Performance of Block IC Preconditioner with Fill-in for Linear Systems Derived from Finite Element Meshes Including Thin Elements

Tomonori Tsuburaya[1], Yoshifumi Okamoto[2], and Zhiqi Meng[1]

[1]Department of Electrical Engineering, Fukuoka University, Fukuoka 814-0180, Japan, tsuburaya@fukuoka-u.ac.jp
[2]Department of Electronics and Electrical Engineering, Hosei University, Tokyo 184-8584, Japan, okamotoy@hosei.ac.jp

The incomplete-Cholesky (IC) preconditioner is often implemented into solving linear system derived from finite element method (FEM). However, when many thin elements are included in an analyzed mesh, the convergence characteristic might be deteriorated. At that time, IC preconditioner with fill-in makes the convergence characteristics of linear solver much better. Furthermore, when the blocked strategy is introduced into IC with fill-in, the fill-in search, forward substitution, and backward substitution is successfully parallelized. This paper shows the validity of block IC with fill-in on linear systems arising in FEM with many flat elements.

***Index Terms*—Fill-in strategy, finite element analysis, linear systems, parallel processing.**

I. Introduction

The incomplete-Cholesky-preconditioned conjugate gradient (ICCG) method [1] is widely used for sparse matrix solver in electromagnetic field analysis based on finite element method (FEM). When many thin elements are included in an analyzed mesh, the convergence of ICCG might be deteriorated. Therefore, a powerful preconditioner exceeding conventional IC is desired for fast electromagnetic field analysis.

One of simple techniques to improve the convergence is the consideration of fill-in elements [2] in IC preconditioner. However, the exploring fill-in elements is high cost. Then, this overhead for preconditioner can be reduced by parallelization technique such as block preconditioner [3]. However, the performance of block IC with fill-in has not been particularly investigated on ill-conditioned system derived from FEM with flat elements.

This paper demonstrates the validity of block IC with fill-in for solving symmetric linear systems arising in magnetic field analysis and frequency-domain eddy current problem. Furthermore, the optimal fill-in level was examined through performance evaluation of higher order fill-in.

II. Block Incomplete Factorization Preconditioner

A. Parallelization of Linear Solver on Distributed Memory Parallel computer

In this paper, to efficiently parallelize block IC with fill-in, this preconditioner is parallelized with a distributed memory parallel computer. Fig. 1 shows the outline of parallelization method for the distributed processing. The global matrix information is sent to each process before solving linear equation as shown in Fig. 1 (a). Next, the blocked strategy is applied to local matrix on each process as shown in Fig. 1 (b). The fill-in search, IC factorization, a forward substitution, and a backward substitution can be parallelized block by block.

B. Block IC Preconditioner with Fill-in

The fill-in technique is performed by procedure shown in [2]. First, the fill-in level p, which describes the criterion for determining fill-in components, is previously configured. Next, the initial level q_{ij} in the position (i, j) is defined by

$$q_{ij} = \begin{cases} 0 & (\text{if } a_{ij} \neq 0) \\ \infty & (\text{otherwise}) \end{cases}, \quad (1)$$

where a_{ij} denotes the component of coefficient matrix. Next, all q_{ij} are updated by

$$q_{ij} = \min\{q_{ij},\, q_{ik} + q_{jk} + 1\} \quad (0 < k < j)\,. \quad (2)$$

This modification of q_{ij} can be achieved by the procedure of Gaussian elimination. When the q_{ij} becomes larger than p, q_{ij} is set to ∞; otherwise the (i, j)-th element becomes fill-in entry. Fig. 2 shows the example of fill-in position on block IC. In Fig. 2, p is set to 2. For example, q_{42} is calculated as follows:

$$\begin{aligned} q_{42} &= \min\{q_{42},\, q_{41} + q_{21} + 1\} \quad (0 < k < 2) \\ &= \min\{\infty,\, 0 + 0 + 1\} = 1 \end{aligned} \quad (3)$$

Similarly, q_{43} is obtained as follows:

$$\begin{aligned} q_{43} &= \min\{q_{43},\, q_{42} + q_{32} + 1\} \quad (0 < k < 3) \\ &= \min\{\infty,\, 1 + 0 + 1\} = 2 \end{aligned} \quad (4)$$

Consequently, because the condition $q_{42} \leq 2$ and $q_{43} \leq 2$ are satisfied, these entries become fill-in elements. The other fill-in entries are determined by procedure similar to (3) and (4).

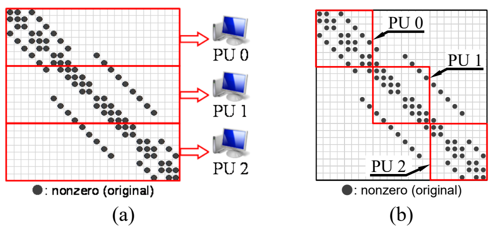


Fig. 1. Outline of parallelization method for distributed processing. (a) Allocation of matrix information to each process. (b) Blocked strategy for preconditioner.

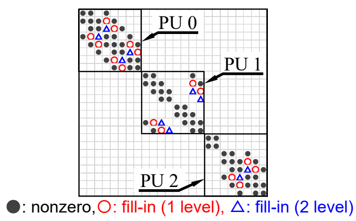


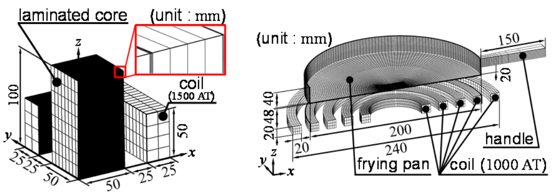
Fig. 2. Example of fill-in position on block IC(2).

III. Numerical Results and Discussion

Fig. 3 shows analyzed models. First, the laminated core model proposed by the institute of electrical engineers of Japan (IEEJ) [4] is analyzed. The direction of lamination is x-axis with 96 % lamination of iron core, and each plate is discretized by 4 layers. Next, Fig. 3 (b) shows the induction heating (IH) model. The frying pan is discretized by 40 layers with thickness of 2 mm, and σ is set to 5.0×10^7 S/m. The frequency of the current in coils is set to 90 kHz. Table I lists the specifications of analyzed models. ε_{CG} is the convergence criterion for CG method. All nonzero components of matrix are stored with the compressed row storage (CRS). The hardware platform is composed of two CPUs, both of which are Intel Xeon E5-2687W v2 (3.4 GHz, 8 cores) with 32 GB RAM. The message passing interface (MPI) is adopted as the API for parallelization.

Fig. 4 shows the nonzero distribution of coefficient matrices. While the nonzero entries are distributed around diagonal in IEEJ laminated core model, the bandwidth of coefficient matrix is large in IH cooker. Table II and III lists the elapsed time for fill-in search. Here, $C^{-1}\boldsymbol{u}$ and $C^{-T}\boldsymbol{u}$ denote the forward substation and backward substitution, respectively. Because the fill-in elements are increased with considering higher order fill-in, the elapsed time for fill-in search, IC factorization, and preconditioning is increased.

Table IV shows the elapsed time for parallelized CG method in IEEJ laminated core model. IC(p) represents the IC preconditioner with fill-level p. N_p is the degree of parallelization. Because some nonzero components are excluded from preconditioner, the convergence of block IC(0) is slower than that of sequential IC(0). On the other hand, the block IC(1) has the ability to improve the convergence in comparison with block IC(0). In block IC(2) and block IC(3), although the number of CG iterations is decreased by consideration of higher order fill-in, total elapsed time could not be reduced. This is caused by the overhead derived from the fill-in strategy. As shown in Table II, when higher level is used as criterion for fill-in position, the cost for determining the place of fill-in is increased. At the same time, because higher fill-in elements are newly added to preconditioner, the increment of elapsed time for forward and backward substitution is occurred. Consequently, the further acceleration could not be realized in block IC(2) and block IC(3).



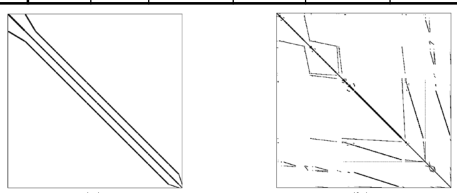
(a) (b)

Fig. 3. Analyzed models. (a) IEEJ laminated core model. (b) IH cooker.

TABLE I

SPECIFICATIONS OF ANALYZED MODELS

model	discret.	formul.	number of elements	matrix type	DoF	nonzero	ε_{CG}
IEEJ laminated core	1st-hexa	$\boldsymbol{A}$	290,241	real	842,384	26,223,568	10^{-6}
IH cooker	1st-hexa	$\boldsymbol{A}-\phi$	794,640	complex	2,742,590	129,062,078	10^{-7}



(a) (b)

Fig. 4. Nonzero distribution. (a) IEEJ laminated core model. (b) IH cooker.

Table V list the performance of parallelized conjugate orthogonal conjugate gradient (COCG) [5]. The convergence of block IC with fill-in is drastically improved in comparison with block IC(0). Consequently, the further speed-up could be achieved. The block IC(1) is the most effective for reducing the elapsed time. Therefore, the optimal p would be 1 from the practical viewpoint. In the full paper, the performance of block IC(1) on large scale problem will be reported.

TABLE II

RESULTANT PERFORMANCE OF BLOCK IC(p) IN IEEJ LAMINATED CORE MODEL (16 PROCESS)

p	number of fill-in	elapsed time for fill-in search [s]	elapsed time for IC [s]	elapsed time for $C^{-1}\boldsymbol{u}$ [s] (1 iteration)	elapsed time for $C^{-T}\boldsymbol{u}$ [s] (1 iteration)
0	–	–	0.01	0.001	0.001
1	14,960,764 (1.00)	16.3 (1.00)	0.04	0.002	0.002
2	23,216,728 (1.55)	20.6 (1.26)	0.08	0.003	0.003
3	35,507,978 (2.37)	26.4 (1.61)	0.10	0.004	0.004

TABLE III

RESULTANT PERFORMANCE OF BLOCK IC(p) IN IH COOKER (16 PROCESS)

p	number of fill-in	elapsed time for fill-in search [s]	elapsed time for IC [s]	elapsed time for $C^{-1}\boldsymbol{u}$ [s] (1 iteration)	elapsed time for $C^{-T}\boldsymbol{u}$ [s] (1 iteration)
0	–	–	0.2	0.020	0.019
1	155,862,560 (1.00)	196.9 (1.00)	1.3	0.059	0.056
2	310,211,102 (1.99)	308.2 (1.56)	3.3	0.099	0.099
3	506,868,558 (3.25)	494.3 (2.51)	7.4	0.140	0.140

TABLE IV

ELAPSED TIME OF PARALLELIZED CG METHOD IN IEEJ LAMINATED CORE MODEL (shift parameter: 1.05)

precond.	N_p	linear it.	elapsed time [s]				
			precond.	$A\boldsymbol{u}$	$C^{-1}\boldsymbol{u}$	$C^{-T}\boldsymbol{u}$	total (T_{Np})
IC(0)	1	15,142	0.3	348.2	191.6	203.1	826.0
Block IC(0)	16	16,037	0.01	86.3	26.1	18.8	242.8
Block IC(1)	16	13,285	16.4	71.0	37.7	31.4	266.8
Block IC(2)	16	12,833	20.7	68.7	45.9	39.9	275.2
Block IC(3)	16	12,451	26.5	66.4	56.5	50.6	292.3

TABLE V

ELAPSED TIME OF PARALLELIZED COCG METHOD IN IH COOKER (shift parameter: 1.1)

precond.	N_p	linear it.	elapsed time [s]				
			precond.	$A\boldsymbol{u}$	$C^{-1}\boldsymbol{u}$	$C^{-T}\boldsymbol{u}$	total (T_{Np})
IC(0)	1	19,767	4.8	6352.6	3171.4	3666.6	14611.8
Block IC(0)	16	15,678	0.2	914.0	323.2	302.2	4157.2
Block IC(1)	16	3,285	198.3	171.4	195.0	185.4	1084.4
Block IC(2)	16	3,283	311.6	164.6	327.2	327.5	1465.7
Block IC(3)	16	3,229	501.8	161.2	452.7	454.2	1902.9

REFERENCES

[1] J. A. Meijerink and H. A. van der Vorst, "An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix," *Math. Comput.*, vol. 31, no. 137, pp. 148-162, Jan. 1977.

[2] Y. Saad, "Iterative Methods for Sparse Linear Systems (2nd ed.)," *SIAM*, Philadelphia, PA, 2003.

[3] P. Concus, G. H. Golub, and G. Meurant, "Block preconditioning for the conjugate gradient method," *SIAM J. Sci. Stat. Comput.*, vol. 6, no. 1, pp. 220-252, Jan. 1985.

[4] Y. Takahashi, S. Wakao, K. Fujiwara, and S. Fujino, "Large-scale magnetic field analysis of laminated core by using the hybrid finite element and boundary element method combined with the fast multipole method," *IEEE Trans. Magn.*, vol. 43, no. 6, Jun. 2007.

[5] H. A. van der Vorst and J. B. M. Melissen, "A Petrov-Galerkin type method for solving $Ax = b$, where A is symmetric complex," *IEEE Trans. Magn.*, vol. 26, no. 2, pp. 706-708, Mar. 1990.