한국 CAD/CAM 학회 논문집
제 1 권 제 1 호 1996년 3월 pp. 65-75

자유 곡면으로 구성되는 3차원 구조물에 대한 자동 요소 분할

이준성* · 矢川 元基** · 박면웅***

Automatic Mesh Generation for Three-Dimensional Structures Consisting of Free-Form Surfaces

Joon-Seong Lee*, Genki Yagawa**, Myon-Woong Park***

Abstract

This paper describes an automatic finite element(FE) mesh generation for three-dimensional structures consisting of free-form surfaces. This mesh generation process consists of three subprocesses: (a) definition of geometric model, i.e. analysis model, (b) generation of nodes, and (c) generation of elements. One of commercial solid modelers is employed for three-dimensional solid and shell structures. Node is generated if its distance from existing node points is similar to the node spacing function at the point. The node spacing function is well controlled by the fuzzy knowledge processing. The Delaunay method is introduced as a basic tool for element generation. Automatic generation of FE meshes for three-dimensional solid and shell structures holds great benefits for analyses. Practical performances of the present system are demonstrated through several mesh generations for three-dimensional complex geometry.

Key words : Automatic mesh generation, Fuzzy theory, Bucketing method, Delaunay triangulation, Free-form surface, Solid geometry, Finite element analysis

1. Introduction

The finite element method(FEM) has been widely utilized in simulating various engineering problems such as structural deformation, thermal conduction, fluid dynamics, electromagnetics and so on. The main reason for this is its high capability of dealing with boundary-value problems in arbitrarily shaped domains. On the other hand, a mesh used influences computational accuracy as well as time so significantly that the mesh generation process is as much important as the FEM analysis itself. Especially, in such large scale nonlinear FEM analyses that approach the limitation of computational capability of so-called supercomputers, it is highly demanded to optimize the distribution of mesh size under the condition of limited total degrees of freedom(DOFs). Thus, the mesh generation process becomes more and more time-consuming and heavier tasks.

Loads for pre-processing and post-processing are increasing rapidly in accordance with an increase of scale and complexity of analysis models to be solved. Particularly, the mesh generation process, which influences computational accuracy as efficiency and whose fully automation is very difficult in three-dimensional cases, has become the most critical issue in a whole process of the FE analyses. In this respect, various researches[1-13] have been performed on the development of automatic mesh generation techniques.

*종신회원, 경기대학교 기계공학과
**동경대학 공학부
***KIST 기전연구부

Among mesh generation methods, the tree model method[7, 8] can generate graded meshes and it uses a reasonably small amount of computer time and storage. However, it is, by nature, not possible to arbitrarily control the changing rate of mesh size with respect to location, so that some smaller projection and notch etc. are sometimes omitted. Also, domain decomposition method[9, 10] does not always succeed, and a designation of such subdomains is very tedious for uses in three-dimensional cases.

In recent years, much attention has been paid to fuzzy knowledge processing techniques[13], which allow computers to treat "ambiguous" matters and processes. In this paper, we explain an FE mesh generation system based on fuzzy knowledge processing and computational geometry techniques. Here, the node density distribution, which is a kind of a node spacing function, was well controlled by means of the fuzzy knowledge processing technique, so that even beginners of the FE analyses are able to produce nearly optimum meshes through very simple operations as if they were experts.

The individual techniques in the present study are as follows:

(a) Adoption of practical geometric modelers such as Designbase[15] which are capable of dealing with Bezier-type free-form surfaces.
(b) Adoption of the bucketing method [16] for fast node generation, which is one of computational geometry techniques.
(c) Adoption of the Delaunay triangulation method[1-3] for fast element generation.

The present mesh generation systems are constructed in one of popular engineering workstations (EWS) using the C and C++ language under the Unix environment.

In the following sections, the fundamental principle of the present algorithm is described. Finally, practical performances of the developed systems are demonstrated through the mesh generation of several complex geometries.

2. General requirement for automatic mesh generation systems

The phase of pre-processing is very important in the sense that the generation of a valid mesh in a domain with a complex geometry is not a trivial operation and can be very expensive in terms of the time required. On the other hand, it is crucial to create a mesh which is well adapted to the physical properties of the problem under consideration, as the quality of the computed solution is strongly related to the quality of the mesh.

Various automatic and semi-automatic mesh generation methods have been investigated so far. The requirements for ideal fully automatic mesh generation systems may be summarized as follows[17]:

(a) Arbitrarily shaped domain can be subdivided into elements.
(b) Mesh size and its changing rate with respect to location can be easily controlled.
(c) Distortion of element shape can be avoided as much as possible.
(d) Total number of nodes can be controlled.
(e) Number of input data is smaller.

The item(a) is fundamental, while(b) and (c) are strongly related to mesh quality. The item(d) corresponds to the controllability of computational time and storage. If any system satisfies the items (a) through (d), optimum meshes can be generated with the balance of computational accuracy as well as efficiency. The item (e) is also indispensable for any systems dealing with three-dimensional complex geometries.

3. Outline of the system

3.1 Definition of geometric model

Geometric modelers are utilized to define geometries of analysis domains. One of commercial geometric modelers, Designbase Ver. 3[15] is employed for three-dimensional solid structures and shell structures. The advantage of Designbase is that a wide range of solid shapes from polyhedra to free-form surfaces can be designed in a unified manner. But for shell structures, a specialized selection for shell structures -firstly, define the three-dimensional solid structures, and selection of surface the shell structures as shown in

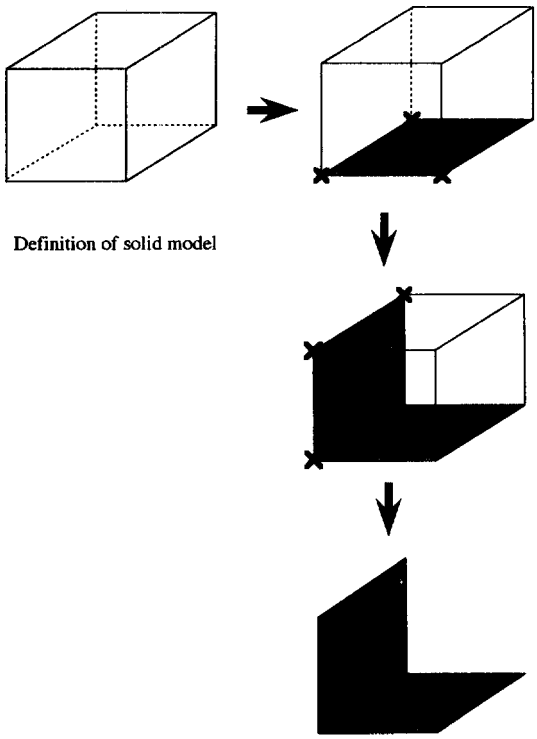


Fig. 1. Definition of three-dimensional shell model.

Fig. 1 is employed. This method is adopted to think over the analyzing compound solid and shell structures. In these modelers, three-dimensional geometric data are stored as a tree structure of domain-surfaces (free-form surfaces such as Bezier or Gregory type surfaces) - edge(B-spline or Bezier type curves) - vertices.

Designbase allows the user to start with a hierarchical compositional Constructive Solid Geometry (CSG)-view of a part and then to refine it with local but consistent operations on the boundary representation of the object. By basing all operations available to the user on well-defined, invertible Euler-operations, it is possible to keep a compact representation of the complete design history of a part, and thus to "undo" and "redo" any sequence of operations. This encourages the designer to try out ideas without fear of destroying a model in which several hours of design time have already been invested. It also makes it possible to store several alternative versions of a design in a natural and efficient manner. As an example, Figs. 2(a) and 2(b) show a geometry

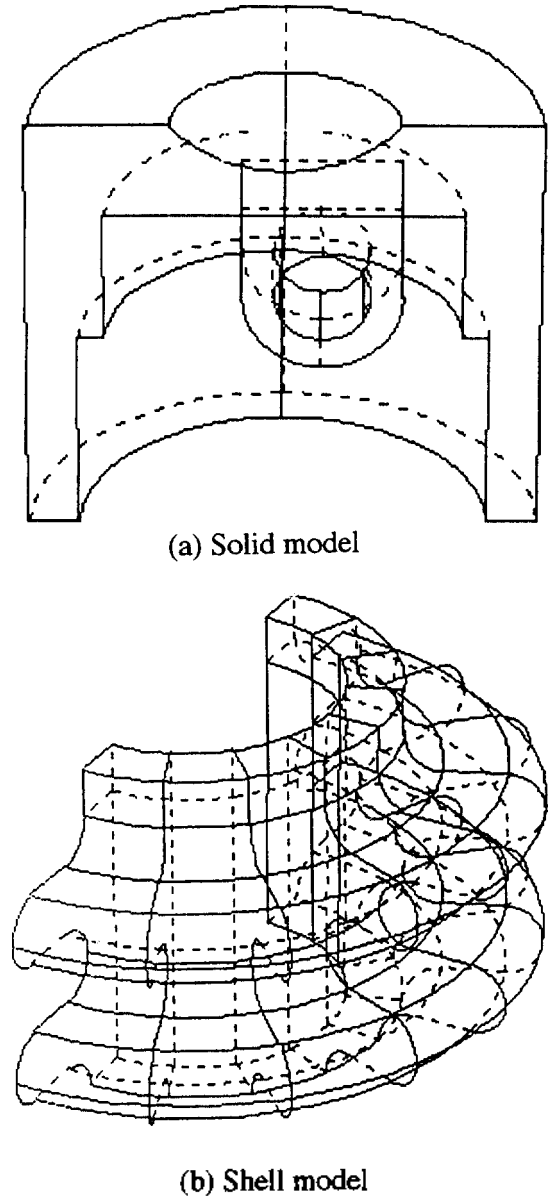


Fig. 2. Examples of geometry model.

model of three-dimensional solid and shell structures using Designbase.

3.2 Designation of node density distributions

In this section, the connecting process of locally-optimum mesh images is dealt with using the fuzzy knowledge processing technique[17, 18]

Performances of automatic mesh generation methods based on node generation algorithms depend

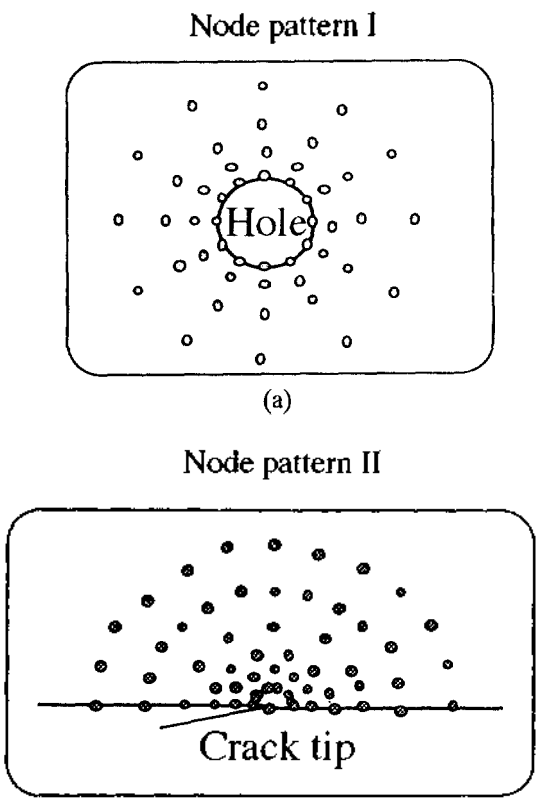


Fig. 3. Examples of local node patterns. (a) For a hole (b) For a crack tip

on how to control node spacing functions or node density distributions and how to generate nodes. The basic concept of the present mesh generation algorithm is originated from the imitation of mesh generation processes by human experts on FE analyses. One of the aims of this algorithm is to transfer such experts' techniques to beginners.

In the present system, nodes are first generated, and then a finite element mesh is built. In general, it is not so easy to well control element size for a complex geometry. A node density distribution over a whole geometry model is constructed as follows. The present system stores several local nodal patterns such as the pattern suitable to well capture stress concentration, the pattern to subdivide a finite domain uniformly, and the pattern to subdivide a whole domain uniformly. A user selects some of those local nodal patterns, depending on their analysis purposes, and designates where to locate them.

For example, when either the crack of the hole exists solely in an infinite domain, the local node patterns as shown in Figs. 3(a) and 3(b) may be regarded locally-optimum around the crack tip or the hole, respectively.

When these stress concentration fields exist closely to each other in the same analysis domain, a simple superposition of both local node patterns gives the result as shown in Fig. 4(b). Namely, extra nodes have to be removed from the superposed region of both patterns.

In the present method, the field A close to the hole and the field B close to the crack-tip are defined in terms of the membership functions used in the fuzzy set theory as shown in Fig. 4(c). For the purpose of simplicity, each membership function is given a function of one-dimension in the figure. In practice the membership function can be expressed as $\mu(x, y)$ in this particular example, and in three-dimensional cases it is a function of three-dimensional coordinates, i.e. $\mu(x, y, z)$. In Fig. 4(c), the horizontal axis denotes the location, while the vertical axis does the value of membership function, which indicates the magnitude of "loseness" of the location to each stress concentration field. That is, a nodal location closer to the stress concentration field takes a larger value of the membership function. As for Figs. 4(b) and 4(c), choosing the mesh pattern with a larger value of the membership function in each location, one can obtain an overlapped curve of both membership functions, and the domain can be automatically divided into the following two sub-domains A and B as shown in Fig. 4(d): the sub-domain close to the crack-tip and that of the hole. Finally, both node patterns are smoothly connected as shown in Fig. 4(e). This procedure of node generation, i.e. the connection procedure of both node patterns, is summarized as follows:

If $\mu_A(x_p, y_p) \geq \mu_B(x_p, y_p)$ for a node $p(x_p, y_p)$ belonging to the pattern A, then the node p is generated, and otherwise p is not generated.

If $\mu_B(x_q, y_q) \geq \mu_A(x_q, y_q)$ for a node $q(x_q, y_q)$ belonging to the pattern B, then the node q is generated, and otherwise q is not generated.

It is apparent that the above algorithm can be easi-

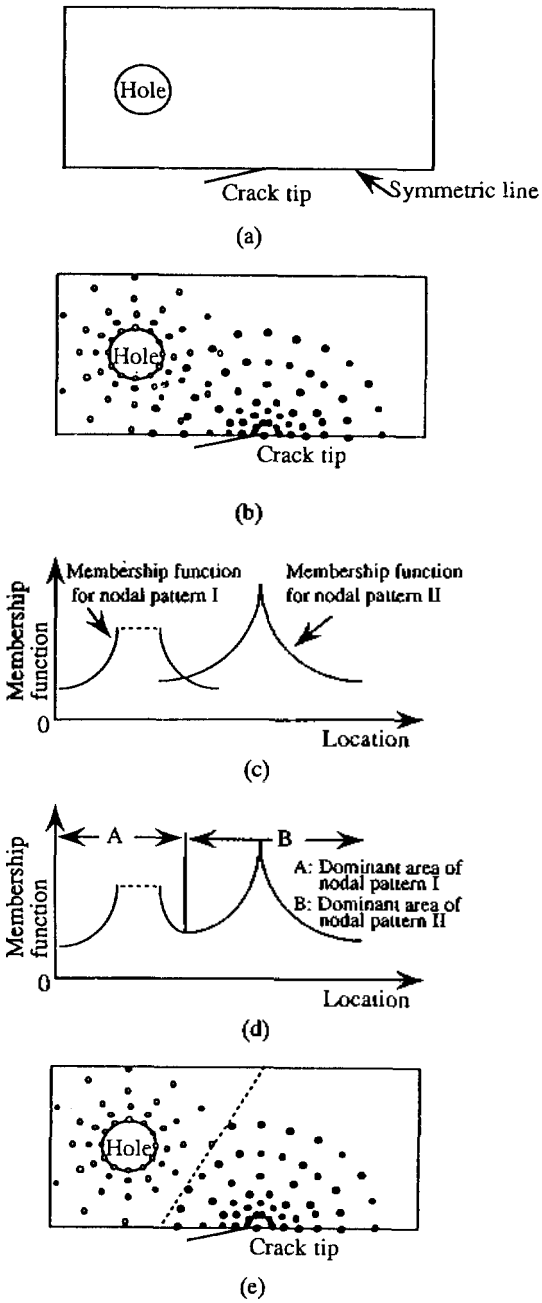


Fig. 4. Superposition of node patterns based on fuzzy theory.

ly extended to three-dimensional problems and any number of node patterns. In addition, since finer node patterns are generally required to place near stress concentration sources, it is convenient to let the membership function correspond to node density as well. According to this definition, Fig. 4(d) also indicates the distribution of node density over the whole analysis domain including the two stress concentration fields.

When designers do not want any special meshing, they can adopt uniformly subdivided mesh. It is possible to combine the present techniques with an adaptive meshing technique.

3.3 Node generation

Node generation is one of time consuming processes in automatic mesh generation. In the present study, the bucketing method[16] is adopted to generate nodes which satisfy the distribution of node density over a whole analysis domain. Fig. 5 shows its fundamental principle, taking the previous two-di-

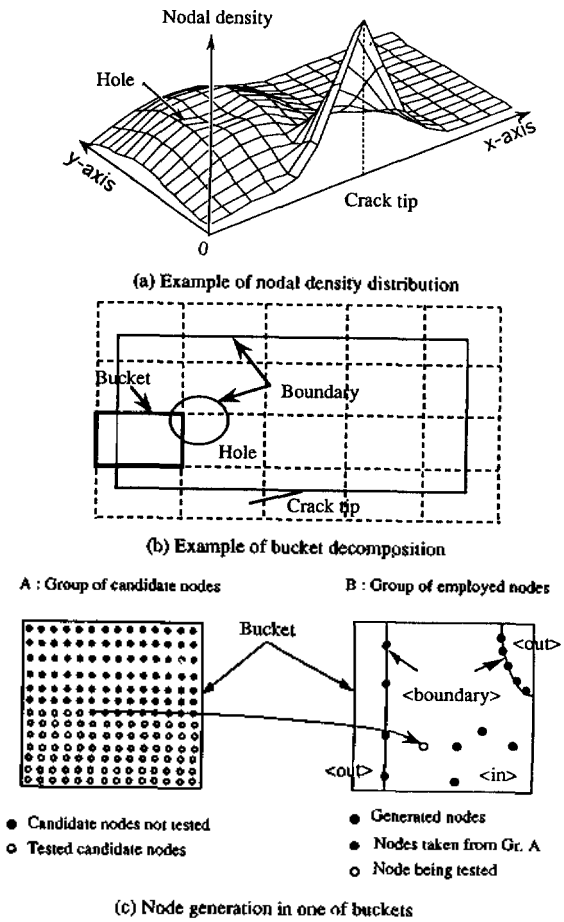


Fig. 5. Node generation based on bucketing method.

mensional mesh generation as an example without any loss of generality. Let us assume that the distribution of node density over a whole analysis domain is already given as shown in Fig. 5(a). At first, a super-rectangle enveloping the analysis domain is defined as shown in Fig. 5(b). In the three-dimensional solid case, a super hexahedron is utilized to envelop an analysis domain. Next, the super-rectangle is divided into a number of small sub-rectangles, each of which is named "Bucket". Nodes are generated bucket by bucket.

At first, a number of candidate nodes with uniform spacing are prepared in one of buckets as shown in Fig. 5(c). The distance of two neighboring candidate nodes is set to be smaller than the minimum distance of nodes to be generated in the relevant bucket. Next, candidate nodes are pick up one by one, starting from the left-bottom corner of the bucket, and are put into the bucket. A candidate node is adopted as one of the final nodes when it satisfies the following two criteria:

(a) The candidate node is inside the analysis domain(IN/OUT check).

(b) The distance between the candidate node and the nearest node already generated in the bucket satisfies the node density at the point to some extent.

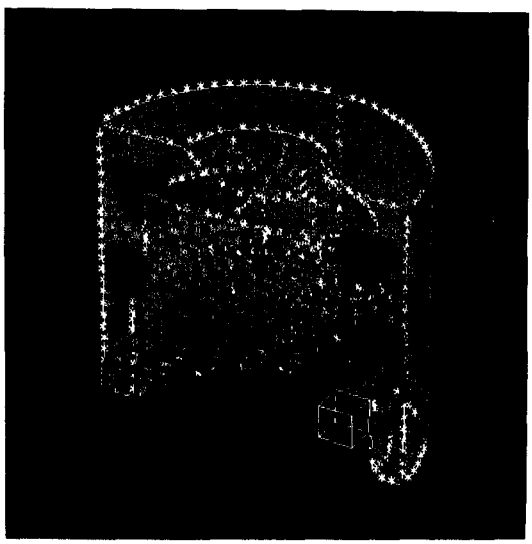


Fig. 6. Generated appearance of nodes.

Practically, the criterion (a) is first examined bucket by bucket. As for buckets lying across the domain boundary, the criterion (a) is examined node by node. It should be noted here that the nodes already generated in the neighboring buckets have to be examined for the criterion (b) as well when a candidate node is possibly generated near the border of the relevant bucket. Thanks to the bucketing method, the number of examinations of the criterion (b) can be

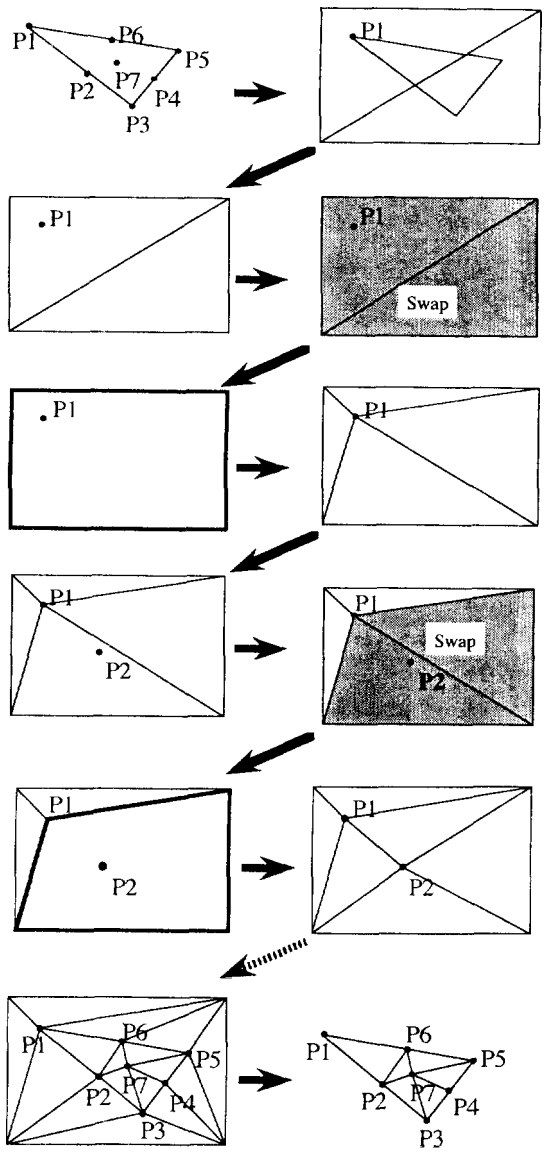


Fig. 7. Delaunay triangulation produced by the iterative algorithm.

reduced significantly, and then a node generation speed is remained to be proportional to the total number of nodes. As for three-dimensional solid geometries, nodes are generated in the following order : vertices, edges, surfaces and domain. Fig. 6 shows the generated appearance of nodes for a half of piston head.

3.4 Element generation

The Delaunay triangulation method[1, 3] is utilized to generate tetrahedral elements from numerous nodes given in a geometry. After all the nodes are placed in the analysis domain and on its boundary, the system creates tetrahedral elements. As an example, the algorithm of triangulation in the two-dimensional case used here is depicted in Fig. 7. Let N be a set of nodes, it has the property that the circumcircle of any triangle in the triangulation contains no point of N in its interior. The remaining points in

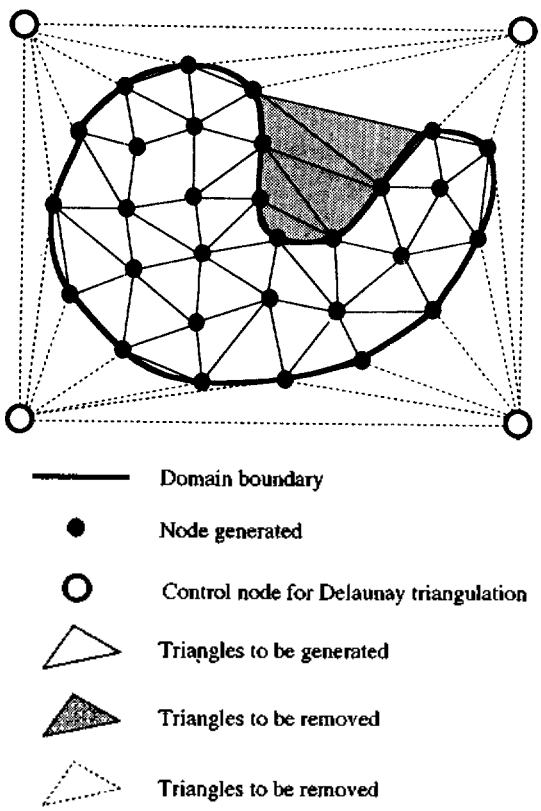


Fig. 8. Techniques of avoiding mis-match elements in Delaunay triangulation (a) Mis-match elements produced in indented region.

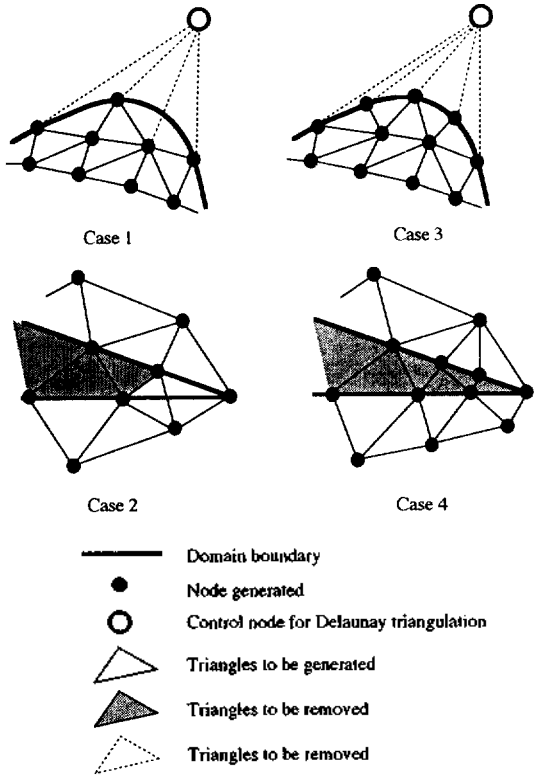


Fig. 8. Techniques of avoiding mis-match elements in Delaunay triangulation (b) Mis-match elements produced near boundary.

N will be iteratively added to the triangulation. After each point is added, it will be connected to the vertices of its enclosing triangle. All internal edges of a triangulation of a finite set N are locally optimal if no point of N is interior to any circumcircle of a triangle. In a three-dimensional domain, tetrahedral elements can be generated by the similar algorithm.

The speed of element generation by the Delaunay triangulation method is proportional to the number of nodes. If this method is utilized to generate elements in a geometry with indented shape, elements are inevitably generated even outside the geometry as shown in Fig. 8(a). However, such mis-match elements can be removed by performing the IN / OUT check for gravity center points of such elements. In addition, it is necessary to avoid the generation of those mismatch elements crossing domain boundary by setting node densities on edges to be slightly higher than those inside the domain near the boundaries as

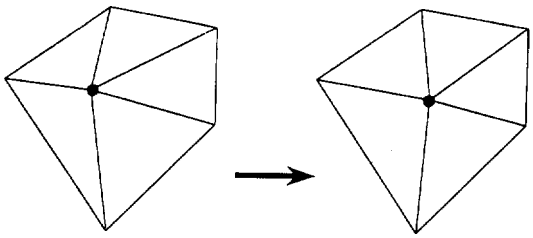


Fig. 9. Smoothing operation.

shown in Fig. 8(b).

3.5 Smoothing operation

The algorithm of element generation mentioned above works well in most cases. However, element shapes obtained are sometimes distorted in a superposed region of several node patterns or near domain boundary. The smoothing method called "Laplacian operation"[19] is here applied to remedy such distorted elements as shown in Fig. 9. In this operation, the location of each node is replaced with a mean value of locations of its neighboring nodes. This operation is iterated several times.

3.6 Additional techniques for 3-D shell geometries

The algorithms described in section 3.2 through 3.5 are sufficient for mesh generation of three-dimensional solid geometries. However, some additional techniques are necessary for mesh generation of three-dimensional shell geometries.

Using the Delaunay triangulation technique, one can generate triangular elements for shell geometry, which was selected surfaces in three-dimensional shell cases, while tetrahedral elements in three-dimensional solid cases. In this paper, we adopted the following technique to generate quadrilateral elements.

Let us assume that an analysis domain is completely divided into a number of triangulars. Two neighboring triangulars are converted into a single quadrilateral. After this operation, a few triangulars still exist. Then, we divide a triangular into three quadrilaterals, and divide a quadrilateral into for smaller quadrilaterals as shown in Fig. 10. Finally, we obtain a complete quadrilateral mesh.

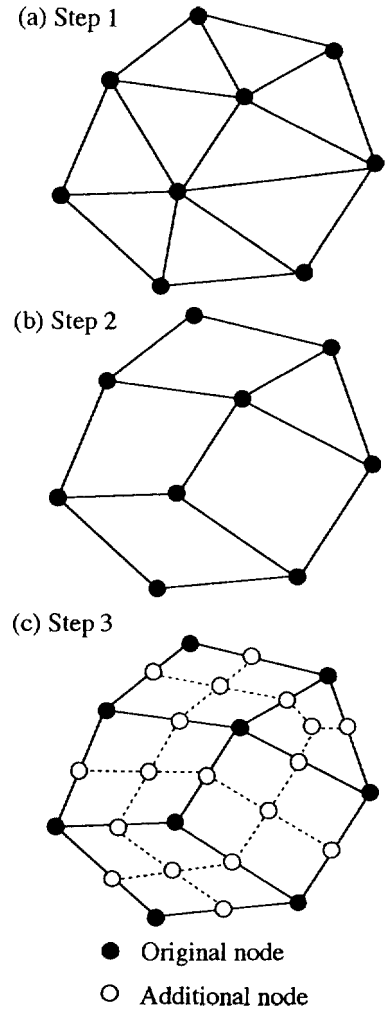


Fig. 10. Technique of converting triangles into quadrilaterals.

4. Examples and discussions

The performance of the system is demonstrated through the mesh generation of several three-dimensional structures. Fig. 11 to 14 show the examples of the application of this mesh generator for three-dimensional geometry. As shown in figures, a uniform mesh and a nonuniform mesh were connected very smoothly. In case of a half of piston head as shown in Fig. 11, it took about 60 minutes to define this geometry model by using Designbase. The mesh consists of 14,250 tetrahedral elements and 27,458 nodes. Nodes and elements are generated in about 13 minutes and in about 2 minutes, respectively. This is

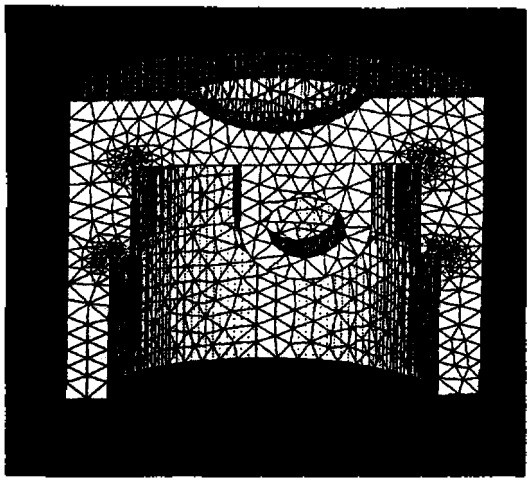


Fig. 11. Mesh for a half of piston head (No. of nodes = 27,458, No. of elements = 14,250).

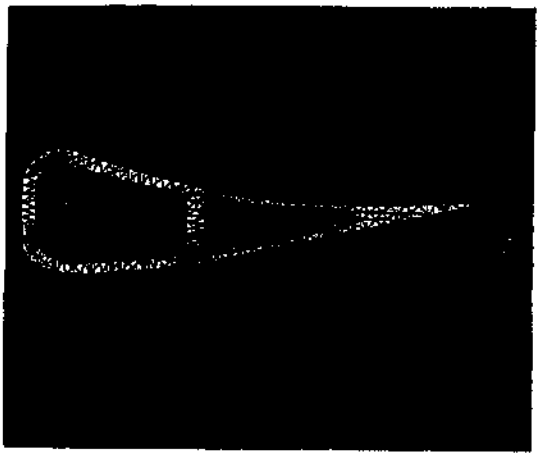


Fig. 12. Mesh for a part of turbine blade (No. of nodes = 16,252, No. of elements = 9,059).

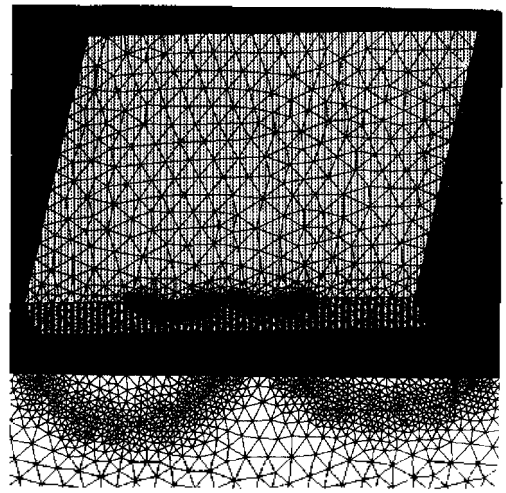


Fig. 13. Mesh of a plate with two dissimilar surface cracks (No. of nodes = 16,252, N0. of elements = 9,059).

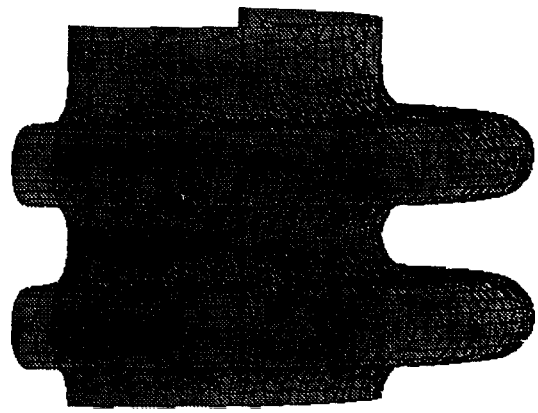


Fig. 14. Mesh for a symmetric 2 convolution of bellows (No. of nodes = 3,588, No. of elements = 3,870).

measured on a popular EWS, SUN SparcStation 10(1CPU, 50 MHz). To complete this mesh, the following two node patterns are utilized; (a) the base node pattern in which nodes are generated with uniform spacing over a whole analysis domain, (b) a special node pattern for stress concentration of four corners. In case of shell geometry as shown in Fig. 14, nodes and elements were generated in about 15 minutes and in about 5 minutes, respectively.

In general, mesh generation time increases in accordance with the increasing number of nodes or the number of total DOFs. Fig. 15 shows a relationship between processing time and number of nodes for a half of piston head. It is observed that the node generation speed hear can be carried out in the averaged run time proportional to the number of nodes. It is pointed out that the node generation algorithm is very important for a very large scale complex problem. To reduce a processing time, we are adopt the following improvements:

(a) Employment of the bucket method for node generation.
(b) Employment of the Delaunay method for element generation.

(c) Implementation of the algorithm on popular EWS with the C and C++ language under the Unix environment.

Owing to (a) and (b), mesh generation can be performed in computation time of $O(n)$ (n : the number of nodes). By such improvement, the generation of three-dimensional meshes of ten thousands DOFs can be performed within half an hour in popular EWS.

5. Conclusions

An automatic mesh generation system for three-dimensional solid and shell structures consisting of free-form surfaces has been presented. Here several local node patterns are selected and are automatically superposed based on the fuzzy knowledge processing technique. In addition, several computational geometry techniques were successfully applied to node and element generation, whose processing speed is proportional to the total number of nodes. The developed system was utilized to generate meshes of three-dimensional complex geometries. The key features of the present algorithm are an easy control of complex three-dimensional node density distribution with a fewer input data by means of the fuzzy knowledge processing technique, and fast node and element generation owing to some computational geometry techniques. The effectiveness of the present system is demonstrated through several mesh generations for three-dimensional complex geometry.

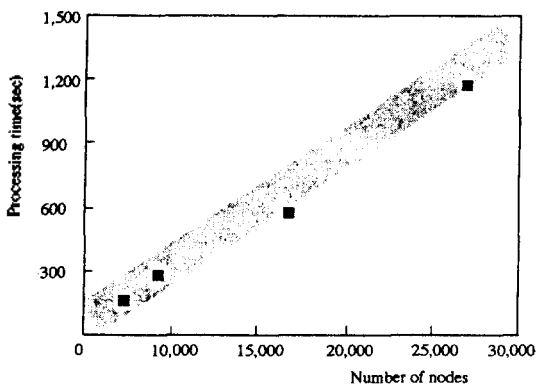


Fig. 15. Processing time vs. total number of nodes.

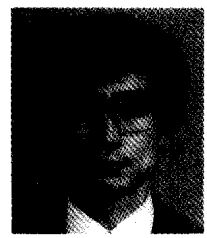
References

1. Lee, D.T. and Schachter, B.J., "Two Algorithms for Constructing a Delaunay Triangulation", *International Journal of Computer and Information Sciences*, Vol. 9, No. 3, pp.219-242, 1980.
2. Shephard, M.S., Yerry, M.A. and Baehmann, L., "Automatic Mesh Generation Allowing for Efficient a Priori and a Posteriori Mesh Refinement", *Computer Methods in Applied Mechanics and Engineering*, Vol. 55, pp.161-180, 1986.
3. Watson, D.F., "Computing The N-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes", *The Computer Journal*, Vol. 24. No. 2, pp. 167-172, 1981.
4. Schroeder, W.J. and Shephard, M.S., "Combined Octree/Delaunay Method for Fully Automatic 3-D Mesh Generation", *International Journal for Numerical Methods in Engineering*, Vol. 29, pp.37-55, 1990.
5. Pourazady, M. and Radhakrishnan, M., "Optimization of a Triangular Mesh", *Computers and Structures*, Vol. 40, No. 3, pp.795-804, 1991.
6. Bowyer, A., "Computing Dirichlet Tessellations", *The Computer Journal*, Vol. 24, No. 2, pp.162-172, 1981.
7. Buratynski, E.K., "Fully Automatic Three-Dimensional Mesh Generator for Complex Geometries", *International Journal for Numerical Methods in Engineering*, Vol. 30, pp.931-952, 1990.
8. Shephard, M.S. and Georges, M.K., "Automatic Three-Dimensional Mesh Generation by the Finite Octree Technique" *International Journal for Numerical Methods in Engineering*, Vol. 32, pp.709-749, 1991.
9. Al-Nasra, M. and Nguyen, D.T., "An Algorithm for Domain Decomposition in Finite Element Analysis", *Computers and Structures*, Vol. 39, No. 3/4, pp.277-289, 1981.
10. Sezer, L. and Zeid, I., "Automatic Quadrilateral/Triangular Free-Form Mesh Generation For Planar Regions", *International Journal for Numerical Methods in Engineering*, Vol. 32, pp.1441-1483, 1991.
11. Green, P.J. and Sibson, R., "Computing Dirichlet Tessellations in The Planes", *The Computer Journal*, Vol. 21, No. 2, pp.168-173, 1977.
12. Taniguchi, T. and Ohta, C., "Delaunay-Based Grid Generation for 3D Body with Complex Boundary

Geometry", Grid Generation Conference, 1991.
13. Sloan, S.W., "Fast Algorithm for Constructing Delaunay Triangulation in the Plane", *Advanced Engineering Software*, Vol. 9, No. 1, pp.34-55, 1987.
14. Zadeh, L.A. "Outline of a New Approach to the Analysis of Complex Systems and Decision Process", *IEEE Transactions on System, Man and Cybernetics*, SMC-3, pp.28-44, 1983.
15. Chiyokura, H., *Solid Modeling with Designbase: Theory and Implementation*, Addition-Wesley, 1988.
16. Asano, T., "Practical Use of Bucketing Techniques in Computational Geometry", *Computational Geometry*, pp.153-195, North-Holland, 1985.
17. Yagawa, G., et al., "Automatic Two-and Three-Dimensional Mesh Generation Based on Fuzzy Knowledge Processing", *Computational Mechanics*, Vol. 9, pp.333-346, 1992.
18. Zadeh, L.A., "Fuzzy algorithms", *Information and Control*, Vol. 12, pp.94-102, 1968.
19. Cavendish, J.C., "Automatic Triangulation of Arbitrary Planar Domains for the Finite Element Method", *International Journal of Numerical Method in Engineering*, Vol. 8, pp.679-696, 1974.

이 준 성
1986년 성균관대학교 기계공학과 학사
1988년 동대학원 기계공학과 석사
1988년-1991년 육군사관학교 기계공학과 전임
1991년~1992년 KIST 기전연구부 연구원
1995년 동경대학교 대학원 공학계 연구과 공학박사
1996년~현재 경기대학교 기계공학과 전임
관심분야 : CAE, 구조물의 자동설계, 마이크로머신의 설계

Yagawa Genki
1970년 동경대학 대학원 공학계 연구과 공학박사. 동대학 강사, 조교수를 역임
1986년 미국 조지아 공대 객원 교수
1984년 현재 동경대학 공학부 교수
일본시뮬레이션학회 부회장,
일본계산공학회 부회장
일본기계학회 이사, 일본응용수리학회 이사
관심분야 : 계산역학, Super parallel computing, 원자로의 안정성



박 면 웅
1977년 서울대학교 기계공학과 학사
1979년 KAIST 생산공학과 석사
1987년 영국 UMIST 공대 기계공학과 공학박사
1988년~1990년 UMIST 연구원
1990년~1993년 KIST 기전연구부 선임연구원
1994년~현재 KIST 기전연구부 책임연구원
관심분야 : 공정설계, 절삭가공, 공작기계설계, Geometric modeling