Annual Report of
ADVENTURE Project
ADV-99-1 (1999)

AUTOMATIC MESH GENERATION OF HEXAHEDRAL ELEMENTS USING INTELLIGENT LOCAL APPROACH

Yoshitaka WADA[*1], Shinobu YOSHIMURA[*2], and Genki YAGAWA[*1]

*1 School of Engineering, University of Tokyo
E-mail: wada@garlic.q.t.u-tokyo.ac.jp, web page: http://garlic.q.t.u-tokyo.ac.jp/

*2 Graduate School of Frontier Sciences, University of Tokyo

Key words: Intelligent Local Approach, Hexahedral Elements, Mesh Generation, Fuzzy Knowledge Processing.

1 INTRODUCTION

Unstructured mesh generation for the finite element method (FEM) and the finite difference method (FDM) is in general very labor-intensive and time-consuming. This issue has been recently becoming more critical because the model to be analyzed tends to become very complex and the complexity of mesh also increases. Numerous research activities have been devoted into the development of automatic mesh generation techniques. The Delaunay method, which automatically generates triangular elements for 2-D case or tetrahedral elements for 3-D case for a given set of nodes, has been well studied and utilized [1-4]. The Quadtree/Octree methods have been also popularly utilized to generate triangular / tetrahedral meshes, respectively [5,6]. The present authors have developed an automatic mesh generation system for triangular or quadrilateral elements for 2-D plane and 3-D shell, and tetrahedral elements for 3-D solid using the fuzzy knowledge processing and some computational geometry techniques such as the Bucketing method and the Delaunay method [7-9]. This FuzzyMesh method has been implemented in CAE systems for car body modeling in several Japanese automobile companies [10-12], and in one commercial pre/post processor [13,14].

Thus, we can say that automatic mesh generation techniques for triangular elements and tetrahedral elements have already been established. On the other hand, the automatic mesh generation of hexahedral elements for 3-D solid is still an open problem, though this element is strongly demanded for some kinds of analyses.

There are two main reasons for such strong demand on hexahedral elements. Firstly, compared with tetrahedral elements, hexahedral elements are more suitable to strongly nonlinear problems such as deformation of tires and metal forming. Secondly hexahedral elements with large aspect ratio are good at modeling thinner 3-D solid and boundary layers in fluid dynamics. Several research activities have been devoted into the development of automatic generation of hexahedral mesh, including the Plastering method [15], the Whisker Weaving method [16] and HEXAR [17]. However, these methods and systems are still under development. They are not always stable, and cannot control element size and aspect ratio as human does manually or as human desires.

In the previous paper [18], the present authors proposed a new automatic mesh generation method named Intelligent Local Approach (ILA), which can control size and aspect ratio of quadrilateral mesh. The ILA is a kind of heuristic method that is derived from the analysis of human expert's approach. This was applied to the generation of several quadrilateral meshes. In the present study, we extend the ILA into an automatic generation of hexahedral mesh.

The structure of this paper is organized as follows. The next section describes a fundamental principle of ILA. The third section describes the algorithms of ILA for hexahedral mesh generation, followed by some additional treatments in the fourth section. Finally fundamental performances of the ILA are demonstrated through several examples of hexahedral meshes.

2 FUNDAMENTAL PRINCIPLE

It is well known that automatic generation of hexahedral mesh is a very complicated and difficult task. Nevertheless, human experts on mesh generation can generate hexahedral meshes for an arbitrarily shaped domain using their superior capability for image recognition and qualitative judgement, if the number of elements to be generated is small. The present authors have invented a new mesh generation method for quadrilateral and hexahedral meshes, basically (a) by analyzing such mesh generation processes by human experts, (b) by abstracting elemental processes hidden, and (c) by systematizing them into a complete procedure using latest computer science technology. The elemental processes abstracted are summarized as follows:

(1) Imagine qualitatively a whole distribution of element size and that of aspect ratio over a whole geometry model.

(2) Generate elements one by one, starting from an outer boundary or inner boundary of the geometry model.

(3) Collect geometrical information such as angle, length and area of generated elements from a local region where one to few elements are to be generated.

(4) Imagine "virtual elements" to be generated, and evaluate their goodness. Then determine actual node location and element, based on the fuzzy judgement for goodness of " virtual elements".

(5) Regenerate old elements if they are not compatible to a newly generated element.

The mesh generation process based on the hypothesis above is named "Intelligent Local Approach (ILA)", and an actual system is developed. ILA's actual implementation to quadrilateral mesh generation and some examples can be found in the previous paper [18].

3 ALGORITHMS

Figure 1 shows the general flow of ILA implemented based on the hypothesis explained in the previous section. In the figure, the superscript denotes the corresponding hypothesis in section 2, while the subscript corresponds to the subsection in which each algorithm is described.

3.1 Front Boundary

In 2-D plane case, an initial front boundary is defined as a set of segments, which represent a geometry model. In 3-D solid case, the initial front boundary is defined as a set of quadrilateral surface patches, each of which consists of four edges and four nodes, as shown in Figure 2 (a). The outer boundary of the model is named an outer (front) boundary, while the inner boundary is named an inner (front) boundary as shown in Figure 2 (b). Starting from one surface patch of the initial front boundary, one to few elements are generated sequentially. The front boundary is then updated by erasing a few surface patches covered by newly generated elements.

3.2 Collection of Local Geometrical Information

After determining one surface patch from which mesh generation starts, geometrical information is collected from a local region near the surface patch. In hexahedral mesh generation, the geometrical information includes sizes of surfaces and angles of two adjacent surfaces as shown in Figure 3.

In the current version of system, such geometrical information is collected along the front boundary as illustrated in Figure 4. The total number of collected sizes of surfaces is 37, while the total number of collected angles of any two adjacent surfaces is 48. As discussed later in section 5, these numbers are not definite values. If we increase these numbers, we can perform more flexible mesh generation. But we should remember that increasing these numbers also increases processing time.

3.3 Categories of Angle Division

Considering the local geometrical information collected, we categorize a geometrical situation. Table 1 summarizes five kinds of categories of angle division, C1, C2, C3, C4 and C12. In C1, the angle of two adjacent surfaces should not be divided. In C2, C3 and C4, the angle should be divided into two, three or four angles, and then one, two or three surfaces are newly generated, respectively. The angle of C12 is dealt as that of either C1 or C2, depending on its circumstance. C1 has the first priority among the five categories. After multiple angles are collected from a local region and classified into the above division categories, an element is generated first in the C1 category angle.

3.4 Patterns of Element Generation

Based on the categories of angle division given in the previous subsection, we define four kinds of 2-D patterns of element generation as shown in Figure 5. "2-D pattern 0" shown in Figure 5 (a) is the pattern that a new element is automatically determined without generating any nodes or edges. In "2-D pattern 1", an edge connecting nodes n1 and n2 are generated in order to generate an element as shown in Figure 5 (b). In "2-D pattern 2", one node, two edges and one element are generated as shown in Figure 5 (c). In "2-D pattern 3" shown in Figure 5 (d), the angle of the two adjacent edges is divided into two, three or four angles, and then one, two or three edges are newly generated, respectively. Priority of the four 2-D patterns is given in Table 2.

In 3-D case, we define 3-D patterns of element generation through the analogy with the 2-D patterns. Nine independent patterns, which are derived considering full combinations of surfaces and edges surrounding a relevant surface, are given in Figure 6. In "3-D Pattern 0" is basically the same as "2-D Pattern 0". In this pattern, a new element is automatically determined without generating any nodes or edges. In "3-D Pattern 1", an element can be generated by newly generating two surfaces, but without generated edges and nodes. In "3-D Pattern 2" and "3-D Pattern 2a", an element can be generated by only generating one surface. In "3-D Pattern 3" and "3-D Pattern 4", an element can be also generated by only generating some surfaces. In "3-D Pattern 5", "3-D Pattern 6" and "3-D Pattern 7", an element is generated by generating both new nodes and edges. We define priority of the 3-D patterns. "3-D Pattern 0" is the easiest pattern to create an element. Excluding this, highest priority is given to "3-D Pattern 1".

These 3-D patterns are not all of possible 3-D patterns. Considering existence of edges, there are 41 more patterns in addition to the previous 9 patterns. In order to reduce the complexity of these patterns a concept of semi-surface is defined. A semi-surface consists of three edges as shown in Figure 7 (a). In a semi-surface, a surface can be easily created by generating only one edge without any new nodes. Figure 7 (b) shows a case of "3-D Pattern 3" with one vertical edge. There are two semi-surfaces. In Figure 7 (b) two dashed lines are additional edges, and this pattern can be converted to "3-D Patterns 2". All of 3-D patterns with some edges can be converted to nine basic 3-D patterns.

3.5 Determination of Node Location Using Fuzzy Knowledge Processing

After determining the pattern of node and element generation as described in subsections 3.3 and 3.4, the location of node is determined so as that the shapes of newly generated elements are appropriate. For the judgement, we employ the following fuzzy knowledge processing [19, 20].

At first, we explain in detail the method of 2-D case, which is a basis for the method of 3-D case. Then the method of 3-D case is explained.

(1) Geometrical Parameters for Goodness of Element Shape

As shown in Figure 8, we employ the following six kinds of parameters for the judgement of goodness of element shape, i.e. (a) ratios of opposite edges, S_1 / S_3 and S_2 / S_4, (b) ratio of diagonal lines, l_1 / l_2, (c) square root of area, $\sqrt{A}$,(d) ratios of opposite angles, θ_1 / θ_3 and θ_2 / θ_4, (e) average of ratios of opposite edges, $(S_1 / S_3 + S_2 / S_4) / 2.0$, (f) ratio of summations of adjacent edges, $(S_1+S_2) / (S_3+S_4)$. These geometrical parameters become 1.0 for a rectangular element. l_0 means length of standard element evaluated from the field of element size specified by a user, which will be explained subsequently.

(2) Information for Control of Mesh Subdivision

The control of both element size and aspect ratio is a key issue for any automatic mesh generation techniques. In the ILA, a user can specify the following two kinds of "fields": (a) the field of element size and (b) that of aspect ratio. A field of priority of element creation is also specified within the ILA. The three fields are stored over a uniform Cartesian grid, i.e. a background cell, enveloping a geometry model. The values at any location are linearly interpolated from the grid point values.

(2a) Field of Element Size

Element size is a scalar value. The reference point for this information is defined as a central point of edge, a gravity center of surface, or that of volume. Even a complicated distribution over a whole geometry model can be easily specified using the fuzzy knowledge processing as described in the FuzzyMesh method [7-9].

(2b) Field of Aspect Ratio

Aspect ratio is also scalar. Priority is given to this information near the boundary. On the other hand, this information is often neglected in the region far from the boundary.

(2c) Field of Priority of Element Creation

In the ILA, elements are generated sequentially. The field of priority of element creation directs the sequence of element generation. One possible method is that elements are generated along boundaries, from the outer to the inner of the geometry model. Another method is that elements are generated in the one direction as if water was fulfilled in vase. In the latter method, the field of priority of element creation plays the same role as "Gravitational potential field" in mechanics. According to the latter method, unmeshed area always contains a part of the outer boundary of the geometry model. Therefore, geometrical constraint is still weak to some extent, even at the last moment of the mesh generation.

(3) Fuzzy Knowledge Processing

Node location is precisely determined, considering the geometrical parameters defined in subsection 3.5(1) and the field information defined in subsection 3.5(2). Taking the case of category C2 as a typical example, the present fuzzy knowledge processing is explained in the following.

(3a) 2-D Case

Figure 9 shows a typical membership function employed in this study. The horizontal axis denotes any of geometrical parameters, while the vertical axis does the degree of membership, i.e. appropriateness.

At the first step, a local x-y region where a new element is generated is mapped onto a $\xi-\eta$ grid space as shown in Figure 10 (a). Here thick solid lines both in the *x-y* space and in the $\xi-\eta$ grid space denote the segment with which a new element is to be generated. At the second step, we presume a virtual node at each grid point, and evaluate goodness of shape of each virtual element using its geometrical parameters. For each parameter, a degree of membership is calculated grid point by grid point. Figure 10 (b) illustrates the distribution of degree of membership μ_a for "ratio of angle" parameter of category C2 plotted in the $\xi-\eta$ grid space. Figure 10 (c) illustrates the similar distribution for "ratio of opposite edges" parameter μ_l. If we consider only the two geometrical parameters, we superpose the μ_a and μ_l distributions by the product operation of the fuzzy set theory [5,6], and get the ($\mu_a \wedge \mu_l$) distribution as shown in Figure 10 (d). In really, we generate more distributions for the six kinds of geometrical parameters of one virtual element, and superpose them. If multiple virtual elements are generated at once, the multiple superposed distributions for these elements are further superposed into the one distribution. Node location to be generated is finally determined as the grid point where the degree of membership of the last superposed distribution takes the maximum as illustrated in Figure 10 (d). In the ILA, the process of the determination of node location takes most computation time. However, this process can be extremely speeded up by employing parallel processing [21,22], since this can be perfectly parallelized in grid point wise.

(3b) 3-D Case

The method for 2-D case can be straightforwardly extended to 3-D case. However, such a straightforward method requires too many grid points of evaluation, i.e. the order of n^3, where n is the division number in one dimension, and is not efficient. Therefore we adopt the following alternate method, which is explained taking "3-D Pattern 5" as an example.

Firstly a plane consisting of two edges (S_1-S_2) are mapped onto 2-D plane as illustrated in Figure 11 (a). Secondly one node is generated in the plane using the same method explained in subsection (3a). Thirdly this node is mapped back to the 3-D space. The same process is repeated to both the plane consisting of (S_3-S_4) and that of (S_5-S_6). Thus we can obtain three locations of nodes, the mean position of which is finally adopted as a new node to be generated. Then one element is generated in the 3-D space. In this method, the number of grid points of evaluation is the order of $3 \times n^2$. Thus, this method is more efficient than the straightforward 3-D method.

4 SOME SPECIAL TREATMENTS

As explained in the previous sections, the basic process of ILA is as follows.

(a) Collect geometrical information from a local region.
(b) Categorize angles and determines patterns of mesh generation.
This is a kind of process to locally determine connectivity.
(c) Precisely determine node location using a fuzzy knowledge processing.

The current version of system stops in the middle of mesh generation. There are two main reasons.

(1) Misjudgment in recognition of patterns
(2) Impossible situation

The first issue is not so easy issue, but not impossible. Basically, we can resolve this issue, by strengthening the capability of recognition of geometrical situation. In other words, we collect geometrical information from a wider area than that mentioned in subsection 3.2.

The second issue is more serious. Figure 12 shows such an example. Here, the initial front boundary consists of all good quadrilateral elements. However, when mesh is being generated, the final element in the top of pyramid becomes inconsist. We cannot predict such a situation at the starting point. There is some idea to resolve it. In the current version of ILA system, we do not change the initial set of surface patches obtained from a geometry model, and utilize their nodes, edges and surfaces as parts of final mesh. It is also possible to modify some of these surface patches, i.e. increase or decrease of nodes, edges and surfaces on the geometry model surface if they lead to inconsistency of mesh. This function will be implemented in the next development.

5 DIFFERENCES BETWEEN ILA AND OTHER MESH GENERATION METHODS

Let us now compare the ILA with some mesh generation methods. First of all, features of the advancing front (AF) method can be summarized as follows.

(1) Elements are generated along front boundaries. It is necessary to inspect intersection of front boundaries.

(2) In case of generation of a triangle element, a new node is generated in consideration of element size, and an existing segment and the node are connected to each other. Finally one triangle is generated.

In the case of height of triangle element of l, the node is generated on the perpendicular line from the center of the selected segment, and one new triangle element is generated using them.

(3) Rules for mesh generation and exception in case of triangular mesh generation are much less than in case of quadrilateral mesh generation.

Secondly, three features of paving method are shown here.

(1) Elements are generated along each row. Row represents a direction of projection to generate elements. There is no consideration of geometrical restriction.

(2) Quadrilateral element generation using projection toward the row influences mesh generation at the next row so that elements may become bigger and bigger or smaller and smaller.

(3) Quadrilateral element generation using projection toward the row certainly influences mesh generation at the next row so that smoothing process is needed after mesh generation at each row.

Finally, the ILA has the following three main features.

(1) Elements are generated even where geometrical restriction is hard. The ILA does not need a concept of row, which is essential in the paving method.

(2) Local connectivity of elements is decided referring a finite number of patterns, which are called 2-D pattern or 3-D pattern.

Element generation patterns depend on local categories of division. A new node is generated at a proper position after evaluation of element size and shapes on the grid.

(3) Decision of node location using the fuzzy knowledge processing keeps good size and shape of element which are defined by a numerical field for element size. Therefore the ILA does not need a smoothing process, which is also essential in the paving method

Furthermore the ILA is abstracted in order to handle 2-D quadrilateral and 3-D hexahedral elements in the same way.

(1) Division of angles is abstracted by categorizing an angle between two adjacent segments or surfaces. We can convert the degree of an angle to a number of divisions using the division categories.

(2) Element generation patterns which are called 2-D pattern or 3-D pattern are utilized. An element generation pattern is a particular set of division categories.

(3) Firstly the element generation pattern is decided by local situation of a front boundary. Secondly, some new nodes are located using the fuzzy knowledge processing. These processes are almost the same way in 2-D and 3D mesh generation.

Moreover we compare the ILA with other mesh generation methods as shown in Table 3. Meaning of each item in the table is as follows.

・Advancing Front (AF) Method

This item shows whether mesh generation methods are similar to the advancing front method. HEXAR is not like the advancing front method.

・Element by Element Generation / Mapping Generation

This item shows whether elements are generated one by one or almost at once.

・Determination of Node Location

This item shows how location of nodes is determined.

・Capability of Controlling Element Shapes

This item shows how well the methods control shapes of elements. In the ILA aspect ratio can be controlled using the fuzzy knowledge processing in consideration of several parameters for element shapes.

・Order of Element Generation

This shows how elements are generated in respect of the order of element generation. In the ILA a location with the strongest geometrical restriction is found, and elements are generated at the location. In the whisker weaving method there are no rules to decide a position starting element generation. In HEXAR uniform and regular elements are generated at once. In the plastering method elements are generated along the row.

・Consistency of Algorithm in Dimensions

In the ILA four geometrical primitives are defined. The first primitive is a node which represents a position in a space. The second primitive is a segment which is constructed with the two nodes. Third primitive is a quadrilateral surface which is constructed with the four segments. Finally the fourth primitive is a hexahedral cubic which is constructed with the six surfaces. These combinations of the primitives do not directly depend on dimensions. Only a node depends on a dimension. We can treat 2-D and 3-D data structure and node generation method in the same way since other primitives are independent of dimensions. The whisker weaving method realizes also generalized mesh generation which is independent of dimensions because of STC representation.

6 EXAMPLES

Mesh generations of all examples given here are performed on a PC with one Pentium II 333MHz, 384 MBytes memory and 8 GBytes disk. Operating system is FreeBSD 3.2 and Compiler is egcs-1.1.1 release. According to some preliminary studies, 200 Mbytes memory is found to be necessary as ILA system's working area. All figures of examples show meshes, numbers of nodes and elements and times of mesh generation.

Figure 13 (a) shows a mesh for a cylinder. Here element size is specified as 1.5 at the upper part, while as 1.0 at the lower part. Aspect ratio is specified as a constant value of 1.0 over the whole domain. Figures 13 (b) shows a mesh for a bearing model. Figure 13 (c) shows that for a simplified model of carbon block of HTTR (High Temperature Engineering Test Reactor). In the two models, both element size and aspect ratio are specified as 1.0 over the whole domain. Figure 13 (d) shows a mesh of a gear. Figure 13 (e) shows a course mesh of air portion in a subway station. By subdividing each element of the coarse mesh into 8 smaller elements, we obtain a finer mesh as shown in Figure 13 (f).

In the current version of system, the capability of controlling aspect ratio is not completely implemented yet, though this capability was verified for mesh generation of quadrilateral elements as in Ref. [18]. This is another issue to be implemented in the next version of the system.

7 CONCLUSIONS

We presented a new mesh generation method named Intelligent Local Approach for hexahedral elements. The ILA consists of the following fundamental features:

(1) Sequential mesh generation.
(2) Collection of geometrical information from a local region.
(3) Explicit specification of user's demand on element size and aspect ratio as fields information.
(4) Fuzzy knowledge processing for multiple criteria on goodness of hexahedral elements.

The developed system was successfully applied to some mesh generation problems. We will develop the new system which has full capabilities to control element size and aspect ratio. Further research is needed to generate a mesh with more complex geometry.

REFERENCES

[1] Sibson, R. (1978), Locally Equiangular Triangulations, The Computer Journal, Vol.21, pp.243-245.
[2] Bowyer, A. (1981), Computing Dirichlet Tessellations, The Computer Journal, Vol.24, pp.162-166.
[3] Watson, D. F. (1981), Computing the n-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes, The Computer Journal, Vol.24, pp.167-172.
[4] Johnson, A. A. and Tezduyar, T. E. (1994), Mesh Update Strategies in Parallel Finite Element Computations of Flow Problems with Moving Boundaries and Interfaces, Computer Methods in Applied Mechanics and Engineering, Vol. 119, pp.73-94.
[5] Shephard, M. S. and Georges, M. K., Automatic Three-dimensional Mesh Generation by the Finite Element Octree Technique, International Journal for Numerical Methods in Engineering, 32 (1991) 709-749.
[6] Shephard, M. S., Update To : Approaches to the Automatic Generation and Control of Finite Element Meshes, Transactions of ASME, Journal of Applied Mechanics, 49 (1996) S5-S14.
[7] Yagawa, G. , Yoshimura, S. , Nakao, K. and Soneda, N. (1992), Automatic Two- and Three-Dimensional Mesh Generation Based on Fuzzy Knowledge Processing, Computational Mechanics, Vol.9, pp.333-346.
[8] Yagawa, G., Yoshimura, S. and Nakao, K. (1995), Automatic Mesh Generation of Complex Geometries Based on Fuzzy Knowledge Processing and Computational Geometry, Integrated Computer-Aided Engineering, Vol.2, pp.265-280.
[9] Yagawa, G., Kawai, H., Yoshimura, S. and Yoshioka, A. (1996), Mesh-invisible Finite Element Analysis System in a Virtual Reality Environment, Computer Modeling and Simulation in Engineering, Vol.2, pp.289-314.
[10] Yagawa, G., Yoshimura, S., Nakao, K. and Ohji, M. (1992), Development of an Automatic Mesh Generation System for Shell Structures Based on Fuzzy Knowledge Processing, JSAE Review, Vol.13, pp.60-64.
[11] Kuboshima, T., Ishizuka, S., Suzuki, S. and Nakamura, H. (1996), Embedded Modeling Routine in IN-HOUSE Design CAD System, Proceedings of Spring Conference of Japan Society of Mechanical Engineers (JSME), No.96-13, pp.253-254 (in Japanese).

[12] Miyoshi, K. (1997), A Shell Mesh Generation Technique for the Automotive Assembly, Proceedings of the Conference on Computational Engineering and Science, Vol.2, pp.589-592.

[13] KSWAD^{TM} Ver.6 User's Manual, KUBOTA Corporation, (1995) (in Japanese).

[14] Nagafuku, R. (1997), New Pre- and Postprocessor, KSWAD Ver.6.0, Journal of Japan Society of Computational Engineering and Science, Vol.2, pp.47-50 (in Japanese).

[15] Blacker, T. D. and Meyers, R. J. (1993), Seams and Wedges in Plastering : A 3-D Hexahedral Mesh Generation Algorithm, Engineering with Computers, Vol.9, pp.83-93.

[16] Tautges, T. J., Blacker, T. and Mitchell, S. A. (1996), The Whisker Weaving Algorithm : A Connectivity-based Method for Constructing All-hexahedral Finite Element Meshes, International Journal for Numerical Methods in Engineering, Vol.39, pp.3327-3349.

[17] Kimura, N., Mochizuki, Y. and Mikami, K. (1996), The HEXAR System as a Software Tol Designed for Automatic, All Hexahedral, Boundary-Fitted Unstructured Grid Generation, Proceedings of the Conference on Computational Engineering and Science, Vol.1, pp.683-686 (in Japanese).

[18] Yoshimura, S., Wada, Y. and Yagawa, G., (1999), Automatic Mesh Generation of Quadrilateral Elements Using Intelligent Local Approach, International Journal for Computer Methods in Applied Mechanics and Engineering, Vol.179, pp.125-138.

[19] Zadeh, L. A. (1968), Fuzzy Algorithms, Information and Control, Vol.12, pp.94-102.

[20] Zadeh, L. A. (1973), Outline of a New Approach to the Analysis of Complex Systems and Decision Process, IEEE Transactions on Systems, Man and Cybernetics, Vol.SMC-3, pp.28-44.

[21] Yagawa, G., Yoshioka, A., Yoshimura, S. and Soneda, N. (1993), A Parallel Finite Element Method with Supercomputer Network, Computers and Structures, Vol.47, pp.407-418.

[22] Yagawa, G. and Shioya, R. (1993), Parallel Finite Elements on a Massively Parallel Computer with Domain Decomposition, Computing Systems in Engineering, Vol.4, pp.495-503.

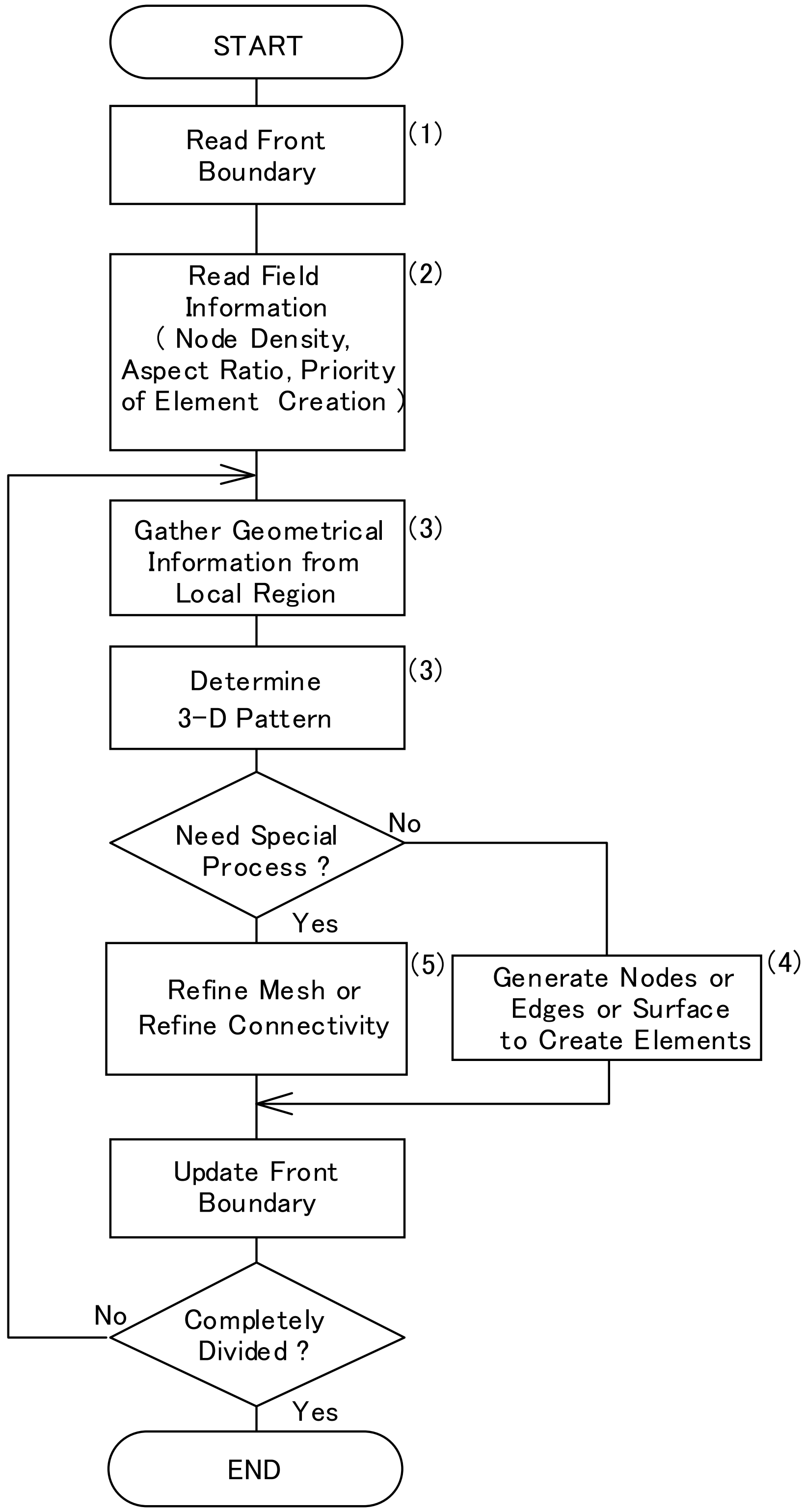
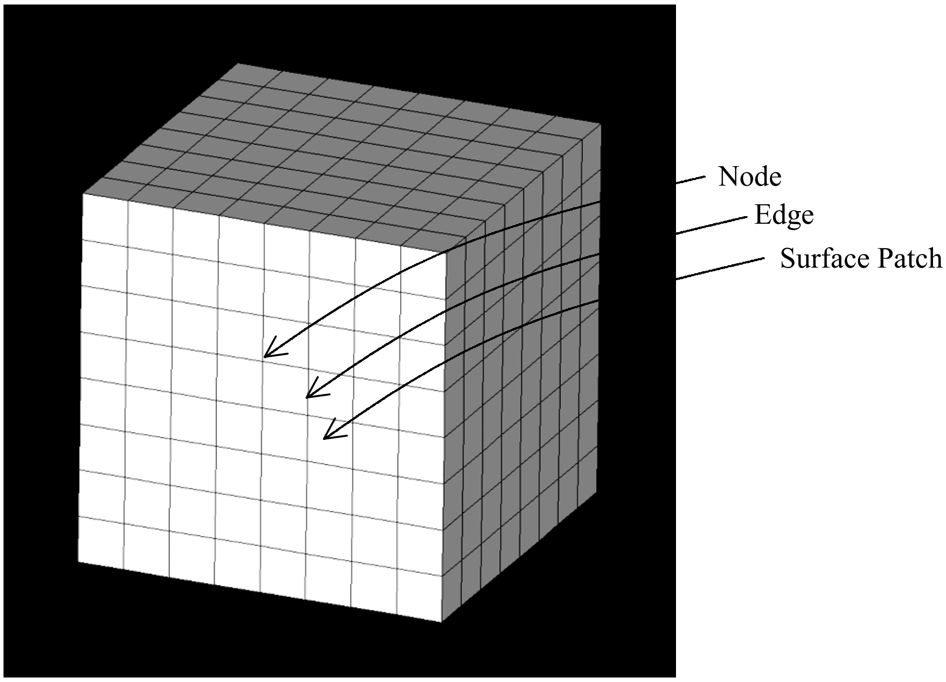
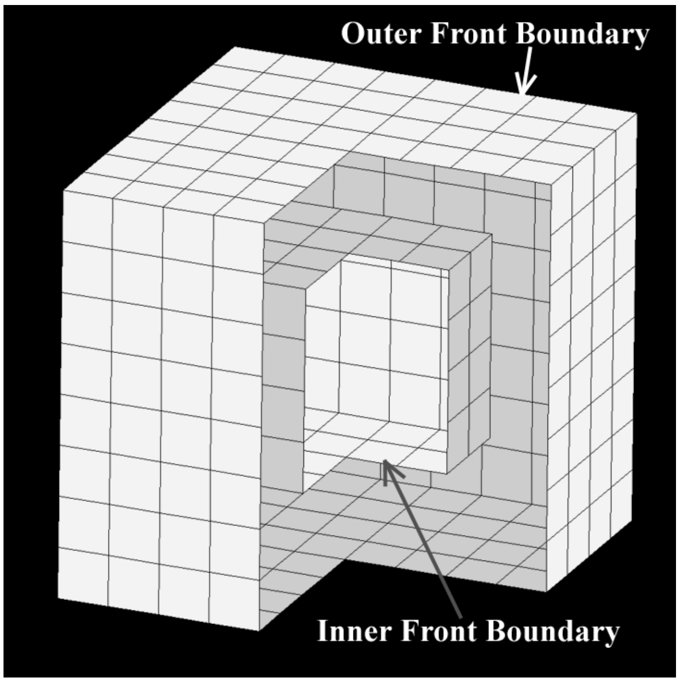


Figure 1. Flowchart of ILA.

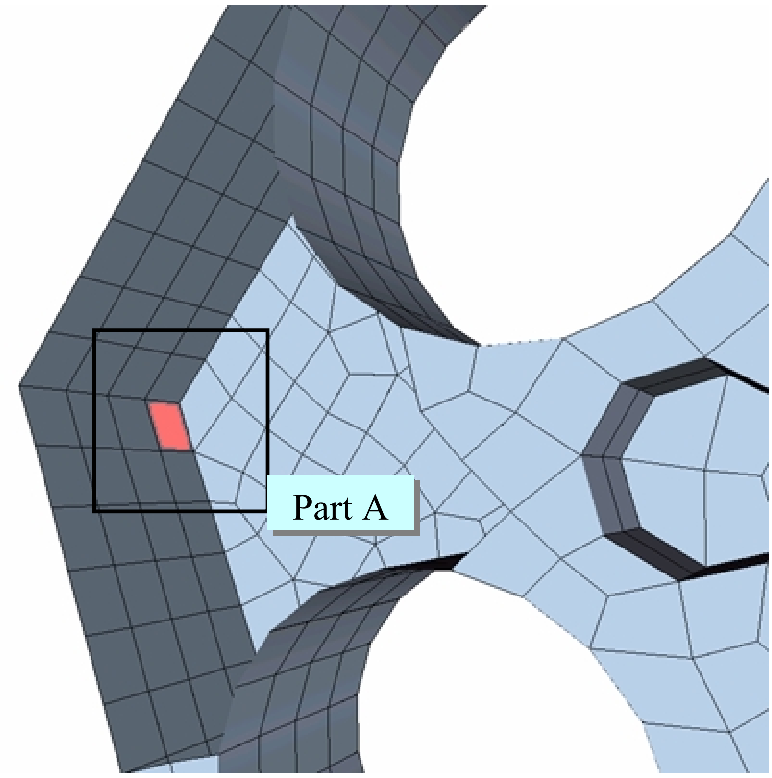


(a) Initial Front Boundary

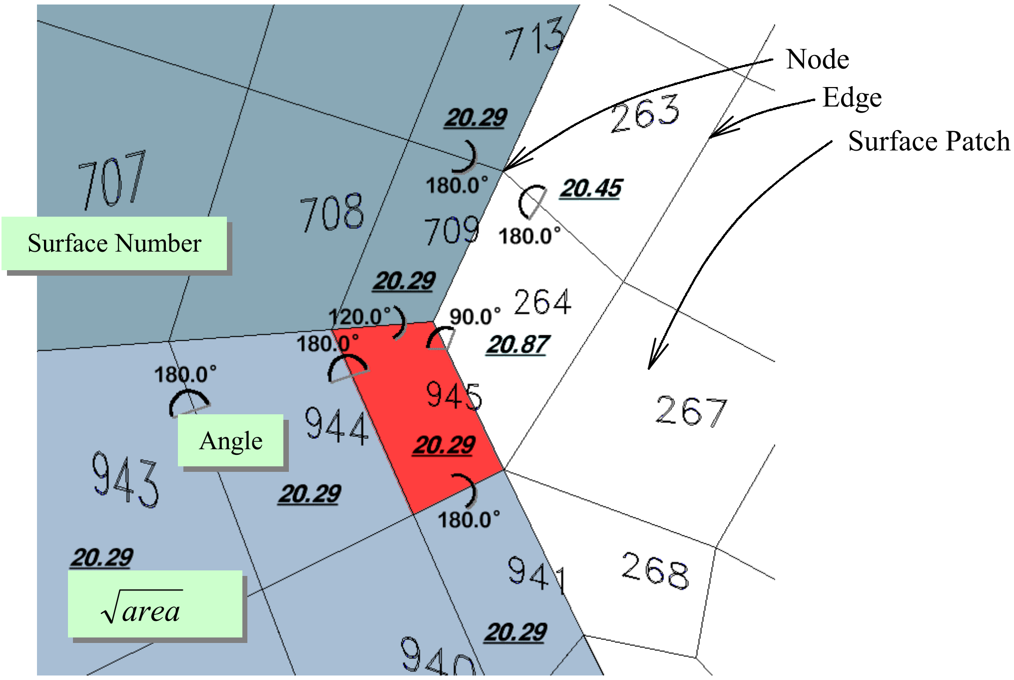


(b) Inner and Outer Front Boundaries

Figure 2. Front Boundary



(a) An Actual Front Boundary



(b) Geometrical Information in Part A

Figure 3. Example of 3-D Geometrical Information along Front Boundary

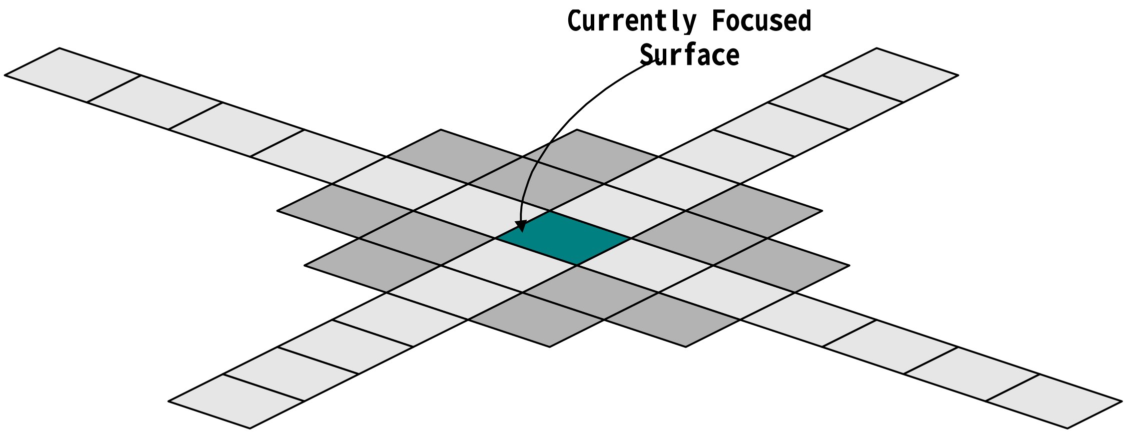
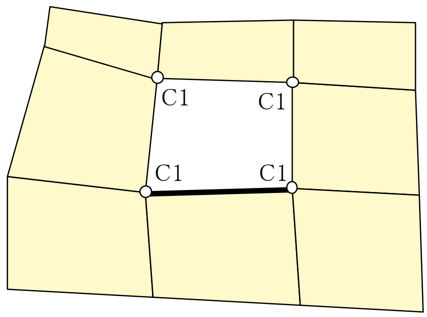
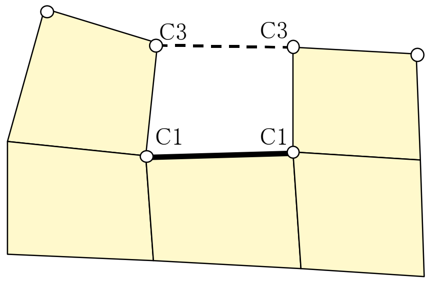


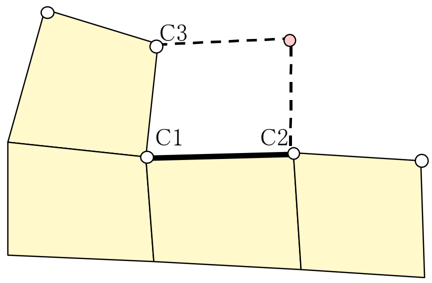
Figure ４． Area of Search for 3-D Geometrical Information along Front Boundary



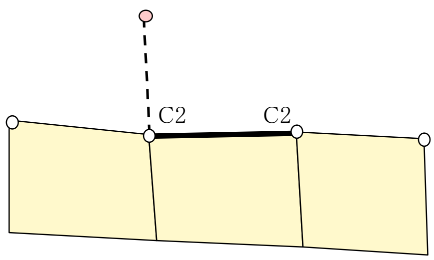
(a) 2-D Pattern 0



(b) 2-D Pattern 1



(c) 2-D Pattern 2



(d) 2-D Pattern 3

Figure 5. 2-D Patterns for Creation of Connectivity

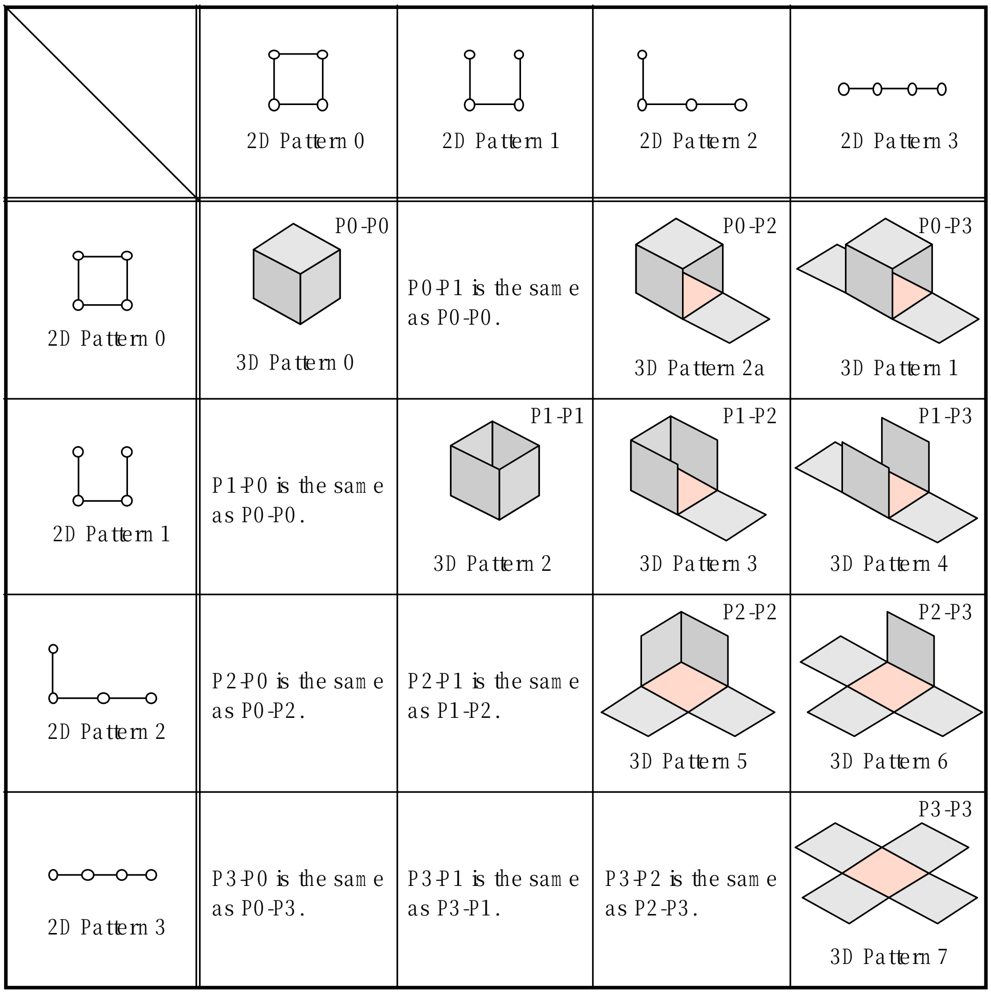
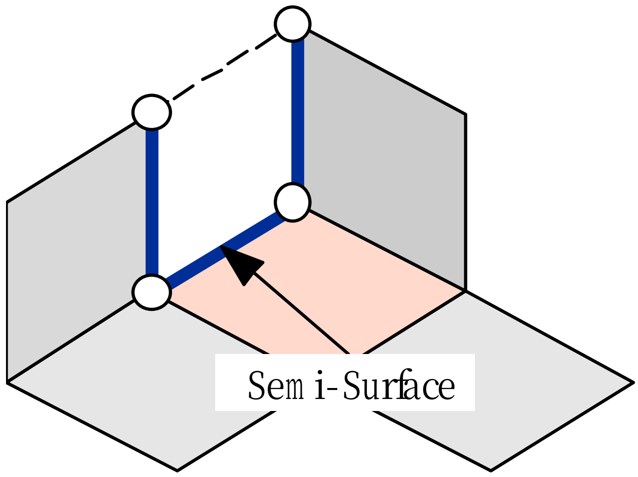
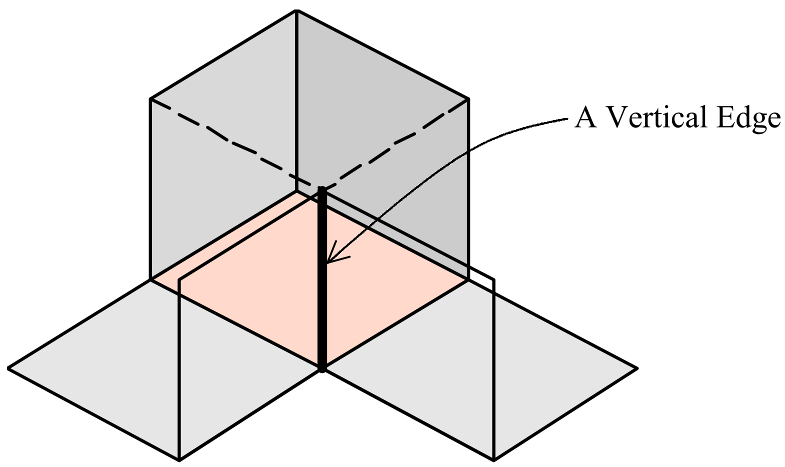


Figure 6. 3-D Patterns for Creation of Connectivity.



(a) Semi-surface



(b) Pattern with a Vertical Edge

Figure 7. Reducing a complexity of 3-D patterns with edges.

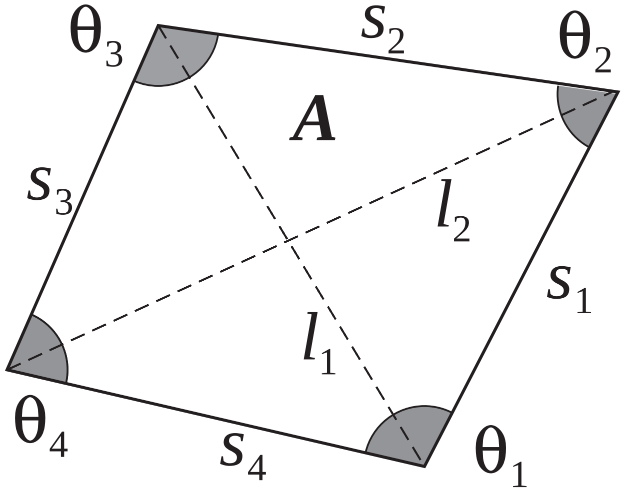


Figure 8. Geometrical Parameters of Quadrilateral Elements.

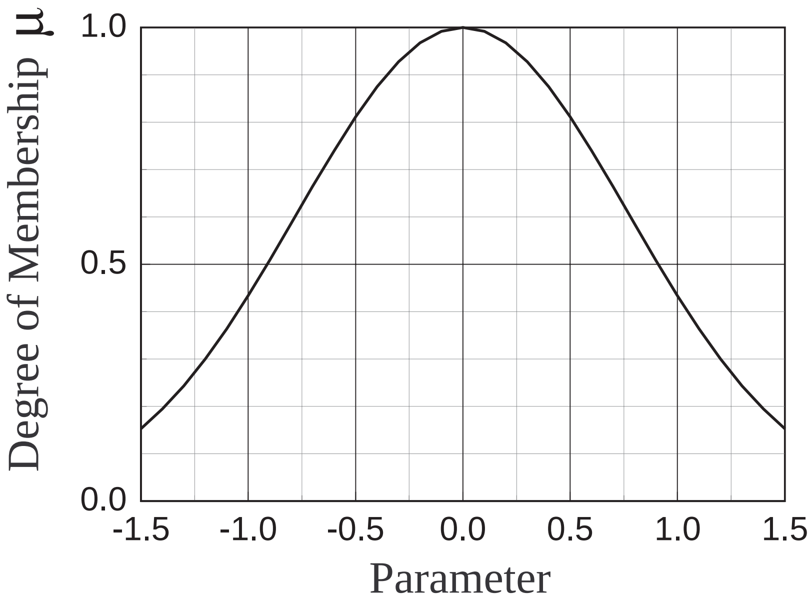
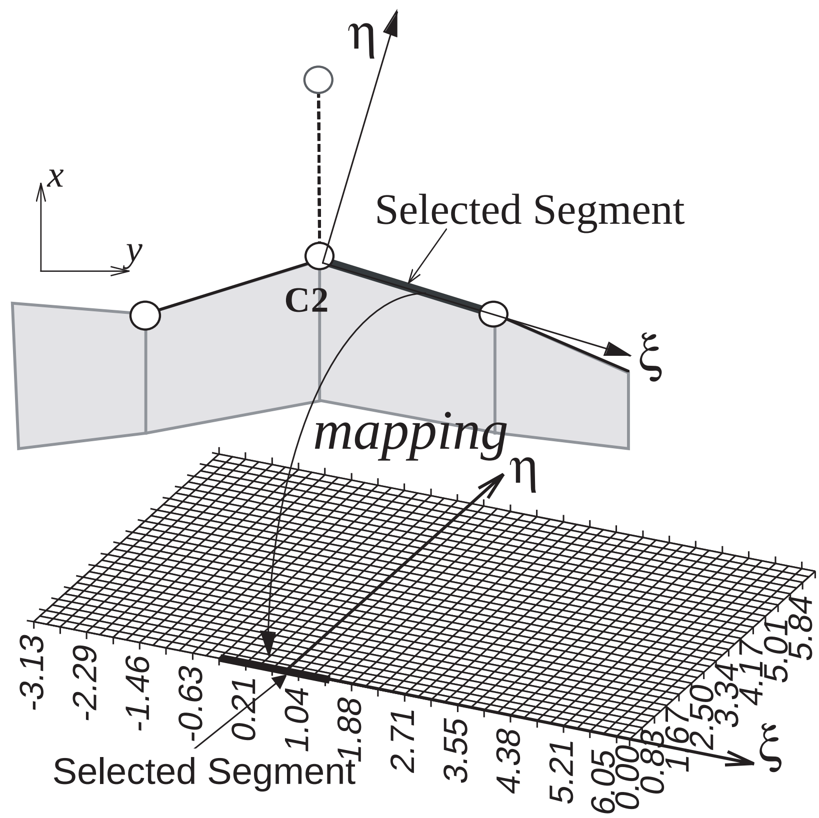
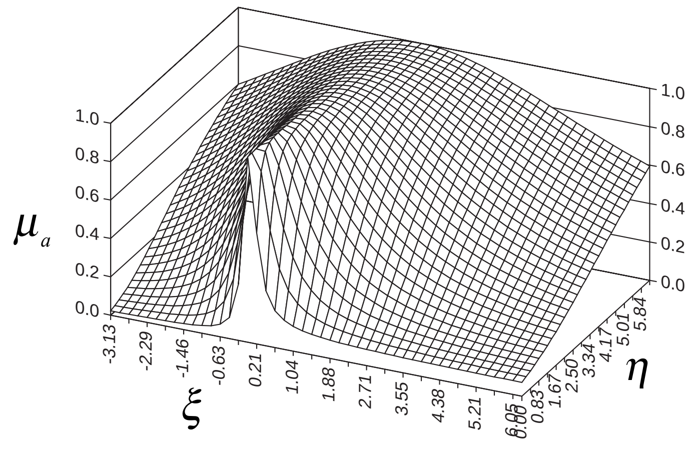


Figure 9. Typical Membership Function.

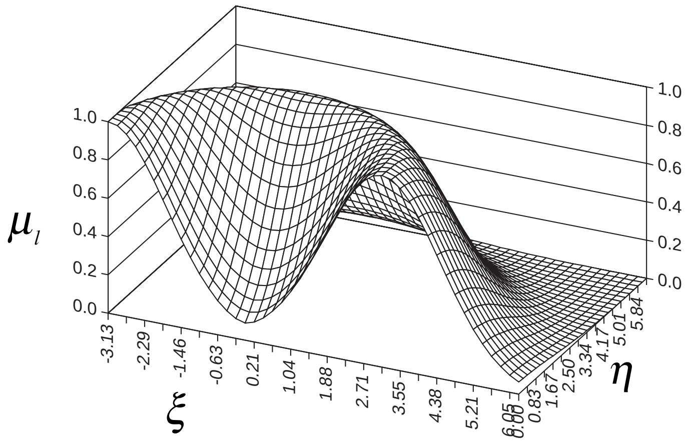


(a) Mapping from Local x-y Space to ξ-η Space

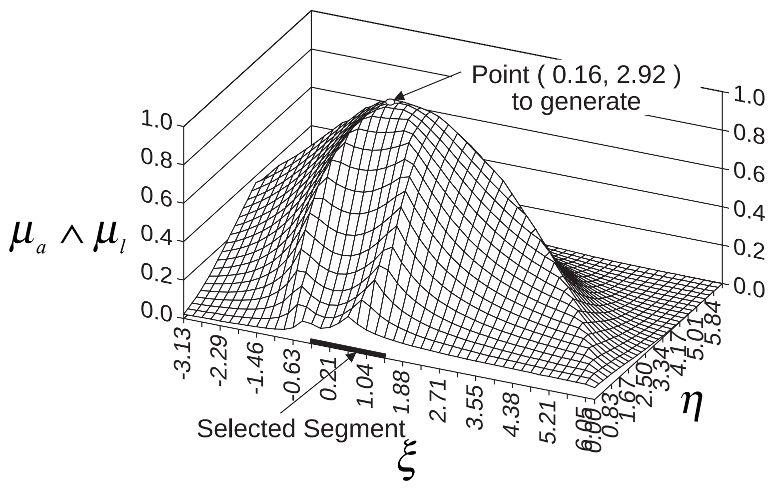
Figure 10. Node Generation Based on Fuzzy Knowledge Processing in 2-D



(b) μ_a Distribution over the Grid

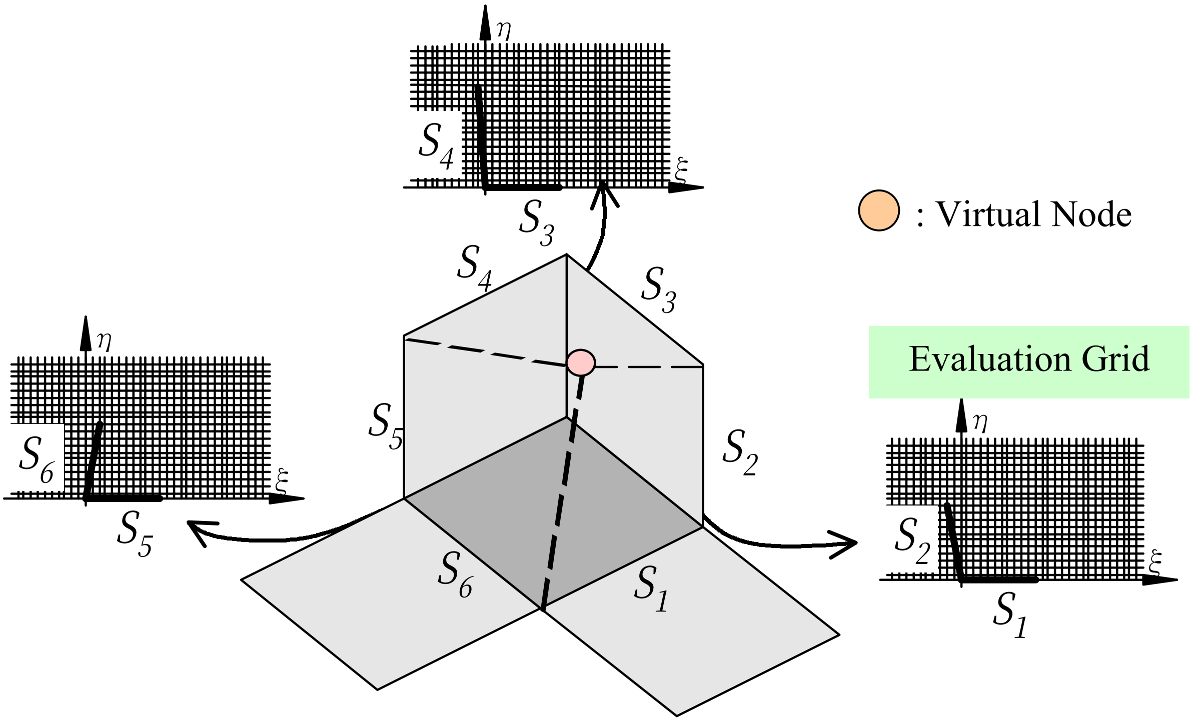


(c) μ_l Distribution over the Grid

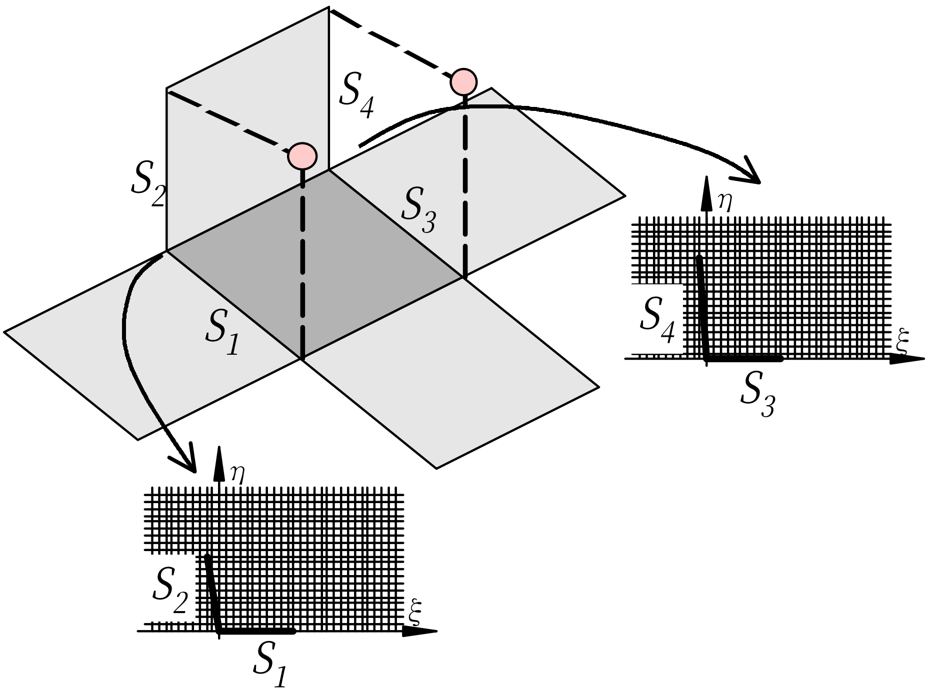


(d) $\mu_a \wedge \mu_l$ Distribution Derived from Fuzzy Product Operation

Figure 10. Node Generation Based on Fuzzy Knowledge Processing in 2-D



(a) Case of 3-D Pattern 5



(b) Case of 3-D Pattern 6

Figure 11. Node Generation in 3-D

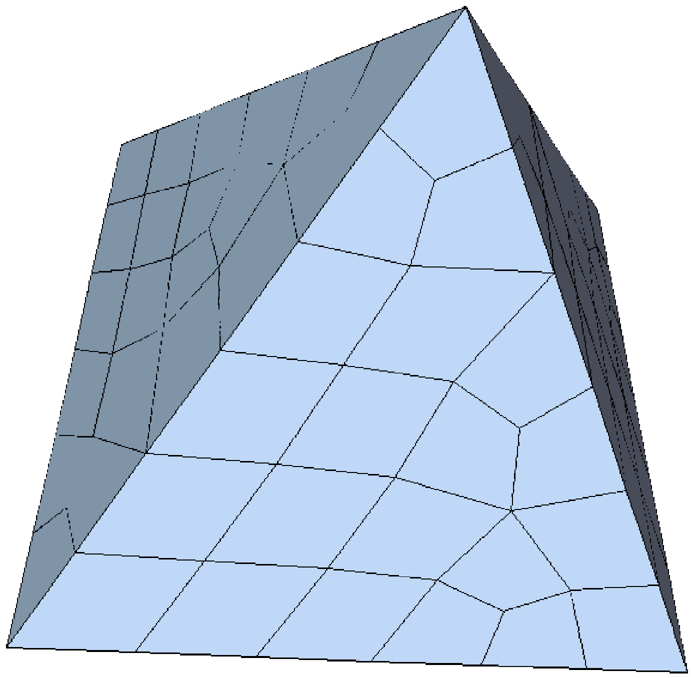
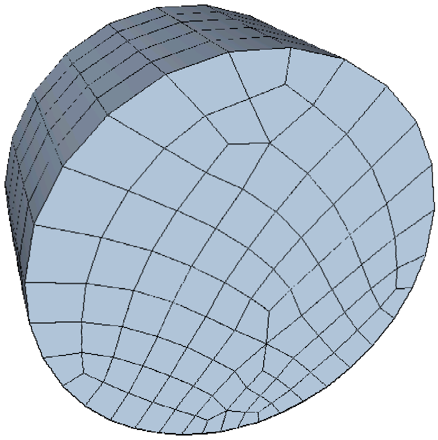
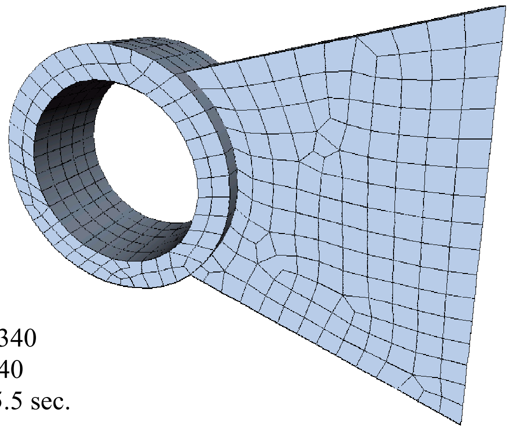


Figure 12. Impossible case to create elements



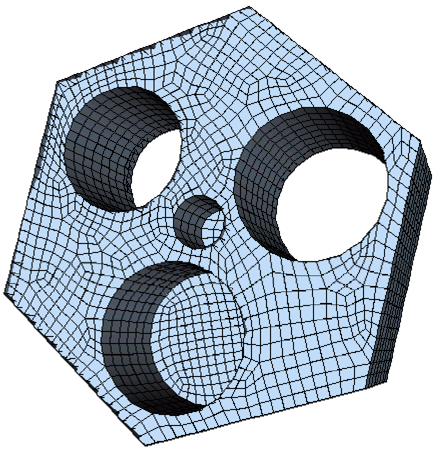
Number of Nodes: 660
Number of Elements: 465
Generation Time: 44.2 sec.

(a) Cylindrical Shape



Number of Nodes: 1340
Number of Elements: 840
Generation Time: 75.5 sec.

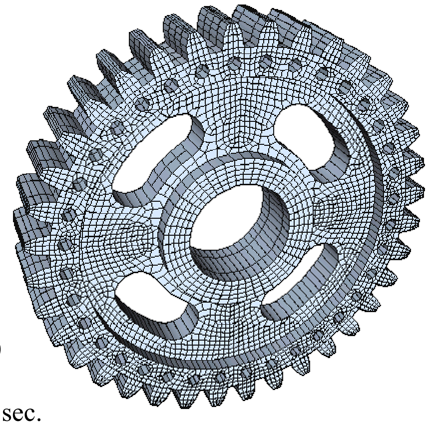
(b) Bearing



Number of Nodes: 8333
Number of Elements: 6360
Generation Time: 64.2 sec.

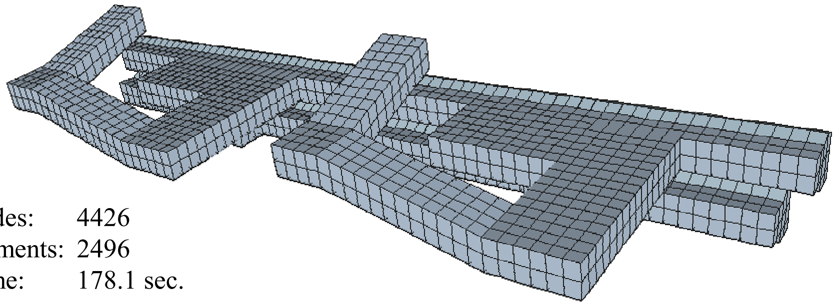
(c) HTTR Carbon Block

Figure 13. Examples of Hexahedral Mesh



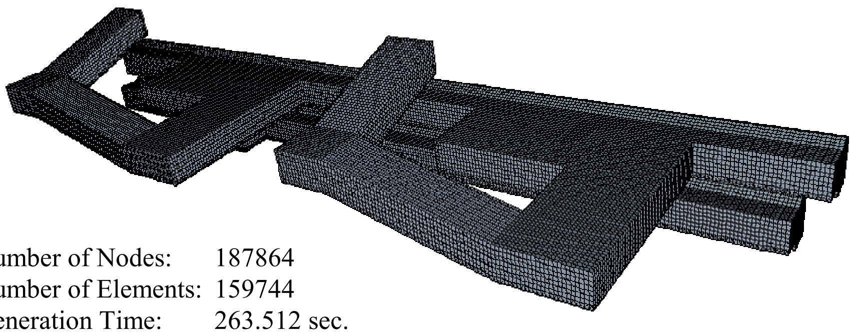
Number of Nodes: 14940
Number of Elements: 9664
Generation Time: 108.3 sec.

(d) Gear



Number of Nodes: 4426
Number of Elements: 2496
Generation Time: 178.1 sec.

(e) Station of Subway (Coarse Mesh)



Number of Nodes: 187864
Number of Elements: 159744
Generation Time: 263.512 sec.

(f) Station of Subway (Fine Mesh)

Figure 13. Examples of Hexahedral Mesh

Table 1. Definition of Division Category

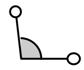
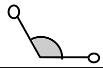
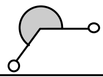
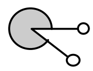
Division Category	Number of Divisions	An Angle of Two Edges (degree)
C1	1	45-105 
C12	1 or 2	105-155 
C2	2	155-200 
C3	3	200-300 
C4	4	300-360 

Table 2. Priority of 2-D Patterns

Creation Pattern	Creation Priority	
2D Pattern 0	1	High
2D Pattern 1	2	
2D Pattern 2	3	
2D Pattern 3	4	Low

Table 3. Comparison with Mesh Generation Algorithms

Method	Advancing Front (AF) Method	Element by Element Generation / Mapping Generation	Determination of Node Location	Capability of Controlling Element Shpes	Order of Element Generation	Consistency of Algorithm in Dimension
ILA	A kind of AF	Element by Element	Grid Based Evaluation in Consideration of Several Parameters	Size / Aspect Ratio	Geometrical Restriction	Consistent Algorithm using Geometrical Primitives
Wisker Weaving	A kind of AF	Element by Element	Undefined	Size	No Rules	Consistent Algorithm using STC
HEXAR	Other	Element by Element	Unknown (Unpublished)	Impossible (Fine elements are generated)	Generation of Whole Elements at once	Unknown (Unpublished)
Plastering	A kind of AF	Element by Element	Projection	Size	Element Generation along Each row	Inconsistent Algorithm (Projection is different depending on 2D or 3D)