Transactions of the 15th International Conference on
Structural Mechanics in Reactor Technology (SMiRT-15)
Seoul, Korea, August 15-20, 1999

Parallel Automatic Mesh Generation of Nuclear Structures with Ten-million Nodes

Shinobu Yoshimura[1], Hideki Nitta[2], Genki Yagawa[1] and Hiroshi Akiba[2]

1) University of Tokyo, Japan
2) Allied Engineering Corporation, Japan

ABSTRACT

This paper describes a parallel automatic mesh generation of ten-million nodes using the fuzzy knowledge processing and computational geometry. The mesh generation process consists of three sub-processes : (a) definition of an analysis domain, i.e., a geometry model, (b) generation of nodes, and (c) generation of elements. One of commercial solid modelers, i.e. MicroCADAM is employed for the sub-process (a). In the sub-process (b), several local distributions of node density are first chosen from the node density distribution database constructed in advance, and their distributions are superposed on one another over the analysis domain by using the fuzzy knowledge processing. Then, nodes are generated by the bucketing method whose processing speed is almost proportional to the total number of nodes. In the sub-process (c), tetrahedral solid elements are generated using the Delaunay triangulation. To solve problems on computation speed and memory requirement for large scale problems with ten-million nodes, a domain decomposition type parallel algorithm is effectively employed in both node and element generation. Fundamental performances of the present method are demonstrated through the mesh generation of graphite core support structure of High Temperature Engineering Test Reactor (HTTR).

INTRODUCTION

Various general-purpose computational mechanics systems have been developed in the last three decades to quantitatively evaluate natural phenomena such as deformation of solids, heat transfer, fluid flow and electro-magnetics. Nowadays such systems are regarded as infrastructural tools for the present industrialized society. The existing systems, however, cannot be used with massively parallel computers with the order of 10^2-10^4 processing elements, which are expected to dominate the high-performance computing market in the 21st century, as they were developed for single-processor computers, which took leadership an age ago. Neither the current systems can be used in heterogeneous computing environments. Owing to the fact, they can deal with only medium scale problems with sub-million degrees of freedom at most.

Under the sponsorship of Research for the Future Program of Japan Society for the Promotion of Science, our research group is developing an advanced general-purpose computational mechanics system which can analyze a model of arbitrary shape with 10-100

millions degrees of freedom within one hour to one day using the world's fastest computer with 30-100 TFLOPS in 2002[1].

The present paper describes a part of the preprocessor of the present computational mechanics system, i.e. a parallel automatic mesh generation of ten-million nodes using the fuzzy knowledge processing and computational geometry. The remaining sections summarize fundamental algorithms with an example of mesh for nuclear structures.

FUNDAMENTAL PRINCIPLE

Flow

The present mesh generation method is of node generation type. Its flow is summarized as follows[2-4]:

(a) Define a whole analysis domain using one of commercial geometric modelers, i.e. MicroCADAM[5].
(b) Generate nodes inside the analysis domain, on its boundary surfaces, edges and vertices. In this process, a node density distribution over the domain is controlled based on the fuzzy knowledge processing[6,7], and nodes are generated using the bucketing method[8].
(c) Generate tetrahedral elements using the Delaunay triangulation technique[9-11].
(d) Remedy distortion of elements and mismatch elements using the Laplacian smoothing technique and others.

Definition of Geometry Model

In the current version of system, one of commercial geometric modelers, MicroCADAM is utilized to define geometries of analysis domains. The constructed geometry data including free-form surfaces are automatically converted into a set of triangular patch data, which will be utilized in the subsequent node and element generation processes, instead of the original geometry model.

Control of Node Density Distribution by Fuzzy Knowledge Processing

The mesh generation system contains various types of local node density distributions as illustrated in Figure 1. A user selects some of them, and designates their locations to be placed. Based on the fuzzy knowledge processing as illustrated in Figure 2, the selected local distributions are then automatically superposed to one another over the whole analysis domain, and a global three-dimensional distribution of node density is obtained. These

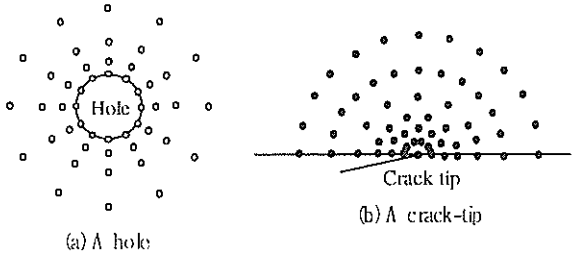


Figure 1 Illustrative examples of local node density distributions

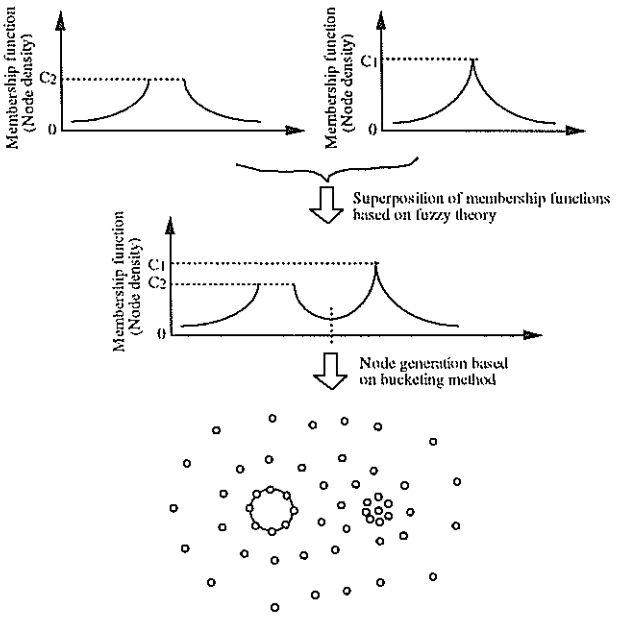


Figure 2 Superposition of local node distributions based on fuzzy knowledge processing

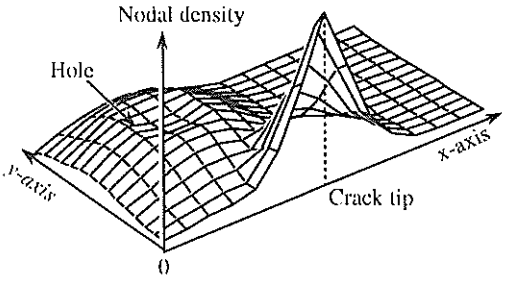
node density distributions are stored as grid point values over a regularly spaced orthogonal grid covering the analysis domain. Any node density value within each unit grid space is linearly interpolated from the eight adjacent grid points when requested. It should be noted here that the global node density distribution can also be constructed, referring to posteriori error estimation if necessary[4].

Node Generation Based on Bucketing Method

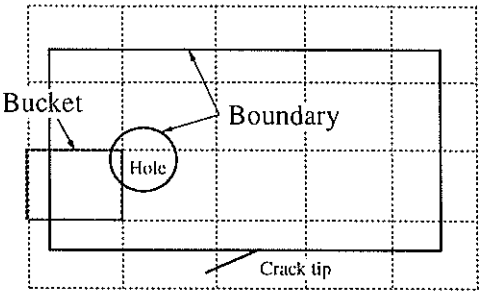
Figure 3 illustrates the flow of the node generation method, taking a two-dimensional mesh generation as an example without any loss of generality. Nodes are generated so as to meet the global distribution of node density defined over the whole analysis domain using the bucketing method as follows.

Let us assume that the global distribution of node density over a whole analysis domain is already given as shown in Figure 3(a). At first, a rectangule enveloping the analysis domain is defined as shown in Figure 3(b). Next, the super-rectangule is divided into a number of small sub-rectangules, each of which is named "Bucket". In the three-dimensional solid case, a box is utilized to envelop an analysis domain. In the present system, the three-dimensional uniform orthogonal grid space which is first prepared for storing node density distributions is further divided into non-uniform smaller grids based on the Octree method[12], considering the global distribution of node density. Each grid space is named "Bucket". Nodes are generated bucket by bucket. Taking one bucket, let us explain the node generation procedure.

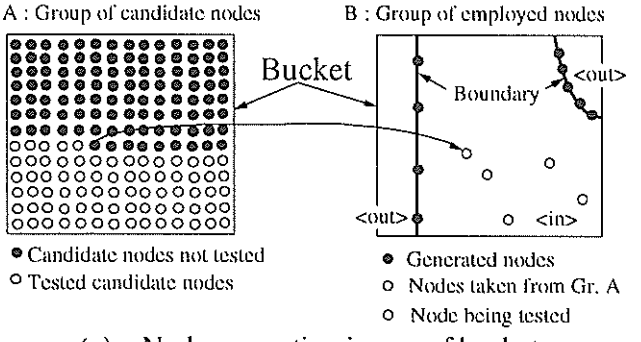
At first, a number of candidate nodes with uniform spacing are prepared as shown in the *lhs* of Figure 3(c). The distance of two neighboring candidate nodes is set to be smaller than the minimum distance of nodes to be generated in the relevant bucket. Here the former distance is chosen about one fifth of the latter one through trials and errors. Next, candidate nodes are pick up one by one, starting from the left-bottom corner of the bucket,



(a) Example of global node density distribution



(b) Example of bucket decomposition



(c) Node generation in one of buckets

Figure 3 Node generation based on bucketing method

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 specific point, which is evaluated from the global distribution of node density, to some extent.

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 also noted 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 reduced significantly, and then a node generation speed is remained to be proportional to n. As for three-dimensional solid geometries, nodes are generated in the

following order ; vertices, edges, surfaces and domain. To shorten computation time for the IN/OUT check, a set of triangular puch data is utilized, instead of the original geometry model consisting of free-form surfaces.

Element Generation

The Delaunay triangulation method is utilized to generate tetrahedral elements from numerous nodes produced within a geometry in the previous process. If the Delaunay triangulation method is utilized to generate elements in a geometry with concave shape, mismatch elements occurs. The mismatch elements can be removed by performing the "IN/OUT check" for gravity center points of the elements. On the other hand, when some point exists closer to the domain boundary compared with the distance between two neighboring points lying on the boundary, mismatch elements across the domain boundary tend to occur. To avoid these kinds of mismatch elements, nodal densities on the domain boundary should be controlled to be slightly higher than those near the boundary. To completely diminish such mismatch elements, some additional mesh remedy techniques are required after the element generation over the whole domain.

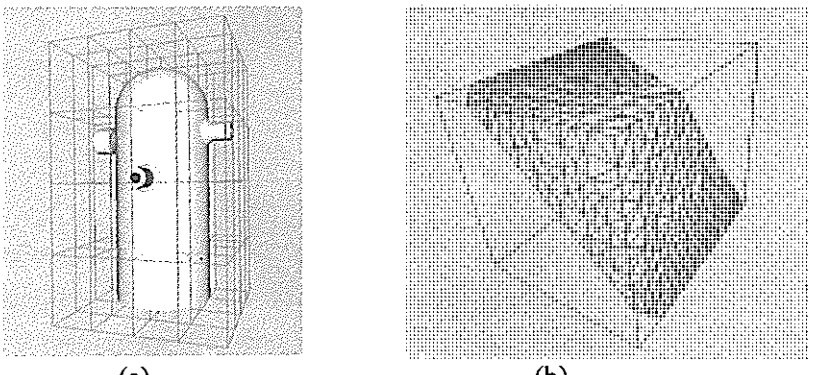
Smoothing Operation

The above algorithm of element generation works well in most cases. However, element shapes obtained are sometimes distorted near domain boundary. The smoothing method called "Laplacian operation" is here applied to modify such distorted elements. 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.

PARALLEL ALGORITHMS AND ENVIRONMENT

To efficiently generate a large scale mesh of ten-million nodes, parallel processing is indispensable in shortening computation time as well as in storing large scale data in core memories. Among all the sub-processes for the present mesh generation, we parallelize the node generation process using the bucketing method and the element generation process using the Delaunay triangulation.

Each unit grid space in the original uniform and orthogonal grid as shown in Figure 4(a) is adopted as a parallel task domain. At first, nodes are generated at all the grid points, edges



(a) (b)

Figure 4 Brick subdomain for parallel processing

and faces of the original grid space. Nodes are then generated in each uniform grid space grid-space by grid-space in parallel. Tetrahedral elements are then generated using the Delaunay trianguation grid-space by grid-space as shown in Figure 4(b). Those parallel tasks for node and element generation are dynamically distributed over a number of processing elements using a server-client model.

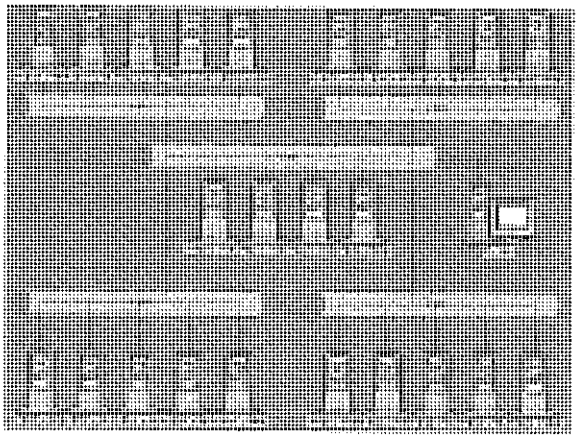


Figure 5 PC cluster

In the present system, all interactive operations such as geometry modeling and selection of local node distributions are executed in PC/Windows environment, while heavy computation tasks such as the bucketing and the Delaunay methods are executed in a parallel environment. In pre-processing, interactive operation is indispensable even if dealing with large scale models of ten-million nodes problems. We have employed PC cluster consisiting of 25 sets of DEC Alpha 533MHz (Unix/Linux, 2MB cache, 1GMB memory and 2GB disk) as shown in Figure 5. C++ and MPI library are employed to build the system. Thus the presnet system is designed so as to operate in various kinds of heterogeneous computer environments.

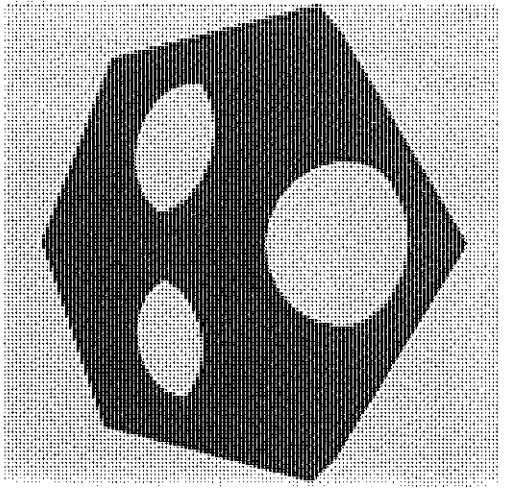


Figure 6 Ten million DOFs tetrahedral mesh of HTTR

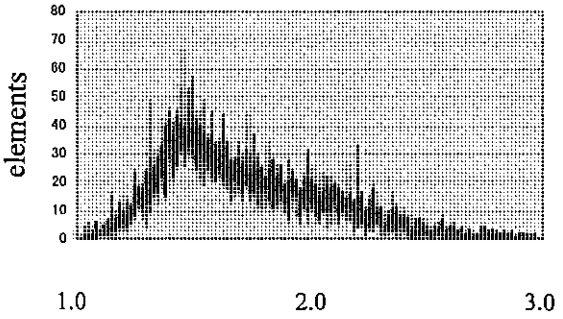


Figure 7 Histogram of aspect ratio

RESULTS

Figure 6 shows one of meshes generated by the present system, i.e. a ten-millon DOFs mesh of simplified model of graphite core support structure of High Temperature Engineering Test Reactor (HTTR). It takes about two hours using one PE, while taking six minutes using 25 PEs. Figure 6 shows a histogram of aspect ratio of tetrahedral elements included in one parallel domain. The peak occurs at the aspect ratio of 1.4, while the worst ratio is 4.3. This distribution is almost the same as that of the result obtained with the sequential version. Thus the present parallel algorithm does not affect much mesh quality.

CONCLUSIONS

This paper proposed a parallel automatic mesh generation system for large scale meshes with ten-million nodes. Here several local node distributions are chosen from the database and are automatically superposed based on the fuzzy knowledge processing technique. Several computational geometry techniques are successfully applied to node and element generation. Domain decomposition type parallel algorithm is implemented to shorten computation speed and to relax memory requirement. The system is constructed in PC cluster using C++ and MPI library. 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 parallel computational geometry techniques.

REFERENCES

1. http://adventure.q.t.u-tokyo.ac.jp/
2. Yagawa, G., Yoshimura, S., Soneda, N. and Nakao, K., "Automatic two- and three-dimensional mesh generation based on fuzzy knowledge processing", *Computational Mechanics*, Vol.9, 1992, pp.333-346.
3. Yagawa, G., Yoshimura, S. and Nakao, N., "Automatic mesh generation of complex geometries based on fuzzy knowledge processing and computational geometry", *Integrated Computer-Aided Engineering*, Vol.2, 1995, pp.265-280.
4. Yagawa, G., Kawai, H., Yoshimura, S. and Yoshioka, A., "Mesh-invisible finite element analysis in a virtual reality environment", *Computer Modeling & Simulation in Engineering*, Vol.1, 1996, pp.289-314.

5. User's manual of MicroCADAM Helix, CADAM System, 1997.
6. Zadeh, L.A., "Fuzzy algorithms", *Information and Control*, Vol.12, 1968, pp.94-102.
7. Zadeh, L.A., "Outline of a new approach to the analysis of complex systems and decision process", *IEEE Transactions on Systems, Man and Cybernetics*, Vol.SMC-3, 1973, pp.28-44.
8. Asano, T., "Practical use of bucketing techniques in computational geometry", *Computational Geometry*, North-Holland, 1985, 153-195.
9. Sibson, R., "Locally equiangular triangulations", *The Computer Journal*, Vol.21, 1987, 243-245.
10. Bowyer, A., "Computing Dirichlet tessellations", *The Computer Journal*, Vol.24, 1981, pp.162-166.
11. Watson, D.F., "Computing the n-dimensional delaunay tessellation with application to Voronoi polytopes", *The Computer Journal*, Vol.24, 1981, pp.167-172.
12. Ludwig, R.A., Flaherty, J.E., Guerinoni, F., Baehmann, P.L. and Shephard, M.S. "Adaptive solutions of the Euler equations using finite Quadtree and Octree grids", *Computers & Structures*, Vol.30, 1988, pp.327-336.