Unstructured Tetrahedral Meshing by an Edge-Based Advancing Front Method

Young-Woong Kim, Gi-Whan Kwon, Soo-Won Chae*, Jae-Kyung Shim
Department of Mechanical Engineering, Korea University, Seoul 136-701, Korea

This paper proposes an unstructured tetrahedral meshing algorithm for CAD models in the IGES format. The work presented is based on the advancing front method, which was proposed by the third author. Originally, the advancing front method uses three basic operators, namely, trimming, wedging, and digging. In this research, in addition to the basic operators, three new operators splitting, local finishing, and octahedral-are added to stabilize the meshing process. In addition, improved check processes are applied to obtain better-shaped elements. The algorithm is demonstrated and evaluated by four examples.

Key Words: Tetrahedral Meshing, Advancing Front Method, Basic Operators, Check Processes, CAD Models, Finite Element Analysis

1. Introduction

The use of 3D finite element analysis in CAD environment has greatly increased recently, which requires automatic three-dimensional mesh generation. For a 3D finite element analysis, tetrahedral elements are widely employed due to the availability of many pertinent algorithms. There are many tetrahedral mesh generation algorithms developed so far, and they can be grouped into three major approaches: advancing front methods (Chae and Bathe, 1989; Jin and Tanner, 1993; Moller, 1995; Chae and Lee, 1999; Chae et al. , 2001), Delaunay triangulations (Weatherill and Hassan, 1994; Borouchaki et al. , 1996), and octree-based algorithms (Yerry and Shephard, 1984; Shephard and George, 1991).

In this paper we propose an edge-based advancing front algorithm for unstructured tetrahedral meshing from CAD models in the IGES format. An edge-based algorithm employs basic operators that work on the edges of triangulated surfaces instead of faces. Our previous advancing front method uses three basic operators for mesh generation: trimming, wedging, and digging. One crucial drawback of the advancing front method, however, is the distortion of remaining volumes, which results in the failure of meshing when a large number of elements is to be generated. In order to handle this problem more effectively, the previous advancing front method is modified in this research by developing and adding three new operators. These are local finishing, octahedral, and splitting operators. The octahedral operator is introduced to solve pathological cases, and the existing splitting operator is modified to include multiple meshing domains.

Out of the two types of data structures in the IGES format such as CSG and B-Rep, B-Rep is adopted in our mesh generation because our advancing front technique starts from surfaces.

2. Geometry Input for Mesh Generation from CAD Models

In the developed system, the geometry information in the IGES format (Reed et al., 1996) is employed in mesh generation. The IGES format is composed of the geometry and topology

* Corresponding Author,
E-mail: swchae@Korea.ac.kr
TEL: +82-2-3290-3367; **FAX**: +82-2-926-9441
Department of Mechanical Engineering, Korea University, Seoul 136-701, Korea. (Manuscript **Received** August 23, 2001; **Revised** November 16, 2001)

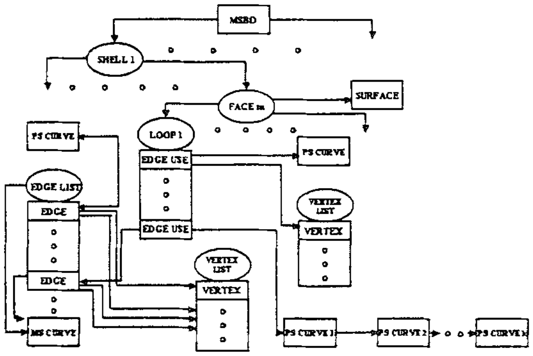


Fig. 1 Construction of the MSBO in IGES(Reed, 1996)

information. Geometry information is obtained from the IGES entity No. 126 (Rational B-Spline Curve) and No. 128 (Rational B-Spline Surface), while topology information is obtained from the IGES entity No. 186 (Manifold Solid B-Rep Object), No. 514 (Closed Shell), No. 510 (Face), No. 508 (Loop), No. 504 (Edge List), and No. 502 (Vertex List). Figure 1 illustrates the hierarchical nature of the IGES representation. The Manifold Solid B-Rep Object (MSBO) defines a manifold solid by enumerating its boundary. This boundary may be decomposed into closed shells. Each shell is composed of faces which have underlying surface geometry. The faces are bounded by loops of edges having underlying curve geometry. The edges are bounded by vertices whose underlying geometry is a set of points. Each edge is used once in each orientation and therefore shall be referenced exactly twice in an MSBO. The geometric entities that are used in an MSBO consist of points, curves, and surfaces. The geometric surface definition used to specify the geometry of a face shall be a 2-manifold which is arcwise connected, oriented, bounded, non-self-intersecting, and has no handles within the region underlying the face.

The data structures in this mesh generation system are composed of two parts as shown in Fig. 2. One is for the geometry and topology information obtained from the input file, which is similar to the MSBO structure in the IGES format. The other is for the mesh information such as triangular and tetrahedral meshes

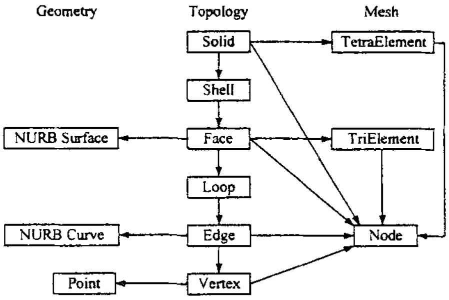


Fig. 2 Data structure

generated at every stage of mesh generation.

3. Triangular Mesh Generation on Trimmed NURBS Surfaces

The employed surface triangulation scheme consists of the following three steps (Chae and Kwon, 2001). First, three-dimensional surfaces with generated key nodes are transformed into two-dimensional planes. For this, by considering the geometry of surfaces, appropriate 2D transformation planes among projection planes, quasi-expanded planes, and parametric planes are used. Then, triangular elements are constructed on these planes by using a domain decomposition algorithm. Finally, the constructed meshes in two-dimensional planes are transformed back to the original surfaces. As far as finite element modeling is concerned, quadratic elements can usually describe the geometry very well in addition to its effectiveness in computational accuracy. So 10-noded tetrahedral elements are preferred to 4-noded tetrahedral elements in practice. Our algorithm basically generates 10-noded quadratic tetrahedral elements and these elements can be easily transformed to 4-noded tetrahedral elements if necessary. For this purpose, the surface triangulation algorithm employed should be able to generate 6-noded triangular elements that describe the curved surface geometry very well. Since most CAD systems provide surface information as NURBS surfaces, in order to exchange data with different systems, trimmed NURBS surfaces in the IGES format are employed in this paper.

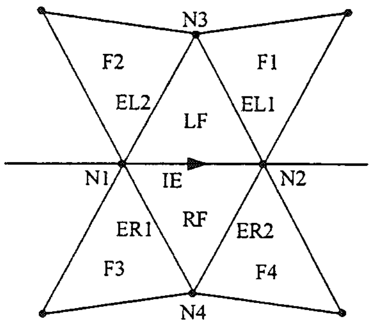


Fig. 3 Expanded view around an edge IE

4. Tetrahedral Mesh Generation

After the surfaces of a 3D object under consideration have been triangulated, volume triangulation then starts from the outside surfaces and proceeds toward the inside by cutting the sharp corner edges using the following basic operators.

4.1 Basic operators

Chae and Bathe (1989) showed that at least three basic operators-trimming, wedging, and digging-are needed to satisfy topological requirements for tetrahedral mesh generation in the advancing front method. In this research, three new operators, called the splitting, local finishing, and octahedral operators, are added to the basic operators in order to stabilize the meshing process.

For the description of the basic operators, the following definitions are employed. Figure 3 shows an expanded view of triangular meshes generated on a surface. An edge IE under consideration has two adjacent faces, a left face (LF) and a right face (RF), and four surrounding edges, EL1, EL2, ER1, and ER2. The adjacent faces of these surrounding edges are F1, F2, F3 and F4, respectively, and they are called the surrounding faces to an edge IE. The edge angle of IE, in our scheme, is defined as a dihedral angle formed by the two adjacent faces, LF and RF.

4.1.1 Trimming operator

Topologically, a trimming corner edge is an edge for which two surrounding faces are identical as shown in the left of Fig. 4. This situation can be characterized by the fact that only three edges meet at node N2. The trimming operator on a trimming edge generates one tetrahedron by removing three faces, three edges and one key node from a loop-boundary and generates one new face as shown in the right of Fig. 4. Hence, this operator reduces the number of loop-boundary edges (E) by 3, faces (F) by 2, and key nodes (N) by 1. That is, E=3, F=2, V=1.

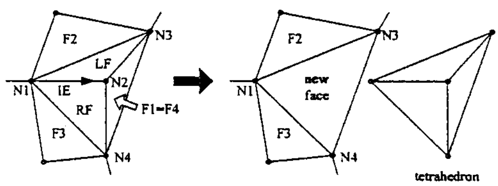


Fig. 4 Trimming operator

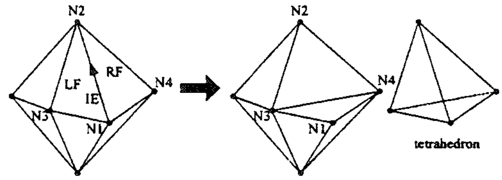


Fig. 5 Wedging opeator

4.1.2 Wedging operator

The wedging operator is designed to generate one tetrahedral element at a wedging corner edge as shown in Fig. 5. A wedging corner edge is an edge for which all surrounding faces are different from each other, hence four or more edges meet at both of the end nodes. With this operator, one edge and two faces are removed, and one new edge and two new faces are introduced (E=0, F=0, V=0).

4.1.3 Digging operator

A digging edge is defined as an edge that has an edge angle less than 175° and is neither a trimming nor wedging edge. The digging operator is designed to generate two tetrahedral elements at a digging corner edge by introducing a new key node as shown in Fig. 6. With this operator, one edge and two faces are removed and four new

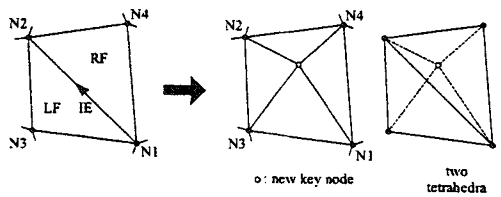


Fig. 6 Digging operator

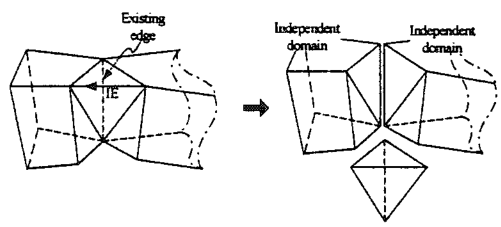


Fig. 7 Splitting operator

edges, four new faces, and one new key node are introduced (E=+3, F=+2, V=+1).

4.1.4 Splitting operator

The previous version of the splitting operator was proposed to change a multi-connected domain into a single-connected domain (Chae and Lee 1999). In this research, the previous splitting operator is modified to be applicable several times in order to divide an initial multi-connected domain, such as an object with holes, into several separated objects. The splitting operator works on an edge that has similar geometrical and topological conditions with a wedging edge, and is designed to construct one tetrahedron by cutting the corresponding edge. In this case, the new edge actually coincides with an existing edge as shown in Fig. 7, and two different edges are employed instead of one edge. With this operator, one edge and two faces are removed and one new edge and two new faces are introduced (E=0, F=0, V=0).

4.1.5 Local finishing operator

The local finishing operator is designed to generate one tetrahedron when a local remaining domain has four faces and six edges as shown in Fig. 8. During the meshing process, an object is divided into several objects, and each object may finally end up with this operator.

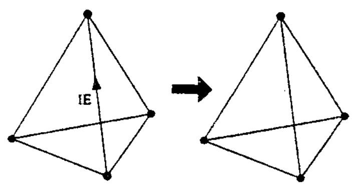


Fig. 8 Local finishing operator

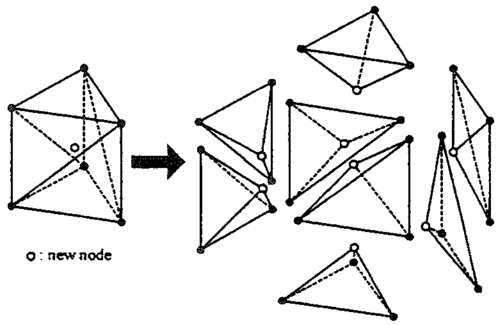
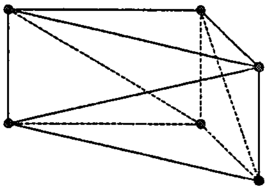


Fig. 9 Octahedral operator

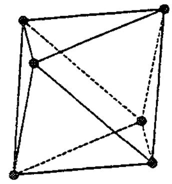
4.1.6 Octahedral operator

The octahedral operator shown in Fig. 9 is designed to solve the pathological case of octahedrons, which is also known as an untetrahedralizable Sch nhardt prism (Bern and Eppstein, 1992). Previous studies attempted to solve this problem by employing the digging operator (Chae and Bathe, 1989) that introduces a new node inside or by adding a Steiner point that is visible to all points (Bajaj et al., 1999).

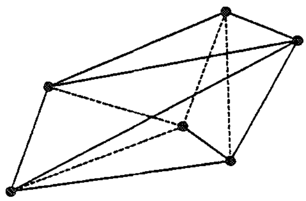
Our new operator, the octahedral operator, is developed based on the following observations. Topologically, a pathological case shown in Fig. 10(a) is the same with the case of an equilateral octahedron of Fig. 10(b). In this case, whether an object can be meshed or not depends on the following geometric condition. If the dihedral angle on the opposite side of an edge under consideration is much less than 180°, it can be meshed by the wedging operator without any difficulty. But if the angle approaches 180°, it will produce very slivering elements. If it becomes 180 °, it will end up with a pathological case. In the case of a twisted pathological case shown in Fig. 10(c), the meshing process can be more complicated if only the digging operator is employed. According to the above-mentioned observation,



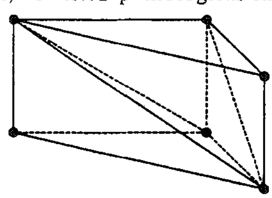
(a) Pathological case



(b) Pathologicalcase : equilateral octahedron



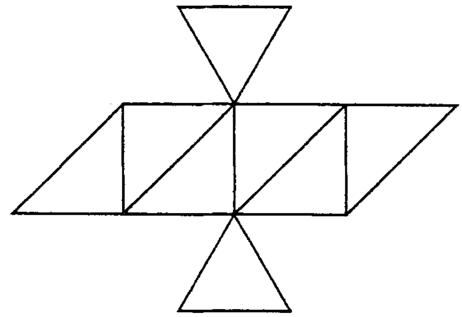
(c) Twisted pathological case



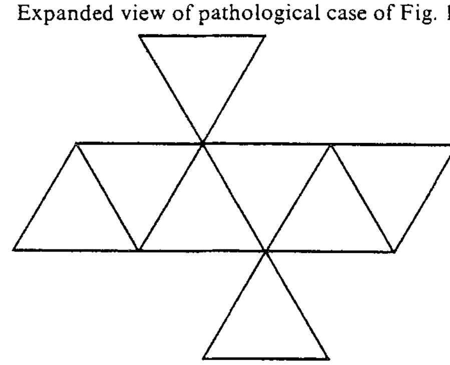
(d) Non-pathological case

Fig. 10 Octahedral cases

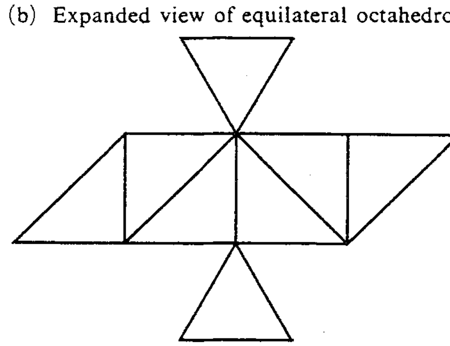
we may conclude that a mesh generation algorithm based on the geometric condition of an octahedron would be quite complex and might not solve pathological cases. On the other hand, topological identification of the pathological case is much simpler. Topologically, octahedrons can be classified into pathological and non-pathological cases as shown in Fig. 10. In the pathological case, each node is connected to four different edges as shown in Figs. 11(a) and (b). In the non-pathological case, however, three nodes are connected to five different edges and the other three nodes to three different edges as shown in Fig. 11(c), and this case can be meshed by applying the trimming operator. For this reason, the proposed octahedral operator only considers the topological condition of an octahedron, that is, whether it belongs to a pathological case or not. As for the pathological case, the octahedral operator generates eight tetrahedral elements by introducing a new node and six new edges inside an octahedron.



(a) Expanded view of pathological case of Fig. 10(a)



(b) Expanded view of equilateral octahedron

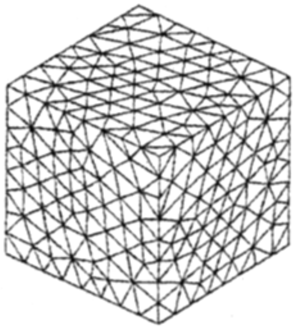


(c) Expanded view of non-pathological case

Fig. 11 Expanded views

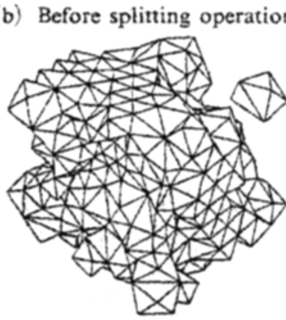
4.2 Check processes

Since the advancing front method starts from the outside of a domain and proceeds toward the inside, the remaining domain frequently becomes too distorted to be meshed. In order to avoid distortions, check processes are usually employed. In our scheme, three types of check processes are employed. Namely, in addition to the two conventional methods called the overlap check and the minimum distance check between new nodes and faces (Chae and Bathe, 1989), an edge-based distance check is added to stabilize our meshing



(a) An initial surface mesh

(b) Before splitting operations



(c) After splitting operations

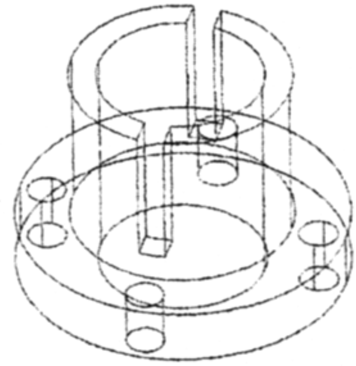
(d) Multiple domains

Fig. 12 Mesh generation process

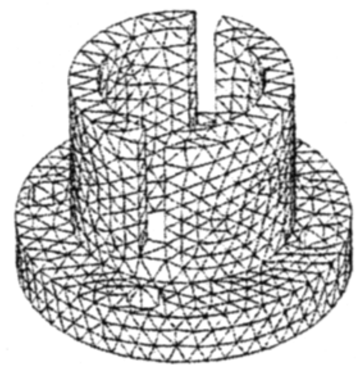
process. An edge-based distance check is employed when the wedging or digging operators are to be performed, in which new edges are tested as to whether they have enough distances from the existing edges (Cho, 2000).

4.3 Mesh generation process

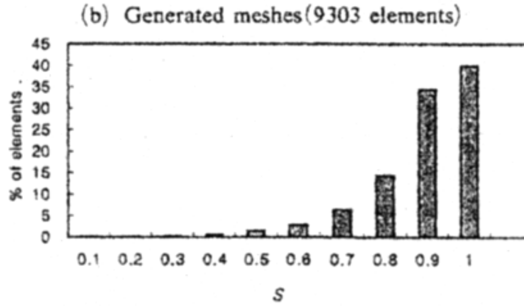
The proposed mesh generation process is shown in Fig. 12. Figure 12(a) shows an initial surface mesh, and Figs. 12(b) and (c) show intermediate processes before and after the splitting operations. Figure 12(d) shows multiple domains during the meshing process.



(a) Input model



(b) Generated meshes (9303 elements)

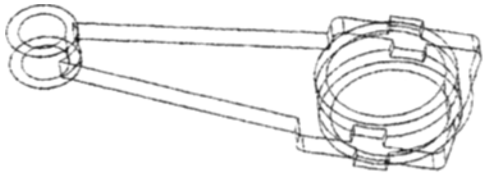


(c) Element quality histogram

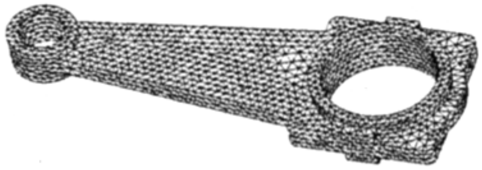
Fig. 13 Cylindrical object with multiple holes

5. Examples

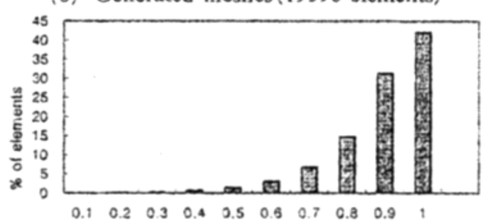
In this section, some examples of finite element mesh generation on three-dimensional objects are given. Figure 13 shows an example of a



(a) Input model

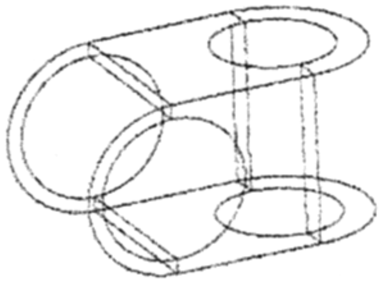


(b) Generated meshes (19596 elements)

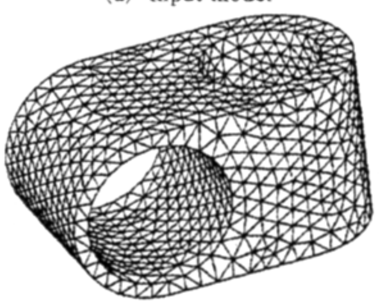


(c) Element quality histogram

Fig. 14 Connecting rod



(a) Input model



(b) Generated meshes (7406 elements)

Fig. 15 Object with holes

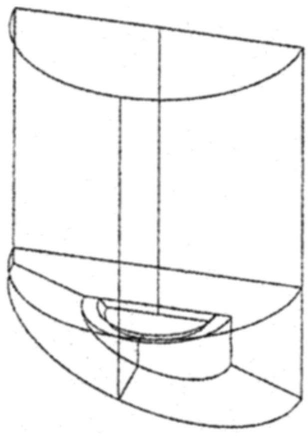
cylindrical object with multiple holes. The input model has 24 faces and 198 curves in the IGES format, and 9303 tetrahedral elements are generated. In order to evaluate the quality of generated meshes, the histogram of the following ratio, S, is plotted.

$$S=3\ r/R \tag{1}$$

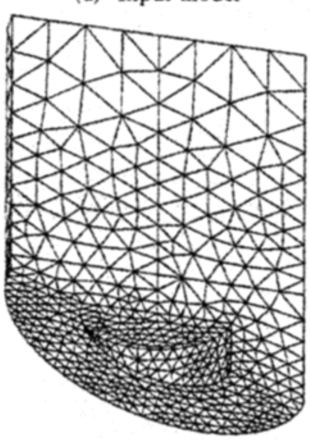
where r and R represent the radii of the inscribed and circumscribed spheres of a tetrahedral element, respectively. Thus, S becomes unity for an equilateral tetrahedron. As shown in Fig. 13(c), most of the generated elements are well shaped, since most of them have the ratio, S, ranging between 0.8 and 1.0. Another example shown in Fig. 14 is a connecting rod model with 38 faces and 324 curves in the IGES format, and 19596 tetrahedral elements are generated. The histogram shown in Fig. 14(c) shows that the generated meshes are of good quality. Figure 15 shows another example of an object with holes, and 7406 elements are generated. Figure 16 shows a die model for metal forming, for which 6728 elements are generated. This figure shows that relatively small elements are generated at the die contact area.

6. Conclusions

An automatic unstructured tetrahedral mesh generation scheme for three-dimensional objects with trimmed NURBS surfaces has been developed. This edge-based advancing front method utilizes six operators and three check processes in total for the mesh generation. Among the three developed operators, the octahedral operator successfully solves the pathological case of octahedrons, and the splitting operator is able to deal with multi-connected domains such as an object with holes. The demonstrated examples show that the proposed algorithm generates different sized meshes in various objects automatically. It has also been shown that the algorithm generates reasonably well-shaped meshes by evaluating the ratios of the inscribed to the circumscribed sphere radii of the constructed tetrahedral elements.



(a) Input model



(b) Generated meshes(6728 elements)

Fig. 16 Forging die

References

Bajaj, C. L., Coyle, E. J., and Lin, K., 1999, Tetrahedral Meshes from Planar Cross-Sections, *Computer Methods in Applied Mechanics and Engineering*, Vol. 179, No. 1-2, pp. 31~52.

Bern, M. and Eppstein, D., 1992, Mesh Generation and Optimal Trianglulation, *Computing in Euclidean Geometry*, edited by Du, D. Z. and Hwang, F. K., World Scientific, pp. 23~90.

Borouchaki, H., George, P. L., and Lo, S. H., 1996, Optimal Delaunay Point Insertion, *Int. J. Numer. Meth. Engng.*, Vol. 39, pp. 3407~3437.

Chae, S. W and Bathe, K. J., 1989, On automatic mesh construction and mesh refinement in finite element analysis, *J. Computers & Structures*, Vol. 32, No. 3/4, pp. 911~936.

Chae, S. W., Kim, Y. W., and Cho, Y. W., 2001, Tetrahedral meshing by an advancing front method through a CAD interface, *Proc. of the First MIT Conference on Computational Fluid and Solid Mechanics*, pp. 1535~1539.

Chae, S. W. and Kwon, K. Y., 2001, Quadrilateral Mesh Generation on Trimmed NURBS Surfaces, *KSME International Journal*, Vol. 15, No. 5, pp. 592~601.

Chae, S. W. and Lee, G. M., 1999, Volume Traingulation from Planar Cross Sections, *J. Computers & Structures*, Vol. 72, pp. 93~100.

Cho, Y. W., 2000, Automatic Tetrahedral Mesh Generation by Advancing Front Technique, Korea Univ., M. S. Thesis in Korea.

Jin, J. and Tanner, R. I., 1993, Generation of Unstructured Tetrahedral Meshes by Advancing Front Technique, *Int. J. Numer. Meth. Engng.*, Vol. 36, pp. 1805~1823.

Moller, P., 1995, On Advancing Front Mesh Generation in Three Dimensions, *Int. J. Numer. Meth. Engng.*, Vol. 38, pp. 3551~3519.

Reed, K., Harrod, D., and Conry, W., 1996, "The Initial Graphics Exchange Specification (IGES) Version 5.3," *NIST Report, USA.*

Shephard, M. S. and George, M. K., 1991, Automatic Three-Dimensional Mesh Generation by the Finite Octree Technique, *Int. J. Numer. Meth. Engng.*, Vol. 32, pp. 709~749.

Weatherill, N. P. and Hassan, O., 1994, Efficient Three-Dimensional Delaunay Triangulation with Automatic Point Creation and Imposed Boundary Conditions, *Int. J. Numer. Meth. Engng.*, Vol. 37, pp. 2005~2039.

Yerry, M. A. and Shephard, M. S., 1984, Automatic Three-Dimensional Mesh Generation by the Modified Octree Technique, *Int. J. Numer. Meth. Engng.*, Vol. 20, pp. 1965~1990.