape metrics of the hybrid mesh. Fig. 7(a) shows the shape metric classes of the tetrahedral elements generated by our method. We can see that high quality elements can be generated by our method.

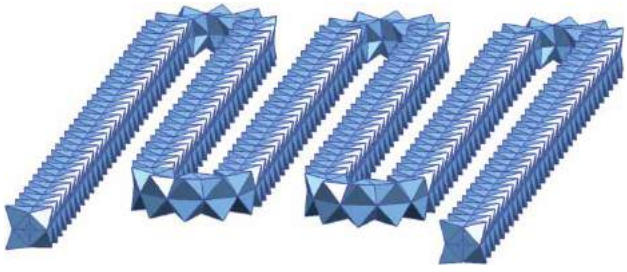

Fig. 5: The hexahedral mesh around the tubular hole.

In order to assess the advantages of the proposed method, the mesh generated by our method is compared to the mesh generated by Hypermesh. Hypermesh is a powerful meshing software. The resulting mesh is all tetrahedral mesh, because this solid with a curved hole is hard to be discretized into all hexahedral mesh automatically. Fig. 8 shows the mesh generated by Hypermesh. We can see that a very fine mesh is generated around the tubular hole. The mesh generated by Hypermesh contains 193190 mesh nodes and 1061195 tetrahedral elements. The total CPU time needed for the mesh generation is 68 seconds. The number of tetrahedral elements generated by Hypermesh is almost 18 times as large as the number of elements generated by our method. Compared to all-tetrahedral mesh, our method can generate hybrid mesh with far fewer elements, resulting in substantial savings in both data preparation and computing costs. Fig. 7(b) shows the shape metric classes of the mesh generated by Hypermesh. The mesh quality of the mesh generated by our method is as good as the mesh created by Hypermesh.

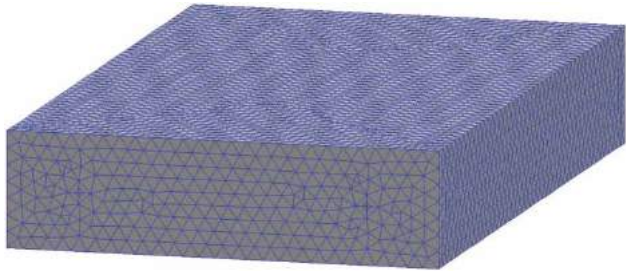

Fig. 6: The hybrid mesh.

6.2. Block with Several Holes of Arbitrary Shape

The second example is a $11 \times 12 \times 11$ block with 4 open-ended cylindrical holes of arbitrary shape, as shown in Fig. 9(a). The diameter of the holes is 0.2. This example is geometrically complicated. It is considered here to show the advantage of our method to generate mesh for structures containing open-ended tubular holes of arbitrary shape. It is very easy to generate mesh for this block with the proposed method. Mesh generated by our method just contains 320 hexahedral elements, 160 pyramidal elements, 8932 tetrahedral elements, and 2584 mesh nodes. The CPU

		Example 1	*Example 2*	*Example 3*
Tetrahedral mesh	Num. element	56446	8932	6885
	CPU Time	64 s	6 s	4 s
	Min. metric	0.362795	0.303524	0.258356
	Avg. metric	0.819253	0.800367	0.811476
Pyramidal mesh	Num. element	716	160	56
	CPU Time	2 s	<1 s	<1 s
	Min. metric	0.563902	0.450880	0.369464
	Avg. metric	0.761018	0.736676	0.789080
Hexahedral mesh	Num. element	1432	320	56
	CPU Time	1 s	<1 s	<1 s
	Min. J_m	0.669033	0.644548	0.498896
	Avg. J_m	0.702369	0.701924	0.696258
	Min. *Cond*	1.413770	1.384605	1.533187
	Avg. *Cond*	1.446319	1.451690	1.774295

Tab. 1: Mesh Results.

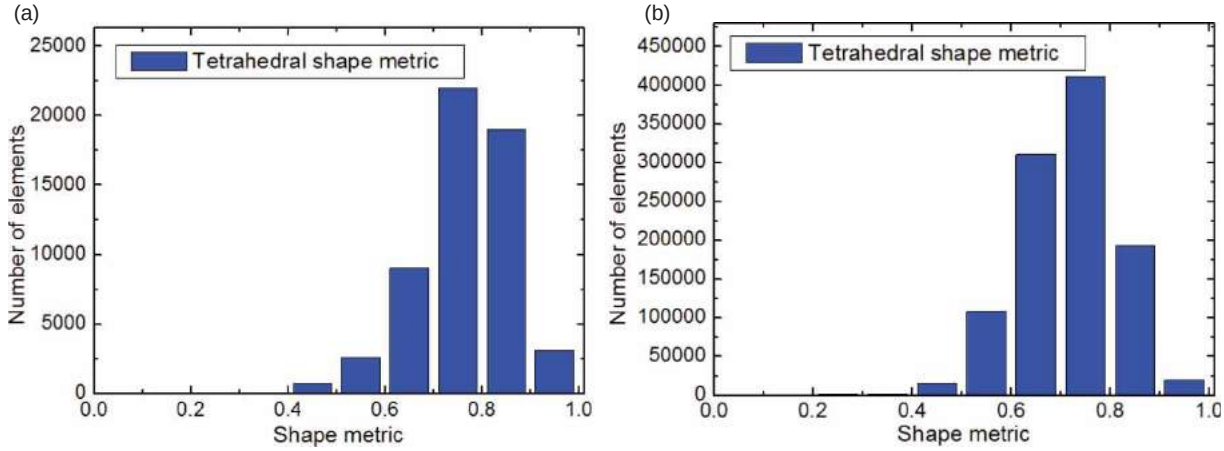

Fig. 7: Example 1: Tetrahedral mesh shape metric classes: (a) the mesh generated by our method, (b) the mesh generated by Hypermesh.

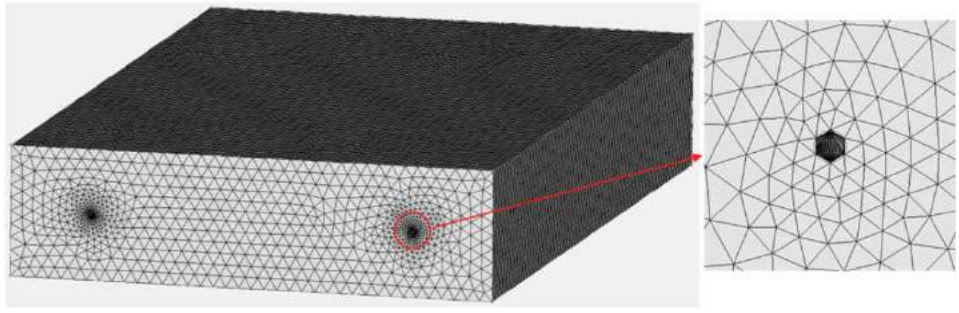

Fig. 8: The mesh generated by Hypermesh.

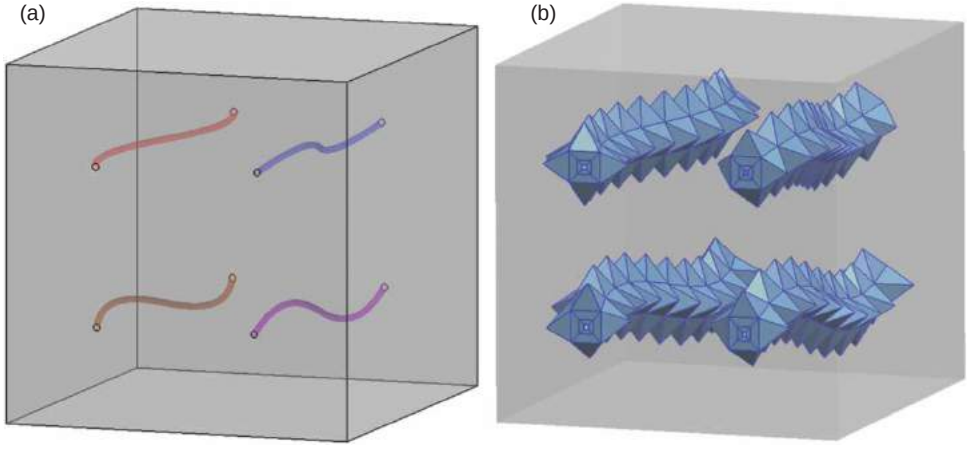

Fig. 9: (a) A block with 4 cylindrical holes of arbitrary shape, (b) the pyramidal mesh and hexahedral mesh.

time needed for the mesh generation is listed in Tab. 1. A view of the one layer of pyramidal elements and the hexahedral mesh is depicted in Fig. 9(b). The final hybrid mesh is depicted in Fig. 10(a). The element shape metrics of the hybrid mesh is listed in Tab. 1. Fig. 11 shows the shape metric classes of the tetrahedral elements generated by our method. In this case, our method also can generate high quality hybrid mesh; even the block contains 4 open-ended cylindrical holes of arbitrary shape.

The mesh generated by Hypermesh contains 13487 mesh nodes, 68154 tetrahedral elements. The total CPU time needed for the mesh generation is 3 seconds. The Fig. 10(b) shows the mesh generated by Hypermesh. And Fig. 11(b) shows the shape metric classes of the tetrahedral elements generated

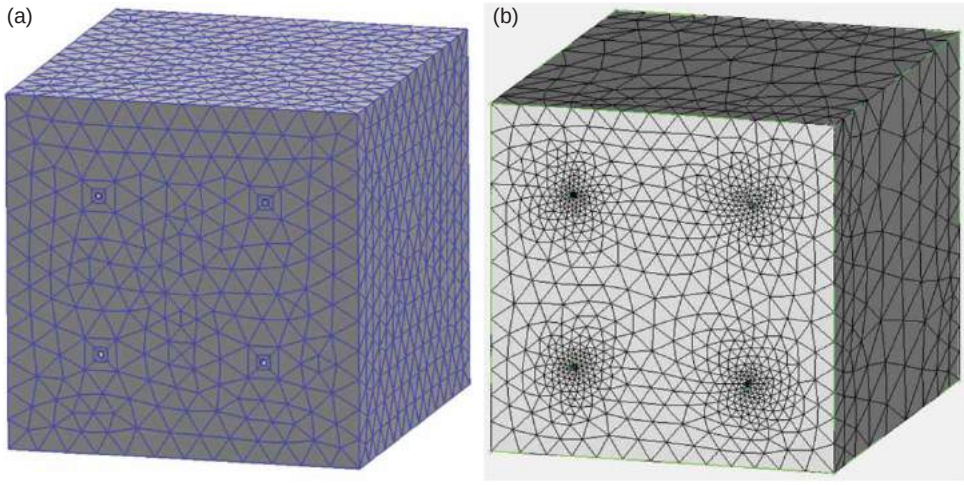

Fig. 10: (a) The hybrid mesh, (b) The mesh generated by Hypermesh.

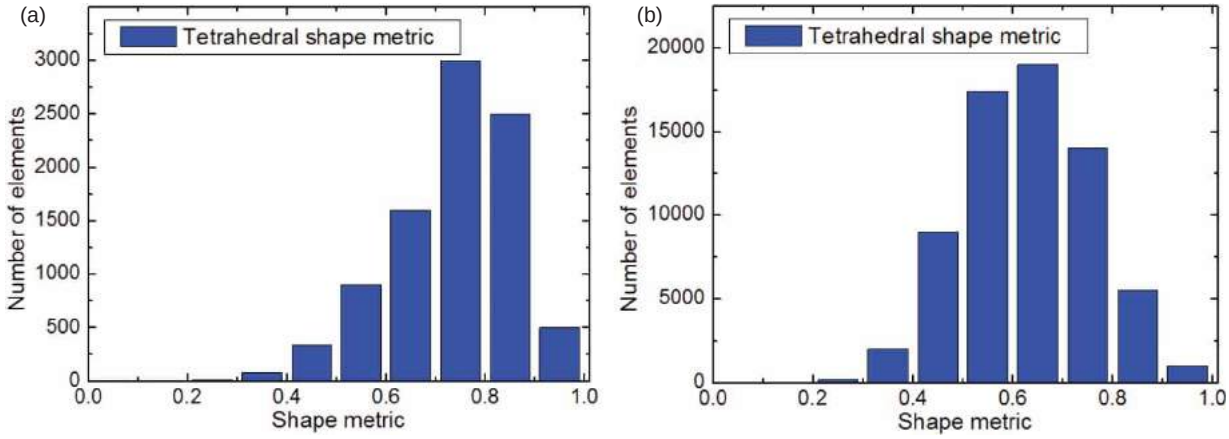

Fig. 11: Example 2: Tetrahedral mesh shape metric classes: (a) the mesh generated by our method, (b) the mesh generated by Hypermesh.

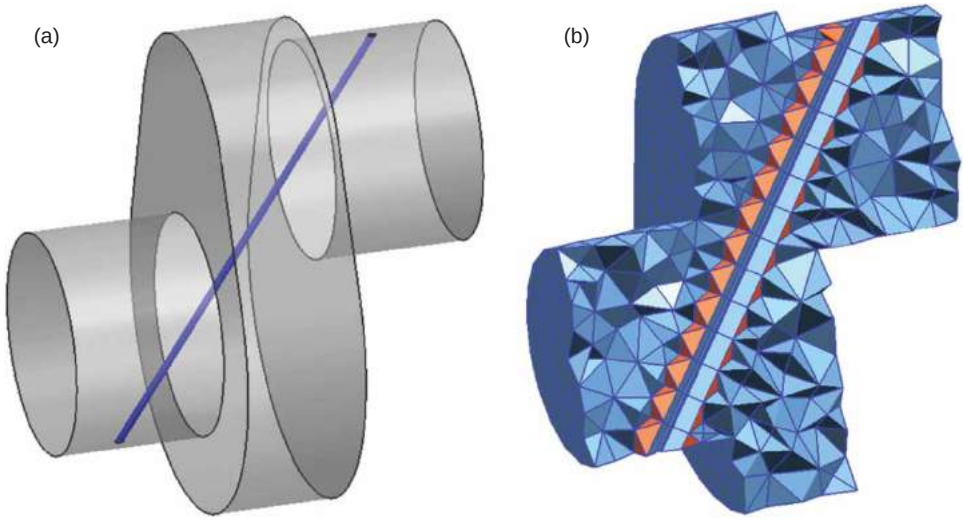

Fig. 12: (a) A part of a large crankshaft used by ship's diesel engine, (b) the tetrahedral mesh and pyramidal mesh.

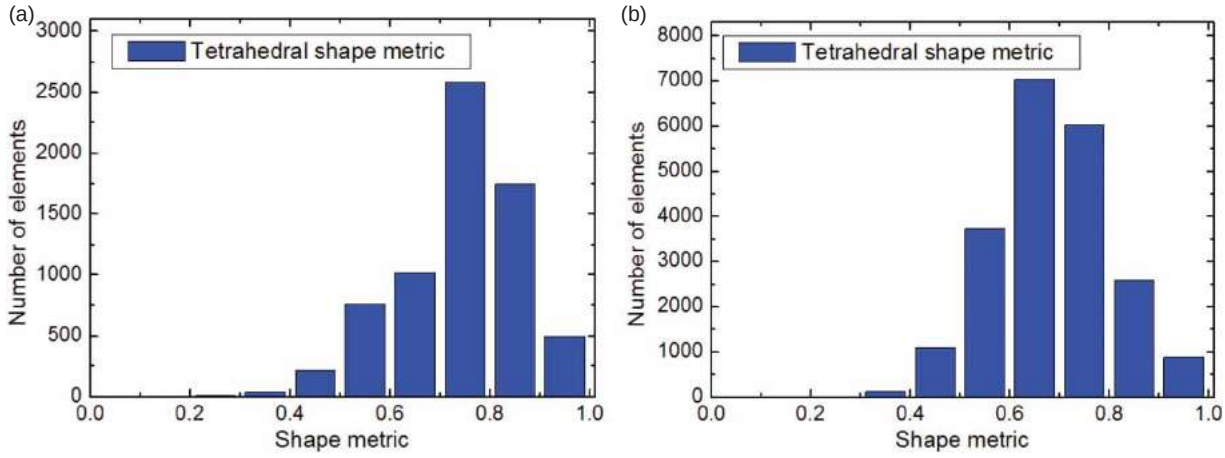

Fig. 13: Example 3: Tetrahedral mesh shape metric classes: (a) the mesh generated by our method, (b) the mesh generated by Hypermesh.

by Hypermesh. The number of tetrahedral elements generated by Hypermesh is almost 7 times as large as the number of hybrid elements generated by our method.

6.3. A Part of a Large Crankshaft

The third example is a part of the large crankshaft used by ship's diesel engine, as shown in Fig. 12(a). This crankshaft is much bigger than the ordinary crankshaft. The diameter of the main journal is 295. The diameter of the connecting rod journal is 290. The length of the whole crankshaft is 49625. However, the diameter of the oilhole is 12, a very small value. The stress concentration always appears near the oilhole, so it can not be ignored during the CAE analysis. Mesh generated by the presented method contains 56 hexahedral elements, 56 pyramidal elements, 6885 tetrahedral elements, and 1778 mesh nodes. A view of the tetrahedral mesh and pyramidal mesh is depicted in Fig. 12(b). Tab. 1 shows the element shape metrics of the hybrid mesh. Fig. 13(a) shows the shape metric classes of the tetrahedral elements generated by our method. We can see that our method can generate high quality hybrid mesh for complex geometry. Thus, the present method is applicable for generating hybrid mesh for complex solid with small tubular holes.

The mesh generated by Hypermesh contains 4321 mesh nodes and 21496 tetrahedral elements. The total CPU time needed for the mesh generation is 1 seconds. We can see that the number of tetrahedral elements generated by Hypermesh is almost 3 times as large as the number of hybrid elements generated by our method. Fig. 13(b) shows the shape metric classes of the tetrahedral elements generated by Hypermesh. The mesh quality of the tetrahedral mesh generated by our method is still as good as the mesh generated by Hypermesh.

7. CONCLUSIONS

A conforming hybrid mesh generation method for three-dimensional solids with open-ended tubular holes has been presented in this paper. A pyramidal element generation method has been proposed to solve the non-conforming problem of hybrid mesh contains tetrahedral and hexahedral elements. The algorithm according to the arc length rule for the discretization of parametric curve has been applied to obtain the discretization of the central line of the tubular hole. The sweep method has been applied to generate hexahedral elements around these tubular holes. The Delaunay tetrahedral mesh generation method coupled with the advancing front method has been applied to generated tetrahedral mesh of the rest of the domain. Several examples have demonstrated that high-quality hybrid mesh can be automatically created by our method. The robustness and efficiency of the proposed method have been verified through these examples. The amount of elements generated by our method is far less than the all-tetrahedral mesh.

ACKNOWLEDGEMENTS

This work was supported in part by National 973 Project of China under grant number 2010CB328005, and in part by National Science Foundation of China under grant number 10972074.

REFERENCES

[1] Benzley, S.-E.; Perry, E.; Merkley, K.: A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elasto-plastic analysis, In: Proceedings of the 4th international meshing roundtable, 17, 1995, 179–191.

[2] Chand, K.-K.: Component - hybrid mesh generation, International Journal for Numerical

Methods in Engineering, 62(6), 2005, 747–773. DOI:10.1002/nme.1191

[3] Cifuentes, A.-O.; Kalbag, A.: A performance study of tetrahedral and hexahedral elements in 3-D finite element structural analysis, Finite Elements in Analysis and Design, 12(3), 1992, 313-318. DOI:10.1016/0168-874X(92)90040-J

[4] Cuilliere, J.-C.: An adaptive method for the automatic triangulation of 3D parametric surfaces, Computer-Aided Design, 30(2), 1998, 139–149. DOI:10.1016/S0010-4485(97)00085-7

[5] Frey, P.-J.; Borouchaki, H.; George, P.-L.: 3D Delaunay mesh generation coupled with an advancing front approach, Computer Methods in Applied Mechanics and Engineering, 157, 1998, 115–131. DOI:10.1016/S0045-7825(97)00222-3

[6] Guan, Z.-Q.; Shan, J.-L.; Zheng, Y.: An extended advancing front technique for closed surfaces mesh generation, International Journal for Numerical Methods in Engineering, 74(4), 2008, 642–667. DOI:10.1002/nme.2190

[7] Huang, C; Zhang, J.-M.; Qin, X.-Y.; Lu, C.-J.; Sheng, X.-M.; Li, G.-Y.: Stress analysis of solids with open-ended tubular holes by BFM, Engineering Analysis with Boundary Elements, 36(12), 2012, 1908-1916. DOI:10.1016/j.enganabound.2012.07.009

[8] Jan, F.: Advancing front mesh generation technique with application to the finite element method, Department of Structural Mechanics of Chalmers University of Technology, 1994.

[9] Khawaja, A.; Minyard, T.; Kallinderis, Y.: Adaptive hybrid grid methods, Computer Methods in Applied Mechanics and Engineering, 189(4), 2000, 1231–1245. DOI:10.1016/S0045-7825(99)00375-8

[10] Knupp, P.-M.: Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part II—a framework for volume mesh optimization and the condition number of the Jacobian matrix, International Journal for Numerical Methods in Engineering, 48(8), 2000, 1165–1185. DOI:10.1002/(SICI)1097-0207(20000720)48:8<1165::AID-NME940>3.0.CO;2-Y

[11] Mouton, T.; Borouchaki, H.; Bennis, C.: Hybrid mesh generation for reservoir flow simulation: Extension to highly deformed corner point geometry grids, Finite Elements in Analysis and Design, 46(1), 2010, 152–164. DOI:10.1016/j.finel.2009.06.033

[12] Owen, S.-J.; Saigal, S.: Formation of pyramid elements for hexahedra to tetrahedra transitions, Computer methods in Applied Mechanics and Engineering, 190(34), 2001, 4505–4518. DOI:10.1016/S0045-7825(00)00330-3

[13] Qin, X.-Y.; Zhang, J.-M.; Liu, L.-P.; Li, G.-Y.: Steady-state heat conduction analysis of solids with small open-ended tubular holes by BFM, International Journal of Heat and Mass Transfer, 2012. DOI:10.1016/j.ijheatmasstransfer.2012.06.091

[14] Roca, X.; Sarrate, J.; Huerta, A.: Surface mesh projection for hexahedral mesh generation by sweeping, In: Proceedings of the 13th international meshing roundtable, 2004, 169–179.

[15] Wu, M.-C.; Lit, C.-R.: Analysis on machined feature recognition techniques based on B-rep, Computer-Aided Design, 28(8), 1996, 603-616. DOI:10.1016/0010-4485(95)00075-5

[16] Zhang, J.-M.; Qin, X.-Y.; Han, X; Li, G.-Y.: A boundary face method for potential problems in three dimensions, International Journal for Numerical Methods in Engineering, 80(3), 2009, 320–337. DOI:10.1002/nme.2633