Acceleration of Dynamic Bubble Mesh Generation for Large-Scale Model

Fumiaki Nobuyama, So Noguchi, and Hajime Igarashi
Graduate School of Information Science and Technology, Hokkaido University
Kita 14 Nishi 9, Kita-ku, Sapporo 060-0814, JAPAN
noguchi@ssi.ist.hokudai.ac.jp

***Abstract*— An automatic mesh generation method employing a dynamic bubble system can provide a high-quality mesh for electromagnetic finite element analysis. However, there is a problem that it takes very long time to compute bubbles' movement when a mesh with a huge number of elements is produced. Bubbles move by the force among them like the van der Waals force. When they are very far, the force can be ignored. Therefore, it is not necessary to calculate the force from all the bubbles.**

In this paper, we propose the subdivided dynamic bubble system. The proposed method divided an analysis region into some tetrahedral elements, and then the bubbles moves independently in the tetrahedral elements. As the result, the reduction of the computation time was achieved.

***Index Terms*—Dynamic bubble system, finite element analysis, mesh generation, parallel computing.**

I. Introduction

In recent years, Finite Element Analysis (FEA) is frequently done for design and performance survey of electromagnetic machines. As a preprocessing of FEA, mesh generation is necessarily required. It is well known that some simulation conditions depend on the accuracy and the computation time. One of such simulation conditions is mesh quality. That is, a mesh of good quality yields high analysis accuracy and short computation time. However, the mesh generation of very large-scale model is usually very laborious and time consuming. In order to solve this issue, a 3-D automatic mesh generation method adopting a dynamic bubble system has been proposed [1]-[4]. It composes a dynamic bubble system and a Delaunay division [5]. The dynamic bubble system generates a set of vertices inside an entire analysis domain, and then the Delaunay division completes connection of the vertices to make a mesh.

The dynamic bubble system proposed in [1] has high ability to generate a high-quality mesh. On the other hand, it has a problem that it takes long computation time, and it is difficult to deal with a very large-scale model. That is, it requires a large computational cost and a large memory to simulate bubbles' movement. The bubbles move according to forces acting on them. However, the force of one bubble is nearly zero from bubbles far from it. Accordingly, it is unnecessary to simultaneously move all the bubbles. To solve this problem, the proposed bubble system generates an initial coarse mesh, and it is generated from the nodes of the initial input data using the Delaunay division. The initial coarse mesh contains some tetrahedral elements, and the bubbles' movement is performed in the individual initial tetrahedral elements. By dividing the computation area, it is possible to reduce a used memory. Recently, a PC has a large memory, however the use of a GPU is proposed for acceleration of a mesh generation [6]. As of now the memory of the GPU is not so large.

In this paper, we propose a subdivided dynamic bubble system for large-scale model, and bubbles moves independently in each tetrahedral element of the initial coarse mesh. Moreover, the proposed method is parallelized with GPGPU in order to accelerate an automatic tetrahedral mesh generation. Here, the large-scale model means that a generated mesh has a large number of elements, more than one million.

II. Subdivided Dynamic Bubble System

A. Ordinary Dynamic Bubble System

The dynamic bubble system [1] is a physical model using multiple bubbles. It generates and moves many bubbles in an analysis domain, and computes a dense disposition of them as possible. The disposition of nodes obtained with the dynamic bubble system has a smooth variation of sparse and dense. Therefore, it is promised that a high-quality mesh is generated. Fig. 1(a) shows the procedure of the ordinary dynamic bubble system [1].

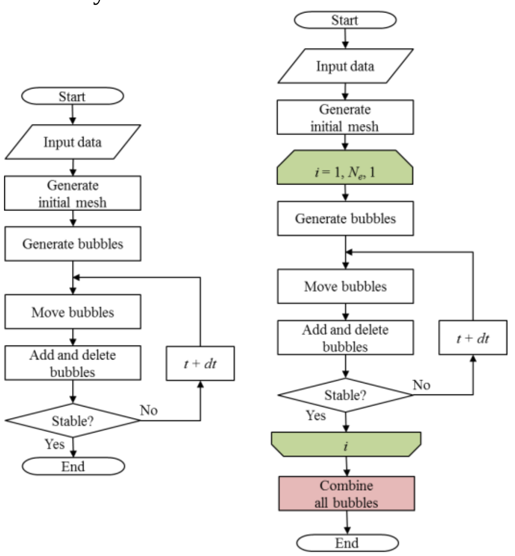
The ordinary dynamic bubble system has an issue that it takes long computation time and a large memory is consumed for a large-scale problem. Let the number of bubbles be N, and $N \times (N-1)$ computations is necessary in every time step. Consequently, the computation time explosively increases with increase of bubbles. In addition, since one bubble receives force from all the other bubbles, a common large memory use for storing the position and size of the bubbles. However, recently, large-scale FEAs with more than one million elements have been occasionally done; therefore the dynamic bubble system has to be improved to deal with more than one million elements at least with short time as possible.

B. Subdivided Dynamic Bubble System

In the proposed subdivided dynamic bubble system, firstly, bubbles are generated independently in each tetrahedral element of the initial coarse mesh. Then, all the generated bubbles move according to the force between the bubbles. The force is mathematically modeled by the van der Waals force.

Generally, remote bubbles far from one bubble are not considered to practically affect the one. Therefore, a force computation from all the bubbles wastes long time. In the proposed system, hence, the bubble movement is performed independently in each tetrahedral element of the initial coarse mesh. The initial coarse mesh is generated by connecting the nodes of the initial input data with the Delaunay division, and some tetrahedral elements are generated at first.

We propose a subdivided dynamic bubble system, which moves bubbles independently in each tetrahedron of the initial mesh. Fig. 1(b) shows the procedure of the subdivided dynamic bubble system.



(a) (b)

Fig. 1. Flowchart of (a) the ordinary dynamic bubble system and (b) the subdivided ordinary dynamic bubble system.

III. Comparison of Computation Time

The usefulness of the automatic mesh generation employing the subdivided dynamic bubble system is verified on an iron core model. The iron core model consisting of an iron core and a coil in 3-D space was meshed with a variety of number of elements. Fig. 2 shows the comparison of the computation time with the ordinary dynamic bubble system and the subdivided dynamic bubble system. Here, a reduction rate γ defined as follows is used as an indicator of acceleration.

$$\gamma = \frac{\left(t_{\mathrm{ord}} - t_{\mathrm{sub}}\right)}{t_{\mathrm{sub}}} \times 100 \,. \tag{1}$$

where t_{ord} are t_{sub} are the computation time in the case of the ordinary dynamic bubble system and the one of the subdivided dynamic bubble system, respectively.

The reduction rate is approximately 40%. However, when the number of bubbles is more than 500,000, the ordinary dynamic bubble system cannot converge in a realistic time. The reason is that it is necessary to access all over a wide common memory to obtain the position and size of bubbles. It takes a long time to access all over the wide common memory. However, the subdivided dynamic bubble system can converge since the memory accessed to calculate the force of one bubble is limited in a narrow area. The acceleration of the dynamic bubble system is achieved for the mesh generation of a large-scale model.

In order to aim to accelerate furthermore, the subdivided dynamic bubble system is parallelized with GPGPU [6]. Fig. 3 shows the comparison with the ordinary dynamic bubble system and the subdivided dynamic bubble system parallelized with GPGPU. The reduction rate is nearly 90% and much higher than that of the subdivided dynamic bubble system without GPGPU. The subdivided dynamic bubble system parallelized with GPGPU successfully divided a mesh with 1,000,000 bubbles.

IV. Conclusion

In this paper, we propose an automatic mesh generation using a subdivided dynamic bubble system and parallelize this method with GPGPU. As a result, we have achieved about 90% reduction rate adopting both subdivision and parallelization with GPGPU. In full paper, we want to indicate algorithm which we cannot explain in this paper.

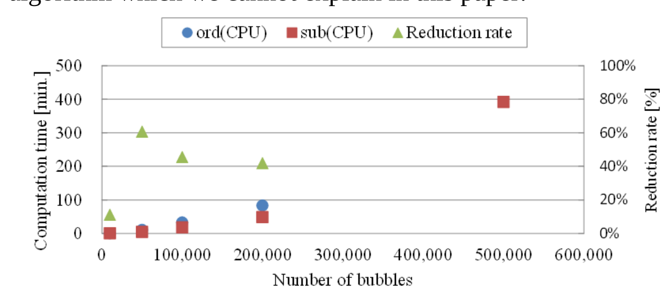


Fig. 2. Comparison of computation times with the ordinary dynamic bubble system and the subdivided dynamic bubble system (CPU computation).

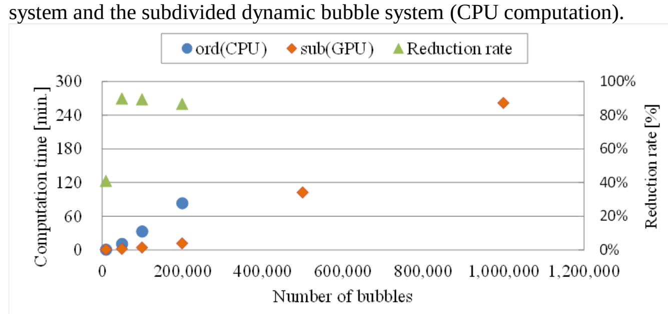


Fig. 3. Comparison of computation times with the ordinary dynamic bubble system with CPU and the subdivided dynamic bubble system with GPGPU.

References

[1] T. Yokoyama, V. Cingoski, K. Kaneda, and H. Yamashita, “3-D Automatic Mesh Generation for FEA Using Dynamic Bubble System,” *IEEE Trans. Magn.*, vol. 35, no. 3, pp. 1318-1321, May 1999.

[2] S. Nagakura, S. Noguchi, K. Kaneda, H. Yamashita, and V. Cingoski, “Automatic quadrilateral mesh generation for FEM using dynamic bubble system,” *IEEE Trans. Magn.*, vol. 37, no. 5, pp. 3522-3525, May 2001.

[3] Y. Marechal, D. Armand, and D. Ladas, “An adaptive remeshing technique ensuring high quality meshes,” *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 1222-1225, Jun. 2008.

[4] D. N. Shenton and Z. J. Cendes, “Three-Dimensional Finite Element Mesh Generation Using Delaunay Tessellation,” *IEEE Trans. Magn.*, vol. 21, no. 6, pp. 2535-2538, Nov. 1985.

[5] K. Shimada and D. C. Gossard, “Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing,” *ACM 3rd Symposium on Solid Modeling and Applications*, pp. 409-419, 1995.

[6] F. Nobuyama, S. Noguchi, and H. Igarashi, “The Parallelized Automatic Mesh Generation Using Dynamic Bubble System with GPGPU,” to be published in *IEEE Trans. Magn*.