A relation of preconditioners for magnetostatic domain decomposition analysis

Hiroshi Kanayama¹, Masao Ogino², Shin-ichiro Sugimoto³, Kaworu Yodo⁴

¹ Department of Mathematical and Physical Sciences, Japan Women’s University
² Department of Information Systems, Daido University
³ Department of Mechanical Engineering, Hachinohe Institute of Technology
⁴ Technology & Development Group, Insight Inc.

*kanayamah@fc.jwu.ac.jp

Received: July 13, 2020; Accepted: December 31, 2020; Published: January 20, 2021

Abstract. A relation of preconditioners of domain decomposition method is shown for numerical analysis of 3-Dimensional (3D) 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 small number subdomain problems. In this paper, we consider its improvement making use of the Balancing Domain Decomposition DIAGONal scaling (BDD-DIAG) preconditioner and show the asymptotic equivalence between BDD-DIAG and the simplified diagonal scaling (diag) preconditioner, which is derived from the following numerical evidences. Finally, nonlinear processing is also tried for the first time.

Keywords: BDD-DIAG, PCG, HDDM, Preconditioning, The coarse problem

1. Introduction

Among various parallel computing techniques, the Domain Decomposition Method (DDM) is a promising one in solving 3-Dimensional (3D) engineering problems. In this paper, we concentrate our concern on only non-overlapping DDM for convenience. Then, DDM decomposes the whole domain into independent subdomains. Different methods can be chosen for solving the equations employed on these subdomains. On the other hand, DDM needs an iteration process in solving the interface problem to obtain the final solution. In this work, for simplicity, we only consider one part problem. Owing to our previous technique [1], an iterative
method can be used to solve the matrix equations employed on subdomains and a Preconditioned Conjugated Gradient (PCG) method can be used for the interface problem. In the overall parallel computing, the Hierarchical Domain Decomposition Method (HDDM) [2] is used. The performance of a perturbed problem with the direct solver was successfully compared with the Incomplete Cholesky Conjugate Gradient (ICCG) method to solve subdomain problems. Our employed CG preconditioner was the Neumann-Neumann (NN) one. For small number of subdomains, this preconditioner was suitable. But due to the absence of a coarse problem, the convergence with the preconditioner decayed rapidly for large number of subdomains. No investigation of the sophisticated preconditioner for large scale 3D magnetostatic problems with large number of subdomains produced effective results so far. Under such situation, our latest and limited numerical results show that only a BDD variant (BDD-DIAG) is effective and that BDD and NN are not so in magnetostatic problems. [3] However, our numerical results were only shown for one part problem because of explicit Schur complement matrix construction. Towards multi-part problems, we are now testing the processing without explicit construction of Schur complement matrix in this research, though only one part problem is considered for simplicity.

This paper is arranged as follows. The formulation of magnetostatic problems is summarized in Section 2. The interface problem is also summarized in Section 3. Section 4 briefly represents the NN preconditioner and is followed by the BDD and BDD-DIAG preconditioners in Section 5. Section 6 is devoted to show the asymptotic equivalence between the Balancing Domain Decomposition DIAGonal scaling (BDD-DIAG) preconditioner and the simplified diagonal scaling (diag) preconditioner which is followed by a nonlinear approach and change of the coarse matrix solver in Section 7. Section 8 shows numerical results including nonlinear processing without explicit construction of Schur complement matrix for a series of magnetostatic test problems and concluding remarks are given in Section 9.

2. Formulations

In this section, we summarize the 3D magnetostatic analysis [1, 3–5] on a polyhedral domain $\Omega$ with the boundary $\partial \Omega$. Assuming that the boundary $\partial \Omega$ consists of two disjoint parts, we firstly consider the magnetostatic problem with the Coulomb gauge condition where the magnetic vector potential is unknown, an electric current density is divergence-free, and the magnetic reluctivity is assumed to be a positive scalar and that depends on the absolute value of the magnetic flux density in the magnetic body. Namely, we consider the B-H curve in this paper. Then we consider suitable function spaces to derive the weak formulations where the Lagrange multiplier is introduced to consider the Coulomb gauge condition. It is easy to see the Lagrange multiplier is uniformly zero in $\Omega$, because the electric current density is divergence-free. After
decomposing $\Omega$ into a union of tetrahedral elements, we consider a finite element approximation. The magnetic vector potential is approximated by the first order Nedelec element of simplex type, and the Lagrange multiplier by the conventional piecewise linear tetrahedral element. It is noted that a corrected electric current density is considered with consideration of the continuity. [3] Thanks to this corrected electric current density, we can show that the approximation of the Lagrange multiplier is also uniformly zero in $\Omega$. By the formal elimination of the Lagrange multiplier, we have a nonlinear system, since the magnetic reluctivity is a positive scalar.

3. The interface problem

The present section summarizes the interface problem in [3], assuming that the magnetic reluctivity is given until Section 7 for simplicity. We consider a non-overlapping partition of the domain $\Omega$, consisting of subdomains, also called substructures $\{\Omega^{(i)}\}_{i=1}^{N}$. We also define the interface as

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

(1)

where $N$ is the number of subdomains and $\Gamma_E$ denotes the essential boundary. The Degrees of Freedom (DOF) inside subdomains are eliminated in parallel by a static condensation. We are then left with a linear system involving only DOF on $\Gamma$. 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 $K^{(i)}$ can be written as

$$K^{(i)} = \begin{pmatrix} K_H^{(i)} & K_{HB}^{(i)} \\ K_{IH}^{(i)} & K_{BB}^{(i)} \end{pmatrix}.$$  

(2)

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

$$Su_B = g, \quad u_B \in W,$$  

(3)

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

\[ S: W \rightarrow W, \ S^{(i)}: W^{(i)} \rightarrow W^{(i)}, \ R_{B}^{(i)}: W^{(i)} \rightarrow W. \]

\( R_{B}^{(i)T} \) is the transpose of \( R_{B}^{(i)} \). The local Schur complement \( S^{(i)} \) is defined as

\[
S^{(i)} = K_{BB}^{(i)} - K_{iB}^{(i)T} \left( K_{II}^{(i)} \right)^\dagger K_{iB}^{(i)}. \tag{4}
\]

Here, \( \left( K_{II}^{(i)} \right)^\dagger \) denotes the generalized inverse of \( K_{II}^{(i)} \) [6]. The problem (3) is solved by a PCG method which requires to solve the following auxiliary problem:

\[
Mz = r \tag{5}
\]

where \( r \) is the residual of (3) and \( M \) is a preconditioner. In the preliminary research for a perturbation problem [1], we tried to implement the Neumann-Neumann preconditioner without a coarse problem. Due to the absence of the coarse problem, its effectiveness was restricted to problems with a 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.

4. The Neumann-Neumann preconditioner

This section briefly explains the NN preconditioner [6–10] which basically involves the solution of local problems.

An important choice for the NN preconditioner is the selection of weight matrices \( D^{(i)} \) that form a partition of unity on the interface space. A straightforward choice for \( D^{(i)} \) is a diagonal matrix with diagonal elements being the reciprocal of the number of subdomains with which the degree of freedom is associated. Again it is important to remember that if \( \Omega^{(i)} \) is an interior subdomain (floating subdomain), then \( S^{(i)} \) may be a singular matrix. We must therefore apply generalized inverse techniques for some problems on floating subdomains. To solve the singular problem in this study, \( S^{(i)} \) is replaced by a regularized one;

\[
S^{(i)} = S^{(i)} + \alpha \max\left\{ \text{diag}\left( K_{BB}^{(i)} \right) \right\} I, \tag{6}
\]

where \( \alpha \) is a small positive value, \( \max\left\{ \text{diag}\left( K_{BB}^{(i)} \right) \right\} \) is the maximum absolute value of
diagonal elements of $K_{BB}^{(i)}$ in (2), and $I$ denotes the unit matrix.

5. BDD and BDD-DIAG

The NN preconditioner can be rewritten as

$$M_{NN}^+ = \sum_{i=1}^{N} R_B^{(i)} D^{(i)} S^{(i)+} D^{(i)T} R_B^{(i)T}, \quad (7a)$$

where $S^{(i)+}$ denotes the generalized inverse of $S^{(i)}$. [6] Later, the second author proposed to simplify the NN role in the framework of the BDD preconditioner. That is BDD-DIAG. In his BDD-DIAG, the NN preconditioner is replaced by the following $M_{DIAG}^+$.

$$M_{DIAG}^+ = \sum_{i=1}^{N} R_B^{(i)} \left\{ \text{diag} \left( K_{BB}^{(i)} \right) \right\}^+ R_B^{(i)T}. \quad (7b)$$

Then, for both cases of BDD and BDD-DIAG, we can formally introduce the idea of Mandel [6] as in [3]. Then, a subspace $U$ of $W$ is defined by

$$U = \left\{ u \in W \mid u = \sum_{i=1}^{N} R_B^{(i)} D^{(i)} u^{(i)}, u^{(i)} \in \text{Range } Z^{(i)} \right\}. \quad (8)$$

Here, $Z^{(i)}$ are $\text{dim } W^{(i)} \times m^{(i)}$ matrices of full column rank ($0 \leq m^{(i)} \leq \text{dim } W^{(i)}$) such that

$$\text{Null } S^{(i)} \subset \text{Range } Z^{(i)}, \quad i = 1, \ldots, N. \quad (9)$$

Setting of suitable $Z^{(i)}$ is a crucial point in the Mandel approach, which depends on problems.

Our construction of $Z^{(i)}$ for magnetostatic problems is as follows. [3] Select one nodal point $P_j$ (a midpoint on one side of a tetrahedral element) on which one of the interface DOF is defined for convenience. Construct the row vector of $Z^{(i)}$ corresponding to the nodal point $P_j$, whose DOF are edge components of the Nedelec element, as $\sum_{P_k \in \partial \Omega^{(i)} \setminus \Gamma_B}$, where $\varphi_k$ is a piecewise linear basis function with respect to any vertex $P_k$ in $\Omega^{(i)}$ and on $\partial \Omega^{(i)} \setminus \Gamma_B$ (the summation range should be suitably set, depending on $P_j$). Then, $Z^{(i)} = \sum_{P_j} B_{P_j}^{(i)}$ where $B_{P_j}^{(i)}$ is the row vector which has the following $m^{(i)}(=3)$ components;

$$Z_{P_j}^{(i)} = \left( \sum_{P_k} \frac{\partial}{\partial x_3}(\varphi_k), \sum_{P_k} \frac{\partial}{\partial y_3}(\varphi_k), \sum_{P_k} \frac{\partial}{\partial z_3}(\varphi_k) \right).$$
and $B_{p_j}^{(i)}$ is the dim $W_{(i)} \times 1$ matrix (the $j$-th unit vector) in this case. Note that the above $Z_{(i)}$ construction is a little changed from the original one. [3]

Finally, the Mandel algorithm is summarized as follows.

**Step 1:** Balance the original residual $\mathbf{r}$ by solving the coarse problem for an unknown vector $\lambda \in \mathbb{R}^{\Sigma_{(i)m}}$:

$$S_0 \lambda = R_0^T \mathbf{r},$$

where $R_0 = \left( R_B^{(1)} D^{(1)} Z_{(1)}, \ldots, R_B^{(N)} D^{(N)} Z_{(N)} \right)$, and $S_0 = R_0^T S R_0$.

**Step 2:** Set $s = \mathbf{r} - S R_0 \lambda$, and distribute $s$ to subdomains,

$$s^{(i)} = D^{(i)^T} R_B^{(i)^T} s, \quad i = 1, \ldots, N, \text{ for BDD},$$

$$s^{(i)} = R_B^{(i)^T} s, \quad i = 1, \ldots, N, \text{ for BDD – DIAG}.$$

**Step 3:** Solve Neumann-Neumann problems (or DIAG problems) using (7a) (or (7b)) and average the results.

$$\tilde{\mathbf{u}} = M_{NN}^t s = \sum_{i=1}^{N} R_B^{(i)} D^{(i)^T} \tilde{D}^{(i)^T} R_B^{(i)^T} s = \sum_{i=1}^{N} R_B^{(i)} D^{(i)^T} \mathbf{S}_{(i)} s^{(i)} \quad \text{for BDD},$$

$$\tilde{\mathbf{u}} = M_{DIAG}^t s = \sum_{i=1}^{N} R_B^{(i)} \left( \text{diag} \left( K_{BB}^{(i)} \right) \right)^T s^{(i)} \quad \text{for BDD – DIAG}.$$

**Step 4:** Compute $\bar{s} = \mathbf{r} - S \tilde{\mathbf{u}}$.

**Step 5:** Solve the coarse problem again for an unknown vector $\mu \in \mathbb{R}^{\Sigma_{(i)m}}$:

$$S_0 \mu = R_0^T \bar{s}.$$

**Step 6:** Find the preconditioned vector $\mathbf{z} = \tilde{\mathbf{u}} + R_0 \mu$.

It is well known [6] that the above steps can be simplified and only once the coarse problem (Step 5) may be solved by preparing the suitable initial residual. Our numerical results also use this simplification technique.

### 6. BDD-DIAG and diag

This section explains the asymptotic equivalence between BDD-DIAG and diag after the suitable initial residual is prepared. Namely, the calculation of Step 1 in the previous section using the original initial residual is followed by the modification of the initial value and the initial residual in the PCG procedure. Then, it is sufficient to rewrite Step 3, Step 4 and Step 5 in the previous section.
Step 3:
\[ \overline{u} = M^\dagger_{\text{DIAG}} r = \sum_{i=1}^{N} R_B^{(i)} \left( \text{diag} \left( K_B^{(i)} \right) \right)^\dagger R_B^{(i)T} r. \]

Step 4: Compute \[ \overline{s} = -S\overline{u}. \]

Step 5: Solve the coarse problem for an unknown vector \[ \mu \in \mathbb{R}^{\Sigma l^{(i)}}: \]
\[ S_0 \mu = R_0^T \overline{s}. \]

Step 6: Find the preconditioned vector \[ z = \overline{u} + R_0 \mu. \]

In the following numerical results, the 2-norm of \( R_0 \mu \) is very small. Therefore, iteration counts of BDD-DIAG are almost the same as iteration counts of diag. This numerical evidence derived the following asymptotic equivalence between BDD-DIAG and diag.

**Theorem**

We assume the following two assumptions:

1) Both BDD-DIAG and diag converge.

2) The coarse matrix \( S_0 \) is regular.

Then, \[ R_0 \mu = -R_0 S_0^{-1} R_0^T S M^\dagger_{\text{DIAG}} r, \]
and the 2-norm of \( R_0 \mu \) is estimated by the 2-norm of \( r \). This means that \( R_0 \mu \) asymptotically converges 0 therefore the asymptotic equivalence between BDD-DIAG and diag.

Proof

\[ R_0 \mu = M^\dagger_{\text{BDD-DIAG}} r - M^\dagger_{\text{DIAG}} r = -R_0 S_0^{-1} R_0^T S M^\dagger_{\text{DIAG}} r. \]

Therefore,
\[ \| R_0 \mu \| \leq \| R_0 S_0^{-1} R_0^T \| \| S \| \| M^\dagger_{\text{DIAG}} \| \| r \|. \]

Since we can show the boundedness of three matrix norms, and due to assumption 1), the convergence to 0 of the 2-norm of \( r \) produces the convergence of \( \| R_0 \mu \| \) to 0. Q.E.D.

It is noted that in the above theorem, Assumption 1) is essential but Assumption 2) is for simplicity and may be weakened.

7. **A nonlinear approach and change of the coarse matrix solver**

This section shortly explains how we construct the BDD-DIAG preconditioner in the Newton method of the nonlinear equations. How to treat nonlinear analysis is mentioned in [4]. The difference is only preconditioners. Namely, the simplified diagonal scaling (diag) [4] and BDD-DIAG [3] are compared. First we notice the following:
Nonzero patterns of the Jacobi coefficient matrixes of the Newton method remain similar to those of the coefficient matrix of the linear case.

Making use of this similarity, we repeatedly use the same coarse matrix as the linear case in the Newton iteration. This simple technique works well at least for the present test problems. Other problems will be examined in the next reports.

In this section, change of the coarse matrix solver is also mentioned. To solve the coarse problems in Step 5 of Section 6, parallel skyline solvers are again used as in [11-13]. Though, in magnetostatic analysis, the coarse matrix may become singular and the parallel skyline solver may not be used generally, we can luckily use it in the present test problems. Further consideration is absolutely necessary, which also remains a future problem in this paper.

8. Numerical results

8.1. One part shaft problem and a linear computation

In Reference [3], we showed effectiveness of BDD-DIAG using (7b) instead of (7a). Very interestingly, BDD-DIAG only gives similar results with the diag in both cases of Schur complement matrix construction, though original BDD and NN produce no effective results. The information of $K_{BB}^{(i)}$ is prepared from the diag computation.

Figure 1 shows the cross section of an axi-symmetric shaft model which comes from Fuji Electric Co., Ltd. and is introduced in the textbook [14]. The value $\alpha$ in (6) was suitably set to be $10^{-5}$. [1] An iterative solver (ICCG with a shift value) [1] was used as the subdomain solver on Intel Core i5-4460 with CentOS. Numbers of parts, subdomains, the total DOF and the interface DOF are shown in Table 1. ADVENTURE-Metis whose basis depended on METIS and ParMETIS of Minnesota University was used for domain decomposition. Others are the same as in the previous paper [3]. It is noted that 10,203* and 7,952* for the none cases respectively show the iteration counts where the minimum relative residuals are achieved but the convergence criterions of $10^{-5}$ are still not satisfied.

Two important remarks should be added. In these numerical results, to prepare preconditioners, the global Schur complement matrix was not constructed and the library TryDDM was not used. The singular problem without the perturbation term was 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.
Figure 1: Cross section of an axi-symmetric shaft model (unit: mm)

Table 1: Comparison of convergence situations

|                | Interface DOF | Total DOF  | Subdomains | Parts | Iteration Counts (IC) of diag | IC of BDD-DIAG | IC of none |
|----------------|---------------|------------|------------|-------|--------------------------------|----------------|------------|
|                | 263,966       | 1,251,007  | 1,251      | 1     | 331                            | 329            | 10,203*    |
|                | 115,376       | 569,406    | 569        | 1     | 245                            | 245            | 7,952*     |
|                | 22,787        | 126,345    | 126        | 1     | 151                            | 151            | 3,461      |

8.2. A nonlinear computation for the same problem

Finally, as a preliminary result, we show results of a nonlinear computation for the same model. Figure 2 shows the B-H curve which is used for the nonlinear computation. The third author
checked the residual decreasing of this Newton iteration. For the shaft model, 4 iterations were enough for the Newton iteration convergence. For each iteration of the Newton method, initial values are reset to zeros. Some improvement may be possible for these initial values of each iteration of the Newton method.

Table 2 shows convergence situations of the Newton iteration for the shaft model. Numbers of parts, subdomains, the total DOF and the interface DOF are shown in Table 2. The value $\alpha$ in (6) and others are the same as in the linear computation. Very interestingly, BDD-DIAG by the second author again shows almost the same iteration counts as diag. The original BDD and NN preconditioners were not tested for this nonlinear problem.

Figure 2: The B-H curve


| Table 2: Comparison of convergence situations |
|-----------------------------------------------|
| **Interface DOF**: | 263,966 | 115,376 | 22,787 |
| **Total DOF**: | 1,251,007 | 569,406 | 126,345 |
| **Subdomains**: | 1,251 | 569 | 126 |
| **Parts**: | 1 | 1 | 1 |
| **Iteration Counts (IC) of 0\textsuperscript{th} Step**: | 329 | 245 | 151 |
| **IC of 1\textsuperscript{st} Step**: | 309 | 241 | 144 |
| **IC of 2\textsuperscript{nd} Step**: | 305 | 233 | 138 |
| **IC of 3\textsuperscript{rd} Step**: | 307 | 230 | 140 |
| **IC of 4\textsuperscript{th} Step**: | 318 | 245 | 131 |

9. **Concluding remarks**

In the present research, it is very important for us to reduce number of iterations and computation time. Since computational time is proportional to number of iterations, we have concentrated to reduce number of iterations in this paper. As one possibility, we are trying to implement the NN preconditioner with a coarse problem which has successfully been used in structural analysis [11] where the null space $\text{Null } K^{(i)}$ expresses the rigid body motion which consists of translation and rotation, in thermal analysis [12] where the null space becomes a vector whose components are the same constant and in incompressible viscous flow analysis [13] where the null space may express translation like moving. These analyses allow the similar approach to the $Z^{(i)}$ construction and numerical evidences for effectiveness of BDD preconditioners (iteration counts, computation time, convergence as a function of the number of subdomains and so on) are shown in [11-13].

Recently, also for magnetostatic problems, Tagami [15] gave a similar idea for the $Z^{(i)}$ construction with his successful numerical results. His idea effectively uses macro-elements. Also, the second author showed that his BDD-DIAG using (7b) instead of (7a) was
effective with an easy coarse matrix construction. In the western countries, the BDD preconditioner in this paper is called the Balancing Neumann-Neumann preconditioner (BNN) [16]. From the above situation, for magnetostatic problems, the present approach was expected to be also effective. In fact, as mentioned before, our latest and limited numerical results show that only BDD-DIAG is effective and that BDD and NN are not so. It goes without saying that much more study should be done.

Acknowledgment

The first author would like to thank all support from Fuji Electric Co., Ltd.. Also he gives sincere thanks to his past laboratory members in Kyushu University for their help. Specially, he thanks Mr. Seigo Terada, whose help was absolutely necessary for completeness of this paper. This work was also supported by his past laboratory members in Japan Women’s University with JSPS KAKENHI Grant Numbers 25840142, 24560075 and 15K04762.

References

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

[2] G. Yagawa, R. Shioya: Parallel finite elements on a massively parallel computer with domain decomposition, Computing Systems in Engineering, 4(1993), 495-503.

[3] H. Kanayama, M. Ogino, S. Sugimoto, K. Yodo, H. Zheng: On the coarse matrix solver of preconditioners for magnetostatic domain decomposition analysis, IEEJ Transactions on Power and Energy, 137:3(2017), 179-185.

[4] S. Sugimoto, M. Ogino, H. Kanayama, 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, 304-309.

[5] H. Kanayama, H. Motoyama, K. Endo, F. Kikuchi: Three-dimensional magnetostatic analysis using Nedelec's elements, IEEE Transactions on Magnetics, 26:2(1990), 682-685.

[6] J. Mandel: Balancing domain decomposition, Communications on Numerical Methods in Engineering 9(1993), 233-241.

[7] B. Smith, P. Bjorstad, W. Gropp: Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press, 1996.
A. Quarteroni, A. Valli: Domain Decomposition Methods for Partial Differential Equations, Oxford University Press, 1999.

A. Toselli, O. Widlund: Domain Decomposition Methods -- Algorithms and Theory, Springer, 2004.

V. Dolean, P. Jolivet, F. Nataf: An Introduction to Domain Decomposition Methods -- Algorithms, Theory, and Parallel Implementation, SIAM, 2015.

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

A.M.M. Mukaddes, M. Ogino, H. Kanayama, R. Shioya: A scalable balancing domain decomposition based preconditioner for large scale heat transfer problems, JSME International Journal, (2006), 533-540.

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

H. Kanayama: Computational Electromagnetism, Iwanami, 2000. (in Japanese)

D. Tagami: A balancing domain decomposition method with a multigrid strategy of magnetostatic problems, A presentation in ICCM 2014, Cambridge, UK, 2014.

S. Badia, A.F. Martin, J. Principe: Enhanced balancing Neumann-Neumann preconditioning in computational fluid and solid mechanics, International Journal for Numerical Methods in Engineering, (2012), 1-28.