

# Application of the Sparse Direct Solver to a High-frequency Electromagnetic Field Analysis Code to Improve Convergence and Save Memory

K. Ohnaka<sup>1</sup>, S. Goto<sup>2</sup>, T. Murayama<sup>2</sup>, A. Takei<sup>3</sup>

<sup>1</sup>Interdisciplinary Graduate School of Agriculture and Engineering, University of Miyazaki, Japan,  
hl16004@student.miyazaki-u.ac.jp

<sup>2</sup>Graduate School of Engineering, The University of Tokyo, Japan

<sup>3</sup>Faculty of Engineering, University of Miyazaki, Japan

The high-frequency electromagnetic field analysis code being developed by the authors is a finite element analysis software that can be run on parallel computer systems. Although this code is capable of detailed, fast, and efficient finite element analysis of large-scale electromagnetic problems using the Iterative Domain Decomposition Method (IDDM) and the corresponding parallel distributed processing environment, it still requires a large number of iterative computation trials and computation time. In this study, a sparse direct solver, which can be set and solved with larger subdomain sizes, is applied to the subdomain problem to reduce the load on the iterative solver and improve convergence. In addition, this improvement also significantly reduced memory usage.

*Index Terms*—Electromagnetic fields, Finite element analysis, Sparse matrices

## I. INTRODUCTION

The authors are currently conducting research and development on large-scale finite element analysis of high-frequency electromagnetic fields from several MHz to several GHz using parallelization techniques based on the Iterative Domain Decomposition Method (IDDM) [1].

Solving the vector wave equation in a high-frequency electromagnetic field using the finite element method involves solving a large set of linear equations. It is difficult to solve all of them by direct methods due to physical limitations of hardware, mainly memory. In addition, the iterative method alone causes a situation where the calculation is not completed due to poor convergence characteristics of high-frequency problems such as eigenvalues near zero. Therefore, based on IDDM, this method applies the Conjugate Orthogonal Conjugate Gradient (COCG) method, an iterative method, to the interface problem and a direct method based on the modified Cholesky decomposition, to the subdomain problem, thereby improving convergence while reducing memory usage [2]. Subdomain problems can be computed independently because they do not depend on each other, and highly efficient parallel computation is possible.

However, the number of iterations and memory usage is still high, and improvements are needed to stabilize the solver and save memory in order to solve higher frequencies and more detailed and extensive models. In this study, the sparse direct solver, which has become more reliable and stable due to recent developments, is applied to a direct method solver for solving subdomain problems. Memory usage is reduced by optimized algorithms and matrix storage formats for sparse matrix calculations, and it allows for partitioning the domain into larger sizes, it is expected to reduce the load on the iterative method solver and improve convergence. This will enable the calculations to be faster and more stable with less memory than ever before.

## II. GOVERNING EQUATIONS AND ALGORITHM

In our code, the full-wave analysis based on an  $E$ -method [3] is considered.  $\mathbf{E}_h$  and  $\mathbf{J}_h$  are finite element approximations of electric field  $\mathbf{E}$  [V/m] and current density  $\mathbf{J}$  [A/m<sup>2</sup>], respectively. The permeability is given by  $\mu = \mu_0 \mu_r$  [H/m],  $\mu_0$  is the vacuum permeability [H/m], and  $\mu_r$  is the relative permeability. The complex permittivity is given by  $\varepsilon = \varepsilon_0 \varepsilon_r - \sigma/j\omega$  [F/m],  $\varepsilon_0$  is the vacuum permittivity [F/m],  $\varepsilon_r$  is the relative permittivity, and  $\omega$  is the angular frequency [rad/s]. The following (1) is the finite element equation to be solved.

$$\iiint_{\Omega} (1/\mu) \text{rot} \mathbf{E}_h \cdot \text{rot} \mathbf{E}_h^* dv - \omega^2 \iiint_{\Omega} \varepsilon \mathbf{E}_h \cdot \mathbf{E}_h^* dv = j\omega \iiint_{\Omega} \mathbf{J}_h \cdot \mathbf{E}_h^* dv \quad (1)$$

The equation contains complex numbers and becomes a complex symmetric matrix. The electric field  $\mathbf{E}$ , which is unknown, is obtained using the COCG method. (1) is rewritten as  $Ku = f$  by the coefficient matrix  $K$ , the unknown vector  $u$ , and the right-hand side vector  $f$ . Next, domain  $\Omega$  is divided into  $N$  subdomains (2). (3) and (4) are obtained from (2).

$$\begin{bmatrix} K_{II}^{(1)} & 0 & 0 & K_{IB}^{(1)} R_B^{(1)T} \\ 0 & \ddots & 0 & \vdots \\ & & K_{II}^{(N)} & K_{IB}^{(N)} R_B^{(N)T} \\ R_B^{(1)} K_{IB}^{(1)T} & \cdots & R_B^{(N)} K_{IB}^{(N)T} & \sum_{i=1}^N R_B^{(i)} K_{BB}^{(i)} R_B^{(i)T} \end{bmatrix} \begin{bmatrix} u_I^{(1)} \\ \vdots \\ u_I^{(N)} \\ u_B \end{bmatrix} = \begin{bmatrix} f_I^{(1)} \\ \vdots \\ f_I^{(N)} \\ f_B \end{bmatrix} \quad (2)$$

$$K_{II}^{(i)} u_I^{(i)} = f_I^{(i)} - K_{IB}^{(i)} u_B \quad (i = 1, \dots, N) \quad (3)$$

$$\begin{aligned} & \left\{ \sum_{i=1}^N R_B^{(i)} \{ K_{BB}^{(i)} - K_{IB}^{(i)T} (K_{II}^{(i)})^{-1} K_{IB}^{(i)} \} R_B^{(i)T} \right\} u_B \\ & = \sum_{i=1}^N R_B^{(i)} \{ f_B^{(i)} - K_{IB}^{(i)T} (K_{II}^{(i)})^{-1} f_I^{(i)} \} \end{aligned} \quad (4)$$

where  $f_B^{(i)}$  is the right-hand vector for  $u_B$ , and  $(K_{II}^{(i)})^{-1}$  is the inverse matrix of  $K_{II}^{(i)}$ . Equation (4) is referred to as an interface problem and is an equation for satisfying the continuity between subdomains in the DDM [2][3].

### III. INTRODUCING THE SPARSE DIRECT SOLVER

#### A. Sparse direct solver

A sparse direct solver is optimized for sparse matrices, it can solve large-scale sparse matrix problems quickly and efficiently. It also suitable for performing complex symmetric matrix operations required for subdomain calculations in our electromagnetic field analysis code. In this study, Intel MKL PARDISO is applied as the solver, and efficient computation can be expected by passing matrix and vector data along with multiple parameters that can be set according to the computational environment.

#### B. Matrix storage format

In conventional our code, for subdomain problems, the matrix data is converted and stored in skyline format and computed by a direct method based on the modified Cholesky decomposition [3]. Although this format considerably reduces the storage of zero elements by storing from the first nonzero element in each row to the diagonal elements, it is not a perfect nonzero storage and is still wasteful in terms of memory efficiency. In contrast, the compressed sparse row (CSR) format used in the sparse direct solver PARDISO records the column indices of the nonzero elements in each row and extracts and stores only the non-zero elements, resulting in complete nonzero storage that does not contain any zero elements and highly memory-efficient.

In addition, our code stores matrix data in a format similar to the CSR format during domain partitioning. Therefore, it is relatively easy to implement a solver that replaces the CSR format with PARDISO as input, and is expected to benefit from the high calculation speed of PARDISO while minimizing the impact on the overall code.

### IV. EVALUATION OF IMPROVED SOLVER

For the performance evaluation of the improved code, we change the subdomain size to 300, 1,000, 3,000, and 10,000 and compared the number of iterations, time required for the analysis, and memory usage. The computing environment is a shared memory machine with an AMD EPYC 9654 CPU and 1.5 TB of memory. The Calculation model and settings are based on the TEAM29 [2][4]. The analysis conditions are shown in Table 1 and the calculation results in Table 2.

TABLE I  
THE ANALYSIS CONDITIONS

| Computer                       | Shared memory machine |
|--------------------------------|-----------------------|
| Num. of parallels              | 72                    |
| Frequency [MHz]                | 120                   |
| Num. of elements               | 16,770,964            |
| Convergence criterion $\kappa$ | $1.0 \times 10^{-7}$  |

TABLE II  
THE CALCULATION RESULTS

| Sub-domain size | Direct method based on the modified Cholesky decomposition |            |                   | Sparse direct solver PARDISO |            |                   |
|-----------------|------------------------------------------------------------|------------|-------------------|------------------------------|------------|-------------------|
|                 | Num. of iterations                                         | Time [sec] | Memory usage [GB] | Num. of iterations           | Time [sec] | Memory usage [GB] |
| 300             | 31,931                                                     | 7,096      | 29.5              | 32,676                       | 12,048     | 47.1              |
| 1,000           | 25,555                                                     | 6,739      | 43.6              | 26,014                       | 9,244      | 44.3              |
| 3,000           | 20,470                                                     | 11,702     | 72.0              | 20,442                       | 11,030     | 52.7              |
| 10,000          | 18,003                                                     | 21,573     | 151               | 17,736                       | 17,492     | 72.0              |

Both the conventional version based on the modified Cholesky decomposition for subdomains solving and the improved version using PARDISO showed that as the subdomain size is increased, the number of iterations decreases and convergence improves in both cases.

In a comparison of the conventional and improved versions, when the subdomain size was increased toward stabilization, the conventional version was faster up to the subdomain size of 1,000, but the speeds were the same at the subdomain size of 3,000, and the improved version was faster at the subdomain size of 10,000.

The memory usage of both the conventional and improved versions increased as the subdomain size was increased, but the increase was smaller in the improved version, with the conventional version consuming 151 GB and the improved version 72 GB, or 53% less memory, for the subdomain size of 10,000.

### V. CONCLUSION

In this study, in the development of high-frequency electromagnetic field analysis code, in order to reduce the load and stabilize convergence by reducing the number of iterations, we applied the sparse direct solver to subdomain solver. Although the conventional version was optimal in terms of computation time and memory usage for the model used in the verification, it is expected that the improved version with a larger subdomain size will be effective in situations where improved convergence is essential for more complex and high-frequency large-scale problems. This improvement has several advantages, including the possibility of solving previously unsolvable problems by improving convergence and allowing analysis to be performed with limited computer resources due to reduced memory usage. there is a major benefit in solving large problems.

### REFERENCES

- [1] J. Mandel, "Balancing domain decomposition", *Communications on Numerical Methods in Engineering*, vol.9, pp.233-241, 1993.
- [2] A. Takei, S. Yoshimura, H. Kanayama, "Large-scale full wave analysis of electromagnetic field by hierarchical domain decomposition method", *Computer Modeling in Engineering and Sciences*, vol.40, no.1, pp.63-81, 2009.
- [3] A. Takei, M. Ogino, and S. Sugimoto, "High-frequency electromagnetic field analysis by COCR method using anatomical human body models", *IEEE Transactions on Magnetics*, Vol.54, No.3, 2018. DOI: 10.1109/TMAG.2017.2756098.
- [4] Y. Kanai, "Description of TEAM Workshop Problem 29: Whole Body Cavity Resonator", TEAM Workshop, Tucson, AZ, USA, 1998.