Parallel Processing of 3-D Eddy Current Analysis with Moving Conductor Using Parallelized ICCG Solver with Renumbering Process

Takeshi Iwashita, *Member, IEEE,* and Masaaki Shimasaki, *Member, IEEE*

Abstract—**A new parallelized ICCG scheme is applied to a finite edge element analysis of a 3-d eddy current problem with a moving conductor. In the new method, the global matrix is automatically reordered to the matrix form appropriate to parallel processing. The method requires no particular pre-processing step. It is shown that the newly proposed method has a better parallel efficiency than the Block ICCG method because of a higher preconditioning effect.**

Index Terms—**Eddy current analysis, moving conductor, parallel processing, parallelized ICCG solver.**

I. INTRODUCTION

CURRENTLY, 3-D finite element (FE) analyzes are often carried out for electrical machines with moving conductor parts [1]–[4]. These analyzes require a large number of computations in order to treat the transient behavior of the eddy current in the moving conductor. When a large-scale problem needs to be solved in high resolution, one approach is to utilize a parallel processing technique.

There are several strategies for parallel processing in FE analyzes [5], [6]. One of the most efficient methods is based on the concept of domain decomposition [5]. This method, however, requires a pre-processing step. Moreover, the number of processors is limited in the analysis of a moving conductor since it is preferable to treat one moving part as a single domain. We, therefore, intend to introduce a parallelized solver after assembling the global matrix without any particular pre-processing [6]. In this case, we can easily utilize various techniques [2]–[4] for simulating moving materials developed on a uni-processor machine. The parallelized solver described here can easily be applied to general eddy current analyzes.

The ICCG method is one of the most popular linear solvers for FE analyzes, but the parallel processing with the ICCG method is difficult due to forward–backward substitutions [7].

The present paper proposes a new parallel-processing technique of the ICCG solver, that we call the Parallelized ICCG method with Renumbering Process (PICCG-RP). This new technique modifies the global matrix into a form similar to the dissection ordering case [8], using a renumbering process.

In this study, a parallelized eddy current FE analysis with a moving conductor is carried out on the parallel computer
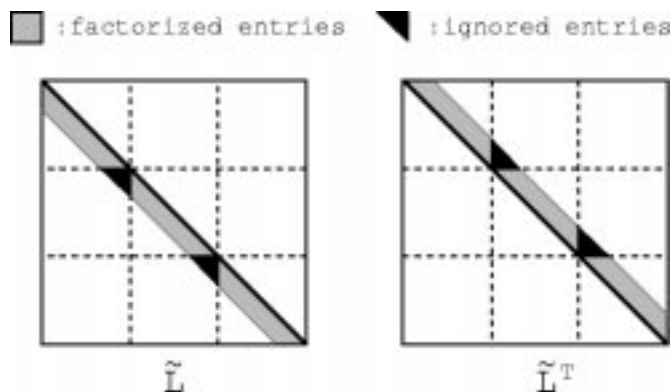
Fig. 1. Block IC factorized matrices (on 3 processors).

HITACHI SR2201. Our results show that the new method can attain a higher preconditioning effect than the Block ICCG method (BICCG) used in [6], [9].

The matrix is first divided into N_p blocks of n_{in} rows, named here as *"interior parts,"* and $N_p - 1$ blocks of n_b rows named as *"interface parts,"* as is shown in Fig. 2(a), where n_{in} is given by the following equation.

$$n_{in} = \{n - (N_p - 1)n_b\}/N_p \tag{1}$$

Next, the matrix is reordered to the form shown in Fig. 2(b), which is similar to a global matrix assembled with a dissection ordering. The vectors are also divided into segments corresponding to the matrix. The ith processor deals with the row blocks and vector segments of the ith interior and interface parts. Though this renumbering process implies a kind of domain decomposition in the analyzed model, users of this solver do not need to take the decomposition into account, unlike the usual domain decomposition method. The renumbering process is performed automatically in the solver, and no pre-processing for the domain decomposition is required.

II. PARALLELIZED ICCG METHOD

In the present paper, we investigate a symmetric positive-definite linear system derived from the FE formulation. It is assumed that the global coefficient matrix of the linear system is a band matrix and that nonzero entries can be located irregularly in the band. The number of processors, the band width, and the dimension of the global matrix are denoted by N_p, n_b, and n, respectively.
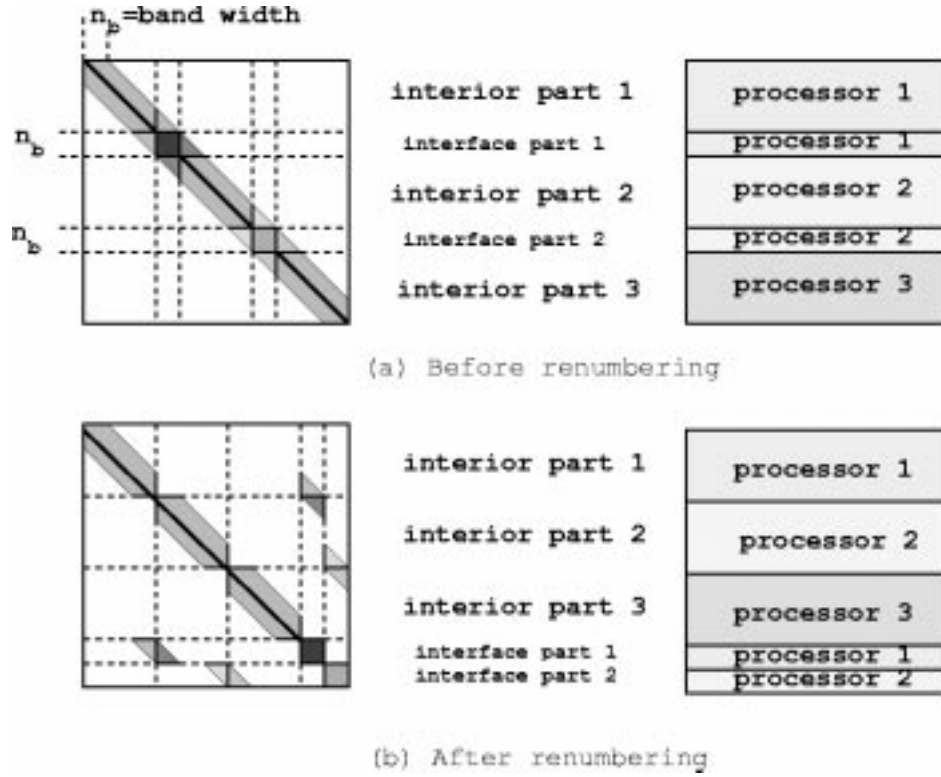
Fig. 2. Renumbering process (on 3 processors).

A. Parallel Processing of the PCG Method

The Preconditioned CG (PCG) method, in particular the ICCG method, is the most popular and effective iterative solver for symmetric positive-definite linear systems. The PCG method involves four main kernels: 1) preconditioning, 2) inner products, 3) matrix-vector products, 4) vector updates. The latter three kernels are easily parallelized with the communications, which depend on the band width, by dividing the vectors (the solution vector, the residual vector, etc.) into N_p segments. On the other hand, it is usually problematic to parallelize the preconditioning kernel, though this difficulty depends on the kind of preconditioner. In the case of the IC preconditioning that is generally used, the preconditioning kernel consists of forward–backward substitutions that are difficult for parallel processing.

In the following subsections, we investigate the parallel processing of the ICCG method, paying special attention to the substitutions.

B. Block ICCG Method

The Block ICCG method (BICCG) is one of the parallel-processing techniques of the ICCG method. This method divides the global matrix into N_p submatrices and performs the IC factorization for the local submatrix in each processor, with the entries between different processors being ignored in the factorization. This block IC factorization results in the preconditioner matrices $\tilde{L}$ and $\tilde{L}^T$, as shown in Fig. 1. forward–backward substitutions with the matrices can be performed in parallel without communications. The BICCG method, however,
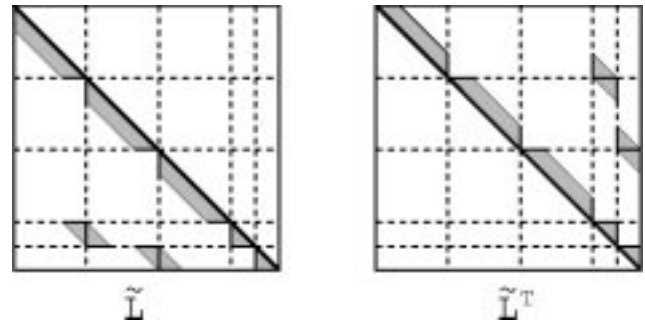

Fig. 3. IC factorized matrices (preconditioner matrices).

suffers from a decline in its preconditioning effect since the ignored entries in the factorization increase in proportion to the number of processors.

C. Parallelized ICCG Method with Renumbering Process

In the present paper, we propose a new technique that we call the Parallelized ICCG method with Renumbering Process (PICCG-RP). In the PICCG-RP, the global matrix is reordered to an appropriate form for parallel processing. Since no entries are neglected in the PICCG-RP, it is expected that this technique will attain a higher preconditioning effect than the BICCG.

Fig. 3 shows the preconditioner matrices $\tilde{L}$ and $\tilde{L}^T$, which are the IC-factorized matrices of the reordered matrix. The forward–backward substitutions with these preconditioner matrices can be performed in parallel as follows (see Fig. 4):

- Step 1: The forward substitution for the interior part is performed in parallel in each processor.
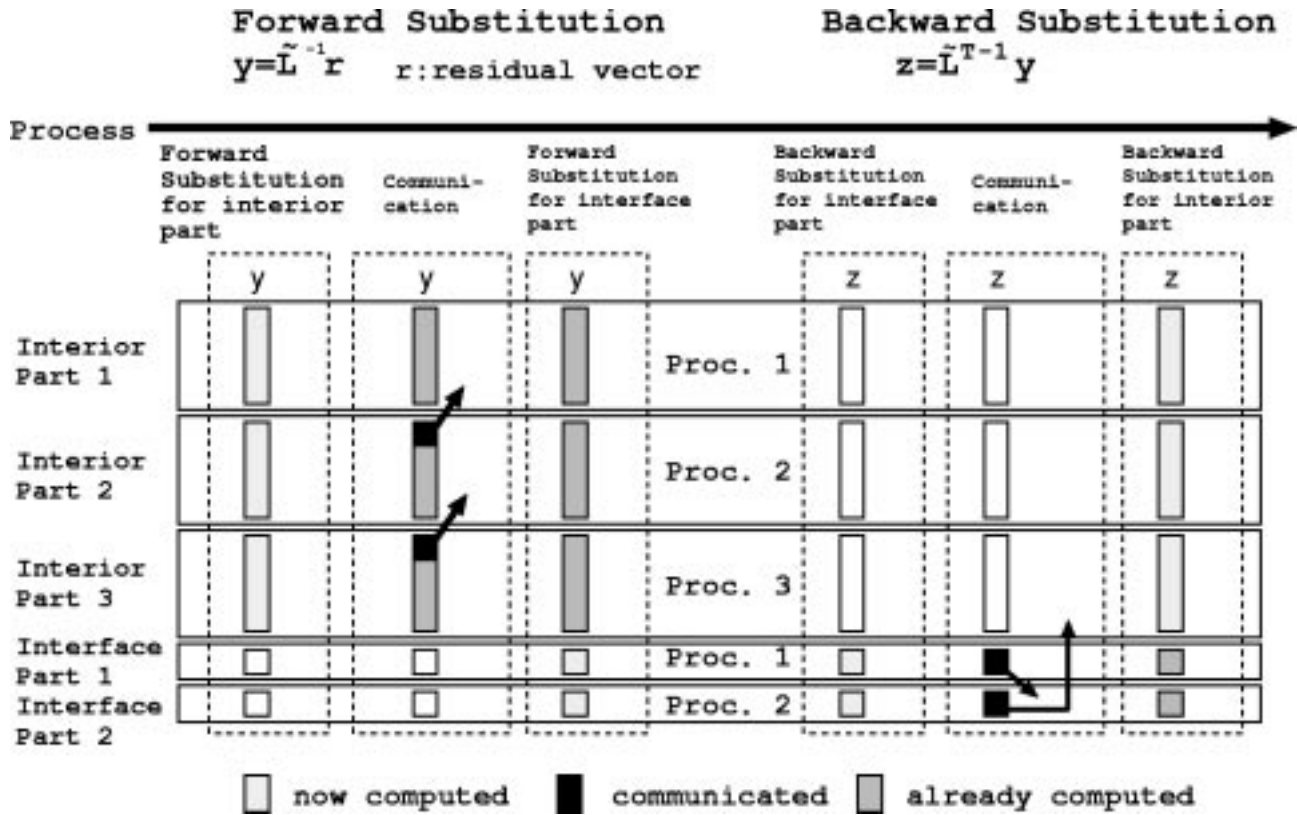
Forward Substitution
$y = \tilde{L}^{-1} r$ r:residual vector

Backward Substitution
$z = \tilde{L}^{T-1} y$

Fig. 4.   Parallelized forward–backward substitutions (on 3 processors).

- Step 2: The n_b upper-side entries of the computed vector segment are communicated from the ith processor to the $(i-1)$th processor.
- Step 3: The forward and backward substitutions for the interface part are carried out in parallel.
- Step 4: The vector segment for the interface part is communicated from the ith processor to the $(i+1)$th processor.
- Step 5: The backward substitution for the interface part is performed in parallel in each processor.

Consequently, one set of forward–backward substitutions can be executed in parallel with the two communications that shift the n_b vector entries to adjacent processors. Moreover, these communications can be overlapped with the computations. If the load balance is considered for the N_pth processor, which is excluded from computations of the interface parts, the rows of the N_pth interior part are increased.

The other kernels (matrix-vector products, update vectors, inner products) are also carried out in parallel.

Fig. 5 shows a flow chart of an analysis using the PICCG-RP. The renumbering cost for the global matrix is about the same as the cost of one matrix-vector product and is negligible in terms of the total computation time. The renumbering costs of the vectors are trivial and also negligible. Moreover, these renumbering processes can be parallelized. While the cost of the IC factorization is amortized over the iterations as with the usual ICCG method, the factorization of the reordered matrix can be carried out in parallel with overlapping n_b rows between adjacent processors.

III. MODEL AND FORMULATION

Fig. 6 shows the analyzed model, which is a simplified model of an eddy current break system. The conductor is assumed to start moving at $t = 0$ with velocity v. The exiting current is DC 10 A.

The model is discretized by first-order brick-type edge elements. When using a moving coordinate system, the basic equation is

$$\nabla \times \left(\frac{1}{\mu} \nabla \times A \right) = -\sigma \frac{\partial A}{\partial t} + J_0 \qquad (2)$$

where A, μ, σ, and J_0 are the magnetic vector potential, the permeability, the electrical conductivity, and the exiting current, respectively [4]. The term for the electric scalar potential is eliminated because of the edge element formulation. By using the backward difference method, the time derivation term is discretized in the moving conductor region as follows:

$$\left(\frac{\partial A(x)}{\partial t} \right)^{t+\Delta t} = \frac{A(x)^{t+\Delta t} - A(x - v\Delta t)^t}{\Delta t} \qquad (3)$$

where
Δt is the time interval,
x is the position on the fixed coordinate system,
v is the velocity of the conductor, and
$A(x)^{t+\Delta t}$ represents the magnetic vector potential at the time $t + \Delta t$ [4].
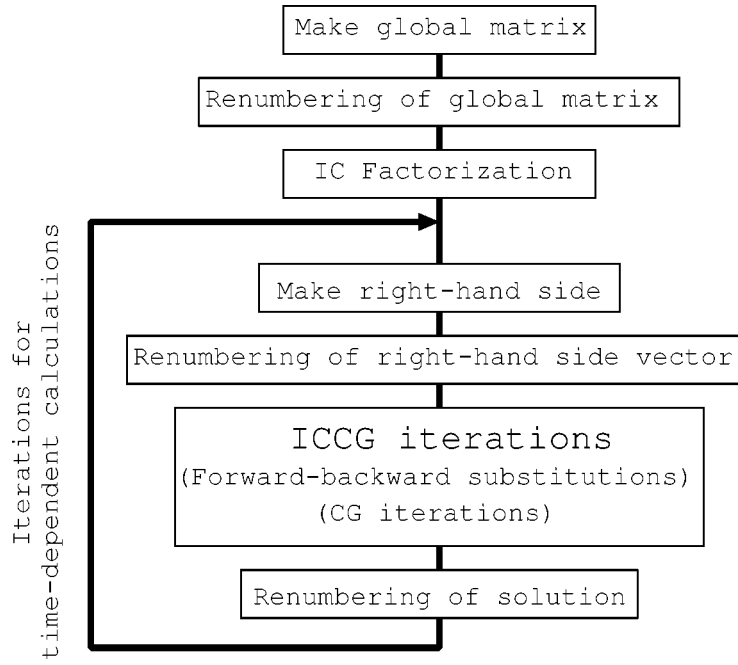
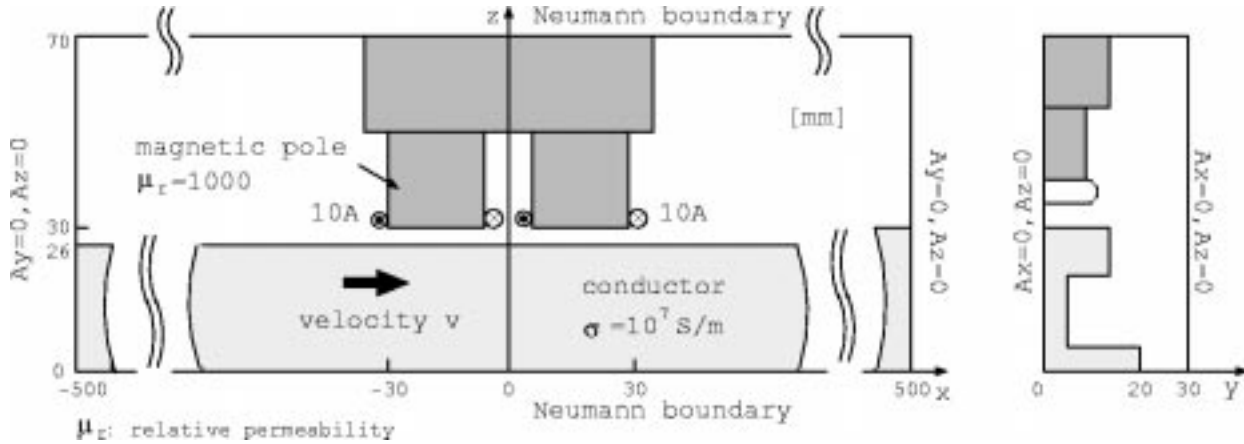Fig. 5.   Flow chart of analysis using PICCG-RP.

Fig. 6.   Analyzed model.

Applying the Galerkin method to (2), we finally obtain the following global linear system.

$$[K + C/\Delta t]\{A_{new}\} = \frac{1}{\Delta t}[C]\{A_{old}\} + \{J\} \qquad (4)$$

$$K_{ij} = \sum_{e}^{m} \iiint_{e} (\nabla \times \boldsymbol{N}_i) \cdot \left(\frac{1}{\mu}\nabla \times \boldsymbol{N}_j\right) dV \qquad (5)$$

$$C_{ij} = \sum_{e}^{m} \iiint_{e} \sigma \boldsymbol{N}_i \cdot \boldsymbol{N}_j \, dV \qquad (6)$$

$$J_i = \sum_{e}^{m} \iiint_{e} \boldsymbol{N}_i \cdot \boldsymbol{J}_0 \, dV \qquad (7)$$

where
m is the number of elements,
e is the element,

$\{A_{new}\}$ is the unknown vector of the potential $\boldsymbol{A}(x)$ at $t = t + \Delta t$, and
$\{A_{old}\}$ represents the previous potential $\boldsymbol{A}(x - v\Delta t)^t$.
While the global coefficient matrix $[K + C/\Delta t]$ is constant, the right-hand side of (4) changes in the conductor region at each time step. The effect of the eddy current induced by the movement of the conductor is taken into account in computing the right-hand side of the equation.

IV. NUMERICAL RESULTS

Parallelized eddy current analyzes are carried out on the distributed memory parallel computer HITACHI SR2201. Table I lists the analysis conditions and the discretization data. The original global matrix is assembled with a natural ordering. The program code is written in FORTRAN language with the MPI library. The convergence criterion of the ICCG method is given by $\|\boldsymbol{r}\|_2/\|\boldsymbol{b}\|_2 < 10^{-8}$, where $\boldsymbol{b}$ and $\boldsymbol{r}$ are the right-hand side vector and the residual vector, respectively.

TABLE I
ANALYSIS CONDITIONS AND DISCRETIZATION DATA

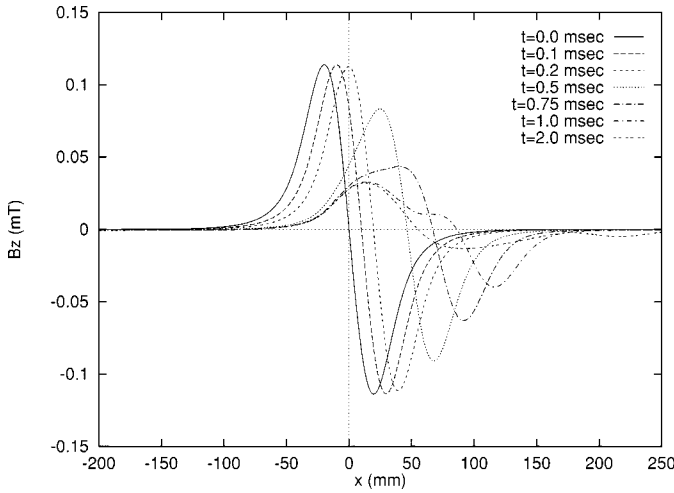|	
number of elements	60000
number of nodes	77077
number of unknowns	213136
velocity of conductor	100 m/s
time step	0.01 msec
analysis time	0 ∼ 200 msec

Fig. 7.   Time-dependent variations of the magnetic flux density.

TABLE II
COMPUTATION TIME

N_p	DPCG (h)	BICCG (h)	PICCG-RP (h)
1	135	30.7	30.7
2	67.8	17.9	16.4
4	33.4	10.0	9.55
8	16.8	5.80	5.49
16	8.75	3.73	3.18
22	6.40	3.10	2.60

TABLE III
AVERAGE NUMBER OF CG ITERATION IN ONE TIME STEP

N_p	DPCG	BICCG	PICCG-RP
1	1746	175	175
2	1746	200	191
4	1746	226	222
8	1746	262	255
16	1746	337	298
22	1746	376	324

Fig. 7 shows the time-dependent variation of the z-component of the magnetic flux density at $y = 0$, $z = 2$ mm. The eddy current induced by the movement of the conductor gradually weakens the magnetic flux density.

Tables II and III list the total computation time and the average number of CG iterations in one time step, respectively. For comparison, the Diagonal Preconditioned CG method (DP)
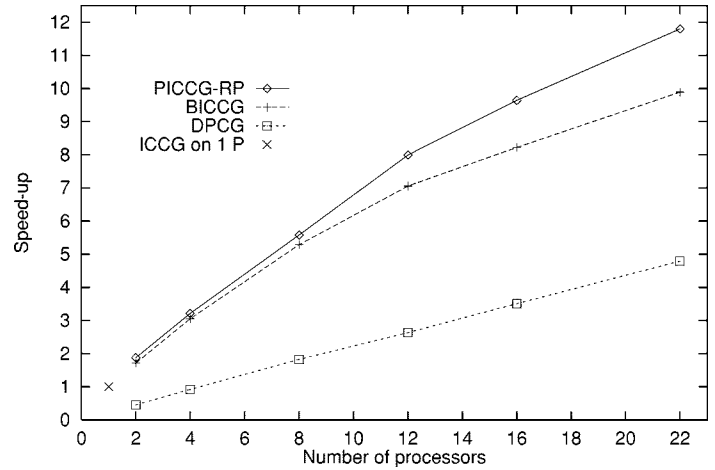

Fig. 8.   Speed-up.

is also examined. Fig. 8 depicts the speed-up ratio of the parallelized solvers compared with the ICCG method implemented on one processor. Fig. 8 shows that the PICCG-RP can attain a better parallel efficiency than the other two solvers. This is due to the better convergence rate of the PICCG-RP. While the DP achieves an ideal linear speed-up, its preconditioning effect is smaller than the IC preconditioning. The present analysis shows that the kernels of the CG method (for example, forward–backward substitutions, matrix-vector products, etc.) are ideally parallelized in both the BICCG and the PICCG-RP. This result implies that the costs for the renumbering process and the communication are negligible in the PICCG-RP. Table III, however, shows that the BICCG and the PICCG-RP suffer from a trade-off between parallelism and the convergence rate. In the BICCG, this effect is due to the global matrix entries being ignored in the Block IC factorization. In the PICCG-RP, the trade-off is caused by the renumbering process. In general, the numbering strategy affects the IC preconditioning, and the natural ordering is one of the best ordering strategies [8]. In the present analysis, the renumbering process in the PICCG-RP destroys the data dependency relationship of the coefficients in a global matrix with natural ordering, which results in a trade-off. But, the trade-off is smaller in the PICCG-RP than in the BICCG, since no matrix entry is neglected in the PICCG-RP. In particular, as the number of processors increases, the advantages of PICCG-RP over BICCG becomes more significant.

V. CONCLUSION

A new parallel-processing technique of the ICCG solver, the Parallelized ICCG Method with Renumbering Process, was proposed in the present paper. A parallelized eddy current analysis with a moving conductor was performed by utilizing parallelized ICCG solvers. The following is a summary of the results.

- Introducing a parallelized linear solver without pre-processing is an effective and easy way of parallel processing in FE analyzes.
- A 12-fold speed-up is achieved by 22 processors working on a problem with approximately 220 000 degrees of freedom.

• The method proposed here, the PICCG-RP, attains a better parallel efficiency than the BICCG because of the higher preconditioning effect.

REFERENCES

[1] M. Fujita, T. Tokumasu, T. Yamada, T. Hirose, Y. Tanaka, N. Kumagai, and S. Uchida, "3-dimensional electromagnetic analysis and design of an eddy-current rail brake system," *IEEE Trans. Magn.*, vol. 34, no. 5, pp. 3548–3551, September 1998.

[2] K. Tani, T. Nishio, T. Yamada, and Y. Kawase, "Transient finite element method using edge elements for moving conductor," *IEEE Trans. Magn.*, vol. 35, no. 3, pp. 1384–1386, May 1999.

[3] K. Tani, T. Yamada, and Y. Kawase, "Dynamic analysis of linear actuator taking into account eddy currents using finite element method and 3-d mesh coupling method," *IEEE Trans. Magn.*, vol. 35, no. 3, pp. 1785–1788, May 1999.

[4] K. Muramatsu, T. Nakata, N. Takahashi, and K. Fujiwara, "Comparison of coordinate systems for eddy current analysis in moving conductors," *IEEE Trans.on Magn.*, vol. 28, no. 2, pp. 1186–1189, March 1992.

[5] K. Iwano, V. Cingoski, K. Kaneda, and H. Yamashita, "A parallel processing method in finite element analysis using domain division," *IEEE Trans. Magn.*, vol. 30, no. 5, pp. 3598–3601, September 1994.

[6] C. Vollaire and L. Nicolas, "Preconditioning techniques for the conjugate gradient solver on a parallel distributed memory computer," *IEEE Trans. Magn.*, vol. 34, pp. 3347–3350, September 1998.

[7] R. Barrett *et al.*, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*: SIAM, 1994.

[8] I. S. Duff and G. A. Meurant, "The effect of ordering on preconditioned conjugate gradients," *BIT*, vol. 29, pp. 635–657, 1989.

[9] M. L. Barton, "Three-dimensional magnetic field computation on a distributed memory parallel processor," *IEEE Trans. Magn.*, vol. 26, no. 2, pp. 834–836, March 1990.

[10] S. Doi, "On parallelism and convergence of incomplete LU factorizations," *Appl. Numer. Math.*, vol. 7, pp. 417–436, 1991.