Large-scale Parallel Magnetic Analysis using Hierarchical Domain Decomposition Tool with Refine Function

Shin-ichiro Sugimoto, Kohei Murotani, Hiroshi Kawai and Shinobu Yoshimura
School of Engineering, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
sugimoto@sys.t.u-tokyo.ac.jp, muro@save.sys.t.u-tokyo.ac.jp,
kawai@save.sys.t.u-tokyo.ac.jp, yoshi@sys.t.u-tokyo.ac.jp

Abstract — This paper deals with a large-scale 3D magnetostatic analysis using the Hierarchical Domain Decomposition Method (HDDM). To reduce time of generating large- or huge-scale meshes, the Refine tool that generates large- or huge-scale mesh from small- or medium-scale one in various parallel environments is considered. This parallelized Refine tool is embedded in hierarchical domain decomposition tool. As a result, time of generating mesh reduces drastically and huge-scale tetrahedral element mesh with 2 billion elements is successfully made in about 27 minutes with 4,096 cores (256 nodes) on the T2K Open Supercomputer. Furthermore, we confirm that a linear magnetostatic problem with 2.2 billion degrees of freedom is solved in about 4 hours on the T2K Open Supercomputer (256 nodes, 4,096 cores).

I. INTRODUCTION

Computational objects tend to be large and complex for numerical analyses recently. In addition, subdivision of the mesh is performed for the improvement of precision. Moreover, in case of coupled analyses with other phenomena, because the models of electromagnetic analyses include structures and space (air, vacuum or any liquid), they tend to become larger than those of structural, fluid and heat conduction analyses. Therefore, large-scale computations are increasingly important in electromagnetic field problems. To meet this requirement, we have already introduced the Hierarchical Domain Decomposition Method (HDDM) [1]-[2] to 3D non-linear magnetostatic problems [3]. And we have shown the possibility of soliving the non-linear magnetostatic problem with 1.2 billion Degrees of Freedom (DOF) [4]. It has been solved in about 3.8 hours with 256 nodes (4,096 cores) on the T2K Open Supercomputer [5].

Furthermore, increasing number of parallel processes by increasing cores of CPU and improvement of parallel computing technology makes large-scale analyses familiar. Computing time of analyzing medium- or large-scale problems is becoming shorter by development of parallel simulation software working efficiently in those parallel environments. Meanwhile, as the scale of analysis becomes large, time of generating meshes increases. Moreover, as computing time by simulation software becomes short, a ratio of time of generating meshes in analysis services increases in medium-scale, too. In this paper, we consider the generation of large-scale mesh from small-scale one by parallelized Refine tool. To verify effectiveness of our method, parallel magnetostatic analyses will be solved with meshes generated by Refine tool.

II. REFINE TOOL

In this paper, we consider the generation of tetrahedral element mesh for finite element method.

We use modules developed in ADVENTURE Project [6] for generation of mesh. First, triangular surface patches are generated from IGES data by ADVENTURE_TriPatch. Then, tetrahedral element mesh is generated from triangular surface patches automatically by ADVENTURE_TetMesh with Delaunay triangulation method. Here, because parallelization of the Delaunay triangulation method is difficult, it increase a ratio of time of generating meshes in analysis services Therefore, we consider parallel Refinement of mesh.

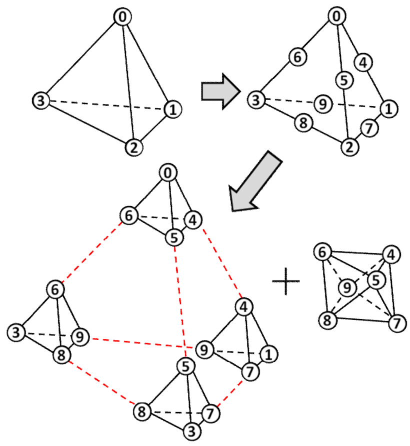

Fig. 1. Division of a tetrahedral element.

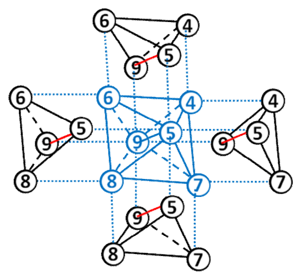

Fig. 2. Division of an octahedral.

A tetrahedral element can be divided into four small tetrahedral and an octahedral (Fig. 1). Then, the octahedral can be divided into four tetrahedral (Fig. 2). As a result, a tetrahedral element is divided into eight small tetrahedral elements. Because this operation can be executed independently in each element, parallelization of the Refine tool is easy. This parallelized Refine tool is embedded in ADVENTURE_Metis. ADVENTURE_Metis is a domain decomposition tool for HDDM. And it works in various parallel environments.

III. NUMERICAL RESULTS

In this chapter, we compare time of generating meshes by the new method (generating small mesh by ADVENTURE_TriPatch and ADVENTURE_TetMesh, Refining mesh by ADVENTURE_Metis with Refine tool) with those by the previous method (using only ADVENTURE_TriPatch and ADVENTURE_TetMesh).

A. Model

TEAM Workshop Problem 20 [7] is considered. This model consists of a center pole, a yoke and a coil. The center pole and the yoke are made of SS400, and the coil is made of polyimide electric wire. The magnetic reluctivity is, for simplicity, a positive constant in each element, the values are $1/(4\pi\times10^{-7})$ [m/H] in the region of air and coil, and 100 [m/H] in the region of center pole and yoke. The electric current in the coil is 1,000 [A].

B. Generating mesh

Generation of triangular surface patches and generation of tetrahedral mesh are performed with one core of a workstation that consists of 4 CPUs of Intel Xeon X7460 (2.66GHz/L2 9MB/L3 16MB/Hexa Core, 24 cores, 128GB Memory). The Refine tool is performed by T2K Open Supercomputer that is in the University of Tokyo and named HA8000. A node of HA8000 consists of 4 CPUs of AMD Quad Core Opteron 8356 (2.3GHz, 16 cores). We use 64, 1,024 or 4,096 cores (4, 64 or 256 nodes).

TABLE. I.
REFINE TIMES, NUMBERS OF ELEMENTS AND NODES, AND DOF.

	Refine	Elements	Nodes	DOF
Mesh(0)	–	471,541	639,724	559,848
Mesh(1R)	1	3,772,328	5,085,503	4,445,779
Mesh(2R)	2	30,178,624	40,484,541	35,399,038
Mesh(3R)	3	241,428,992	322,937,721	282,453,180
Mesh(4R)	4	1,931,431,936	2,579,465,969	2,256,528,248
Mesh(1N)	none	4,127,720	5,520,421	4,841,610
Mesh(2N)	none	33,060,941	43,918,710	38,560,173
Mesh(3N)	none	262,076,977	347,025,463	304,833,522
Mesh(4N)	none	–	–	–

TABLE. II. TIME OF GENERATING MESHES.

	Refine	TriPatch, TetMesh [s]	Refine tool		Total [s]
			cores	Time [s]	
Mesh(0)	–	86	–	–	86
Mesh(1R)	1	86	64	22	108
Mesh(2R)	2	86	1,024	120	206
Mesh(3R)	3	86	1,024	340	426
Mesh(4R)	4	86	4,096	1,507	1,593
Mesh(1N)	none	437	–	–	437
Mesh(2N)	none	5,571	–	–	5,571
Mesh(3N)	none	171,876	–	–	171,876
Mesh(4N)	none	–	–	–	–

First, a small-scale mesh that has about 500 thousand elements is generated (Mesh(0)). Then, Refinement is executed up to 4 times with Mesh(0) (Mesh(1R) – Mesh(4R)). Meanwhile, meshes of the same scale as Mesh(1R) – Mesh(4R) are generated by the previous method (Mesh(1N) – Mesh(4N)). TABLE. I shows numbers of elements and nodes, and Degrees Of Freedom (DOF) of the A method [3]. TABLE. II shows time of generating meshes. "TriPatch, TetMesh" means time of generating triangular surface patches and tetrahedral meshes by the previous method. Mesh(1R) – Mesh(4R) are generated from Mesh(0) by Refine tool. Therefore, their "TriPatch, TetMesh" are the same as that of Mesh(0). The new method is faster than the previous method. Moreover, as the scale of mesh becomes large, difference between the new method and the previous method becomes large.

Furthermore, Mesh(4N) could not be generated, because number of nodes was more than 2^{31} that is the limit of 32bit signed integer type. However, in case of Mesh(4R), because the Refine tool works in parallel environments, number of nodes in each process was less than 2^{31}. Therefore, we have succeeded to generate mesh that has more than 2^{31} nodes.

C. Magnetostatic Analysis

In this section, we consider linear magnetostatic analyses [4] of Mesh(1R) – Mesh(4R). A simplified block diagonal scaling is used as the preconditioner in the CG procedure of the interface problem. The convergence criterion δ of the interface problem of the HDDM is set to 10^{-4}. Computations are performed by T2K Open Supercomputer.

TABLE. III shows the computation time and the amount of memory. Then, the linear magnetostatic analysis of the model with 2.2 billion DOF is successfully solved in about 4 hours with 4,096 cores (256 nodes).

TABLE. III. COMPUTATION TIME AND AMOUNT OF MEMORY.

	cores	Time [s]	Memory per core [MB]	Total Memory Usage [GB]
Mesh(1R)	64	176	67.1	4.19
Mesh(2R)	1,024	248	26.7	26.7
Mesh(3R)	1,024	3,343	211	211
Mesh(4R)	4,096	14,310	423	1,690

IV. REFERENCES

[1] A. Quarteroni and A. Vali, *Domain Decomposition Methods for Partial DifferentialEequations*, CLARENDON PRESS・OXFORD, 1999.

[2] R. Shioya and G. Yagawa, "Iterative domain decomposition FEM with preconditioning technique for large scale problem", *ECM'99 Progress in Experimental and Computational Mechanics in Engineering and Material Behaviour*, pp.255-260, 1999.

[3] H. Kanayama, H. Zheng and N. Maeno, "A domain decomposition method for large-scale 3-D nonlinear magnetostatic problems", *Theoretical and Applied Mechanics*, 52, pp.247-254, 2003.

[4] S. Sugimoto, M. Ogino, H. Kanayama and S. Yoshimura, "Introduction of a Direct Method at Subdomains in Non-linear Magnetostatic Analysis with HDDM", *BWCCA 2010*, (Full paper is in CD-ROM), 2010.

[5] T2K Open Super Computer HP: http://www.open-supercomputer.org

[6] ADVENTURE Project HP: http://adventure.sys.t.u-tokyo.ac.jp/

[7] IEEJ, *IEEJ Technical Report No.486*, IEEJ, (in Japanese), 1994.