A Coarse Matrix Iterative Solver for Magnetostatic Domain Decomposition Analysis

Hiroshi Kanayama[1], Masao Ogino[2], and Shin-ichiro Sugimoto[3]

[1]Japan Women's University, Bunkyo-ku, Tokyo112-8681, Japan, kanayamah@fc.jwu.ac.jp
[2]Nagoya University, Chikusa-ku, Nagoya464-8601, Japan, masao.ogino@cc.nagoya-u.ac.jp
[3]Tokyo University of Science, Suwa, Chino, Nagano391-0292, Japan, sugimoto@rs.tus.ac.jp

An iterative domain decomposition method is proposed for numerical analysis of 3-Dimensional (3D) linear magnetostatic problems taking the magnetic vector potential as an unknown function. The iterative domain decomposition method is combined with the Preconditioned Conjugate Gradient (PCG) procedure and the Hierarchical Domain Decomposition Method (HDDM) which is adopted in parallel computing. Our previously employed preconditioner was the Neumann-Neumann (NN) preconditioner. Numerical results showed that the method was only effective for smaller problems. In this paper, we consider its improvement with the Balancing Domain Decomposition DIAGonal scaling (BDD-DIAG) preconditioner. Specially, the coarse matrix solver is changed to an iterative solver and numerical results show that the incomplete CG iterative solver like CG(10) is more effective than the standard approach.

***Index Terms*— Domain decomposition method, Large-scale magnetostatic problems, Preconditioning, The coarse problem**

I. Introduction

THIS DOCUMENT deals with a coarse matrix iterative solver for magnetostatic domain decomposition analysis.

II. The Interface Problem

The present section assumes that the magnetic reluctivity ν is given for simplicity. Then, the linear magnetostaic problem (1) without the Coulomb gauge condition is considered on a polyhedral domain Ω with the boundary $\partial\Omega$:

$$rot(\nu \, rot \, A) = J \quad in \, \Omega, \tag{1a}$$

$$A \times n = 0 \quad on \, \Gamma_E, \tag{1b}$$

$$(\nu \, rot \, A) \times n = 0 \quad on \, \Gamma_N, \tag{1c}$$

where A is the magnetic vector potential and J is an electric current density. Assume that the boundary $\partial\Omega$ consists of two disjoint parts Γ_E and Γ_N. Let n be the unit outward normal vector to the boundary.We then consider a non-overlapping partition of the domain Ω, consisting of subdomains, also called substructures $\{\Omega^{(i)}\}_{i=1,\dots N}$. We also define the interface as

$$\Gamma \equiv \cup_{i=1}^{N} \partial\Omega^{(i)} \setminus \Gamma_E, \tag{2}$$

where N is the number of subdomains.

For the given ν, the finite element discretization of (1) gives a symmetric linear system. The Degrees of Freedom (DOF) inside subdomains are eliminated in parallel by a static condensation [1]. We are then left with a linear system involving only DOF on Γ. If a local vector $(u^{(i)})$ in $\Omega^{(i)}$ is divided into two subvectors; DOF $(u_I^{(i)})$ corresponding to edges inside $\Omega^{(i)}$ and DOF $(u_B^{(i)})$ on $\partial\Omega^{(i)} \setminus \Gamma_E$, respectively, the local stiffness matrix of $K^{(i)}$ can be written as

$$K^{(i)} = \begin{pmatrix} K_{II}^{(i)} & K_{IB}^{(i)} \\ K_{IB}^{(i)T} & K_{BB}^{(i)} \end{pmatrix}. \tag{3}$$

Let $W^{(i)}$ be the space of interface DOF for the subdomain $\Omega^{(i)}$ and W be the space of all DOF on Γ. After eliminating DOF inside subdomains, the original problem reduces to a problem with a smaller dimension;

$$Su_B = g \, , \quad u_B \in W \, , \tag{4}$$

where $S = \sum_{i=1}^{N} R_B^{(i)} S^{(i)} R_B^{(i)T}$ is the global Schur complement matrix related to Γ and g is the resultant right hand side vector. We define the operators:

$$S: W \to W, \;\; S^{(i)}: W^{(i)} \to W^{(i)}, \qquad R_B^{(i)}: W^{(i)} \to W \, .$$

$R_B^{(i)T}$ is the transpose of $R_B^{(i)}$. The local Schur complement $S^{(i)}$ is defined as

$$S^{(i)} \equiv K_{BB}^{(i)} - K_{IB}^{(i)T} K_{II}^{(i)\dagger} K_{IB}^{(i)} \, . \tag{5}$$

Here, $\left(K_{II}^{(i)}\right)^{\dagger}$ denotes the generalized inverse of $K_{II}^{(i)}$ [1]. The problem (4) is solved by a PCG method which requires to solve the following auxiliary problem:

$$Mz = r \tag{6}$$

where r is the residual of (4) and M is a preconditioner. In the preliminary research for a perturbation problem [2], we tried to implement the Neumann-Neumann (NN) preconditioner without a coarse problem. Due to the absence of the coarse problem, its effectiveness was restricted to problems with small number of subdomains. BDD; the Neumann-Neumann preconditioner with a coarse problem, or BDD-DIAG; the simplified diagonal scaling preconditioner with a coarse problem, is the present challenge of this research.

III. Change of the Coarse Matrix Solvers

In this section, change of the coarse matrix solver is mentioned. To solve the coarse problems in *Step 1* and *Step 5* of Chapter 5 [3], parallel skyline solvers were used in [4]-[6]. However, in magnetostatic analysis, the coarse matrix becomes singular and the paralell skyline solver can not be used. In this paper, we use the library Lis [7] as follows:

Compressed Sparse Row (CSR) format is used. CG with no preconditioner is also used. The tolerance value is set to be 1.0e-03. These are standard conditions. Exceptional cases are separately mentioned. The initial vector for the iterative solver is always zero, which is a default value in the Lis library. Generally these solvers require much computaional time at present. The shortnening of computaional time is absolutely necessary, which remains a future problem in this paper. The format change to Lis from the parallel skyline solver is relatively easy because the CSR format is familiar with the skyline format.

IV. Numerical Results

A. *A shaft Problem*

An axi-symmetric shaft model comes from Fuji Electric Co., Ltd. . As a preliminary result, we compared convergence histories with the simplified diagonal scaling (diag) and without preconditioning (none) for a simple shaft perturbed problem [2] (the perturbation parameter; epsilon = 10^{-4}) by one core computation on the first author's PC (Let's note with Cygwin). Numbers of parts, subdomains and elements are 1, 130 and 13,389, respectively. The interface DOF is 4,828. The regularized value α was suitably set to be 10^{-5} [2]. An iterative solver (ICCG with a shift value) [2] was used as the subdomain solver. Iteration counts until the 10^{-5} relative residual were 103 for the simplified diagonal scaling (diag) and 1,885 without preconditioning (none), respectively. diag was 18 times faster than none in iteration counts and 13 times faster in computational time. Very interestingly, BDD-DIAG by the second author showed almost the same iteration count as diag. On the other hand, the original BDD and NN preconditioners produced no effective results. Two important remarks should be added. In these numerical results, to prepare preconditioners of (6), the global Schur complement matrix was constructed by following the library TryDDM in [8] and using the same global Schur complement matrix, the singular problem without the perturbation term was also solved by BDD-DIAG, whose iteration counts were again almost the same as diag. This may mean that the role of basic preconditioners (diag vs. NN) is very important and produces a big difference between BDD-DIAG and BDD. Furthermore, Table 1 shows change of iteration counts depending on the Lis solver options. For example, CG(10) means that upper limit of the CG iteration count of the coarse problem is limited to 10.

B. *The TEAM Workshop Problem 20*

We have compared convergence histories with the simplified diagonal scaling (diag), BDD-DIAG and without preconditioning (none) for the TEAM Workshop Problem 20 [9] (the perturbation parameter; epsilon = 0) by one core computation on Intel Core i5-4460 with CentOS. Numbers of parts, subdomains, elements and the interface DOF are shown in [3]. The regularized value α and others are the same as in the previous example. Very interestingly, BDD-DIAG by the secomd author again shows almost the same iteration count as diag. On the other hand, the original BDD and NN preconditioners also produced no effective results.

TABLE I
ITERATION COUNTS FOR SEVERAL OPTIONS OF THE LIS SOLVER

Options	Iteration Counts	Explanations
diag	103	Default option of the solver in ADVENTURE_Magnetic
CG (10)	103	Upper limit of the CG iteration count is limited to 10.
CG (30)	104	Upper limit of the CG iteration count is limited to 30.
CG (1.0e-03 convergence)	121	Relative residual 1.0e-03 is kept.

C. *An IEEJ Model*

We have also compared convergence histories with the simplified diagonal scaling (diag), BDD-DIAG and without preconditioning (none) for an IEEJ model established by IEEJ (the perturbation parameter; epsilon = 0) by one core computation on Intel Core i5-4460 with CentOS. It is noted that this model includes the natural boundary condition (1c) but that the previous examples only include the essential boundary condition (1b).

Numbers of parts, subdomains, elements and the interface DOF are shown in [3]. Others are the same as in the previous examples. Very interestingly, BDD-DIAG again shows almost the similar iteration count as diag. On the other hand, the original BDD and NN preconditioners also produced no effective results.

References

[1] J. Mandel, "Balancing domain decomposition," Communications on Numerical Methods in Engineering **9**, pp. 233-241 (1993)

[2] H. Kanayama, A.M.M. Mukaddes, M. Ogino, and S. Sugimoto, "A domain decomposition preconditioner for large scale 3-D magnetostatic analysis," Proceedings of Joint Technical Meeting on Static Apparatus and Rotating Machinery, IEEJ, pp. 21-26 (2005)

[3] H. Kanayama, M. Ogino, S. Sugimoto, K. Yodo, and H. Zheng, "On the coarse matrix solver of preconditioners for msgnetostattic domain decomposition analysis," IEEJ, **137**-3, pp.179-185 (2017)

[4] M.Ogino, R.Shioya, and H.Kanayama, "An inexact balancing preconditioner for large-scale structural analysis," *Journal of Computational Science and Technology*, **2**-1, pp.150-161 (2008)

[5] A.M.M.Mukaddes, M. M.Ogino, H.Kanayama and R.Shioya, "A scalable balancing domain decomposition based preconditioner for large scale heat transfer problems," *JSME International Journal*, pp.533-540 (2006)

[6] Q.Yao, H.Kanayama, H.Notsu and M.Ogino, "Balancing domain decomposition for non-stationary incompressible flow problems using a characteristic-curve method," *Journal of Computational Science and Technology*, **4**(2), pp.121-135 (2010)

[7] The Scalable Software Infrastructure Project, Lis: Library of Iterative Solvers for Linear Systems, Version 1.5.66 (2016)

[8] M.Ogino and R.Shioya, "Development of a library of iterative solvers based on the domain decomposition method," *A keynote presentation in ICCM 2015*, Auckland, NZ, (2015)

[9] TEAM Workshop Problem 20, http://www.compumag.org/jsite/team.html