

Algebraic Multicolor Ordering for Parallelized ICCG Solver in Finite-Element Analyses

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

Abstract—This paper proposes a new black-box-type parallel processing method for the incomplete Cholesky conjugate gradient (ICCG) solver. The new method is based on a multicolor ordering concept and an automatic reordering process in the solver. Parallel performance is evaluated in the context of three-dimensional finite edge-element eddy-current analysis. The proposed method attains high parallelism with a small increase in CG iterations and achieves high parallel performance.

Index Terms—Algebraic multicolor ordering, black-box-type solver, ICCG method, parallel processing.

I. INTRODUCTION

THE INCOMPLETE Cholesky conjugate gradient (ICCG) method [1] is the most popular solver for symmetric positive-definite linear systems arising in finite-element (FE) analyses and finite-difference (FD) analyses. Parallel processing of the ICCG method is, however, difficult due to the forward-backward substitutions performed for preconditioning. Several strategies have been presented for the parallelization of the ICCG method [2], [3].

A reordering technique is one of the most effective methods for parallel processing the ICCG solver [2], [3]. The technique, however, entails a tradeoff between convergence and parallelism [4]. In order to overcome the tradeoff, Doi and Lichniewsky proposed the use of many colors, about 30 or more, in multicolor ordering [5], [6] (*color* means one subset of nodes or unknowns independent of each other). Their technique, which is called *large-numbered multicolor ordering*, attains high parallelism with a small increase of CG iterations. Since their technique is developed for structured FD analyses and cannot be directly applied to unstructured FE analyses, we have investigated new strategies for utilizing multicolor ordering in the context of FE analyses. In our previous work [7], we proposed a new technique using virtual nodes and succeeded in exploiting the advantages of large-numbered multicolor ordering in FE analyses. This technique, however, requires a preprocessing step using information on geometric relationships between the elements. From the view point of portability and flexibility, black-box-type solvers are preferable.

This paper proposes a new black-box-type parallelized ICCG method, which we call the *algebraic multicolor ordering* method. While the new method is also based on the concept of

multicolor ordering, a reordering process for the unknowns is automatically performed in the solver. The key of the method is algebraic coloring of the unknowns. Several colors are assigned to the unknowns, where the unknowns belonging to the same color must not have data dependency on each other. In this paper, we propose a new coloring strategy without using some techniques presented in research regarding graph theory [8], [9]. This is because these techniques, which have been developed for computational fluid dynamics based on finite nodal-element analyses, are not efficiently applicable to electromagnetic field analyses with edge elements and are not completely algebraic. On the other hand, our new method is a complete black-box solver, and its implementation is easy. Consequently, the new solver can be used in various analyses including both FE and FD analyses.

A parallelized finite edge-element eddy-current analysis is performed for evaluation of the proposed method. It is shown that the new method attains a better performance with small increase of CG iterations than the Block ICCG method which is a typical black-box-type parallelized ICCG solver.

II. ALGEBRAIC MULTICOLOR ORDERING METHOD

The ICCG method consists of four main kernels: 1) forward and backward substitutions; 2) inner products; 3) matrix-vector products; and 4) vector updates. The latter three kernels can be easily parallelized by dividing the vectors into several segments. On the other hand, the forward-backward substitution cannot be easily parallelized. This paper focuses mainly on the parallel processing method of forward and backward substitutions based on algebraic multicolor ordering.

A. Notation and Overview of Algebraic Multicolor Ordering

In this paper, we solve a linear system of equations $\mathbf{Ax} = \mathbf{b}$ derived from a FE formulation. The coefficient matrix \mathbf{A} is assumed to be symmetric and positive or semi-positive definite. The vectors \mathbf{x} and \mathbf{b} are the solution vector and the right-hand-side vector, respectively. The dimension of the matrix is given by n . The following notations are here used. The matrix \mathbf{L} represents the strictly lower triangular part of \mathbf{A} . We denote the column numbers of nonzero entries in the i th row of the matrix \mathbf{L} by $\text{lnzc}(\mathbf{i}, \mathbf{j})$ ($\mathbf{j} = 1, \dots, \text{lnz}(\mathbf{i})$), where $\text{lnz}(\mathbf{i})$ is the number of nonzero entries in the i th row.

In the multicolor ordering method, the unknowns are divided into several subsets in each of which the unknowns have no direct relationship. Here, the no direct relationship between the p_i th unknown and the p_j th unknown means that the (p_i, p_j) entry of the coefficient matrix \mathbf{A} is zero. The forward-backward

Manuscript received July 5, 2001; revised October 25, 2001.

T. Iwashita is with the Data Processing Center, Kyoto University, Kyoto 606-8501, Japan (e-mail: take@kudpc.kyoto-u.ac.jp).

M. Shimasaki is with the Department of Electrical Engineering, Kyoto University, Kyoto 606-8501, Japan (e-mail: simasaki@kuee.kyoto-u.ac.jp).

Publisher Item Identifier S 0018-9464(02)02350-6.

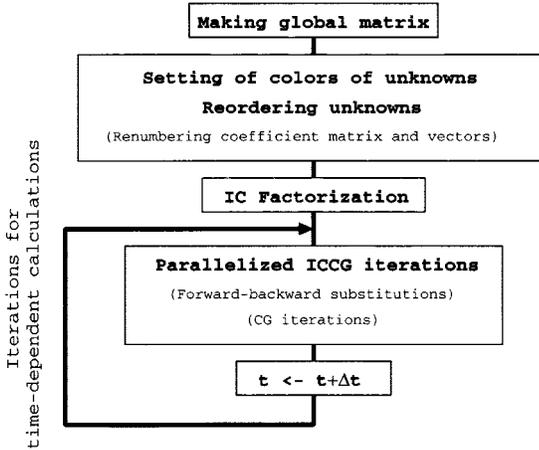


Fig. 1. Flow chart of analysis using algebraic multicolor ordering.

substitution is parallelized in each subset. In the multicolor ordering method, one subset is regarded as one color. The division of the unknowns is, therefore, given by the assignment of colors to the unknowns. In the *algebraic* multicolor ordering method, the assignment is carried out by using only information obtained from a coefficient matrix. After the colors of all the unknowns are determined, the unknowns are reordered according to the order of the colors. A coefficient matrix, a right-hand side vector and a solution vector are also renumbered corresponding to the new order of the unknowns. The ICCG kernels are parallelized in accordance with the parallelism of the ordering. Fig. 1 illustrates a flow chart of an analysis using the algebraic multicolor ordering method. The details of the method are described in the following subsections.

B. Assignment of Colors to the Unknowns and Renumbering Process

The first step in the method is setting the colors of the unknowns. The algebraic multicolor ordering method automatically sets the color so as to satisfy the following condition for parallel processing.

Condition(MC: Multicolor)—The unknowns with the same color do not have direct relationships with each other.

In our technique, one special array $\mathbf{color}(i)$ ($i = 1, \dots, n$) is allocated. The array shows the color of the i th unknown. Fig. 2 shows the procedure of setting the array \mathbf{color} in Fortran language. (Note that the array \mathbf{lnz} and \mathbf{lnzc} are introduced for the sake of explanation and are not necessary in the actual program code.) The number of colors n_{color} is set first. The array \mathbf{color} is initially set at zero. The colors from one to n_{color} are assigned to the unknowns basically in a cycle. The color of the i th unknown is determined as follows:

Let $icolor$ be the current color (i.e., the color of the $(i - 1)$ th unknown is $icolor - 1$).

- *Step 1:* Check the colors of the $\mathbf{lnzc}(i, j)$ th unknowns ($j = 1, \dots, \mathbf{lnz}(i)$). If $icolor$ is not assigned to these unknowns, set $icolor$ as the color of the i th unknown and move to the next $(i + 1)$ th unknown. Otherwise, move to Step 2.
- *Step 2:* Update $icolor$ to $\text{mod}(icolor, n_{color}) + 1$, and return to Step 1.

```

!-----
ncolor=some value      ! Set number of used colors
color(1:n)=0          ! Initialize the array color
icolor=1
do i=1,n
  j=1
  do while(j <= lnz(i))
    if (color(lnzc(i,j))==icolor) then
      icolor=mod(icolor+1,ncolor)+1 !To next color
      j=0
    endif
    j=j+1
  enddo
  color(i)=icolor      ! Assignment of color
  icolor=mod(icolor+1,ncolor)+1 !To next color
enddo
!-----

```

Fig. 2. Fortran program code of setting the array color.

The procedure above is sequentially applied to all the unknowns. For maintaining the condition(MC), the number of colors n_{color} has a lower boundary, which is given by the maximum value of $\mathbf{lnz}(i) + 1$ ($i = 1, \dots, n$). When n_{color} satisfies this condition, Step 1 above guarantees the condition(MC) which is required for parallel processing. The computational cost for setting the color is trivial compared with the ICCG iterations.

After the setup of the colors of the unknowns, the unknowns are reordered following the order of color. The coefficient matrix and the vectors are also renumbered. The renumbering cost is usually negligible in terms of the total computation time [10].

C. Parallelized ICCG Method Based on Algebraic Multicolor Ordering

Here, we describes the procedure of parallelized forward and backward substitutions using the algebraic multicolor ordering method.

Let $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ be the renumbered linear system obtained from Section II-B. Corresponding to the colors of the unknowns, the solution vector $\tilde{\mathbf{x}}$ and the coefficient matrix $\tilde{\mathbf{A}}$ are split as

$$\tilde{\mathbf{x}} = (\tilde{\mathbf{x}}_{C(1)}, \tilde{\mathbf{x}}_{C(2)}, \dots, \tilde{\mathbf{x}}_{C(ncolor)})^T \quad (1)$$

$$\tilde{\mathbf{A}} = \begin{pmatrix} \tilde{\mathbf{C}}_{1,1} & \tilde{\mathbf{C}}_{1,2} & \cdots & \tilde{\mathbf{C}}_{1,ncolor} \\ \tilde{\mathbf{C}}_{2,1} & \tilde{\mathbf{C}}_{2,2} & \cdots & \tilde{\mathbf{C}}_{2,ncolor} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{C}}_{ncolor,1} & \tilde{\mathbf{C}}_{ncolor,2} & \cdots & \tilde{\mathbf{C}}_{ncolor,ncolor} \end{pmatrix} \quad (2)$$

where $\tilde{\mathbf{x}}_{C(ic)}$ corresponds to the unknowns with ic th color, and $\tilde{\mathbf{C}}_{i,m}$ represents the relationship between $\tilde{\mathbf{x}}_{C(i)}$ and $\tilde{\mathbf{x}}_{C(m)}$. When the colors of the unknowns are determined by the procedure described in the preceding subsection, $\tilde{\mathbf{C}}_{ic,ic}$ ($ic = 1, \dots, n_{color}$) is given by diagonal matrices (see Fig. 3). Accordingly, the incomplete factorized matrices of $\tilde{\mathbf{A}}$ for preconditioning are given by

$$\tilde{\mathbf{L}} = \begin{pmatrix} \tilde{\mathbf{D}}_1 & & & \\ \tilde{\mathbf{L}}_{2,1} & \tilde{\mathbf{D}}_2 & & \mathbf{O} \\ \vdots & \ddots & \ddots & \\ \tilde{\mathbf{L}}_{ncolor,1} & \tilde{\mathbf{L}}_{ncolor,2} & \cdots & \tilde{\mathbf{D}}_{ncolor} \end{pmatrix} \quad (3)$$

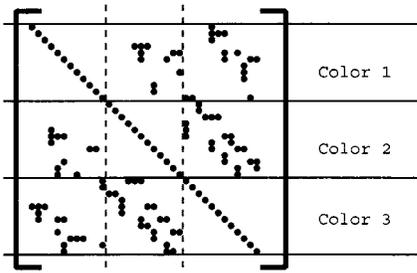


Fig. 3. Example of a coefficient matrix renumbered using algebraic multicolor ordering (three colors).

$$\tilde{D} = \begin{pmatrix} \tilde{D}_1^{-1} & & O \\ & \ddots & \\ O & & \tilde{D}_{ncolor}^{-1} \end{pmatrix} \quad (4)$$

where

$$\tilde{A} \approx \tilde{L}\tilde{D}\tilde{L}^T \quad (5)$$

and \tilde{D}_{ic} ($ic = 1, \dots, ncolor$) are diagonal matrices.

The forward substitution in the ICCG iteration is given by

$$\tilde{L}\tilde{y} = \tilde{r} \quad (6)$$

where \tilde{r} is the residual vector. From (3), the substitution corresponding to the ic th color in (6) is given by

$$\tilde{y}_{C(ic)} = \tilde{D}_{ic}^{-1} \left(\tilde{r}_{C(ic)} - \sum_{k=1}^{ic-1} \tilde{L}_{ic,k} \tilde{r}_{C(k)} \right) \quad (7)$$

where $\tilde{y}_{C(k)}$ and $\tilde{r}_{C(k)}$ are segments of \tilde{y} and \tilde{r} corresponding to the k th color. When $\tilde{r}_{C(k)}$ ($k = 1, \dots, ic-1$) are already computed and are communicated among all the processors, the forward substitution (7) consists of matrix-vector products and is easily parallelized. Accordingly, the forward substitution can be parallelized in each color. After the substitution corresponding to the ic th color (7) is performed in parallel, the obtained values of $\tilde{y}_{C(ic)}$ are communicated among all the processors. The backward substitution is also parallelized in the same way. Fig. 4 shows the procedure of the parallelized substitutions based on the algebraic multicolor ordering method. In one substitution, $(ncolor - 1)$ -times communications are needed. The degree of parallelism obtained from the present method is given by the number of unknowns in one color, which is about $n/ncolor$.

III. RESULTS

A. Test Model and Computation Environment

In the present analysis, we use the IEEJ standard benchmark model of three-dimensional eddy-current analyses [11] for evaluation of the proposed method. Fig. 5 shows the analyzed model, which is discretized by first-order brick-type edge elements. Table I lists the discretization data. The electromagnetic field equations are solved by the Galerkin method with A formulation and the backward time-difference method. The original linear system is assembled with natural ordering.

The present parallelized analysis was implemented on the distributed-memory parallel computer Fujitsu VPP-800 at the

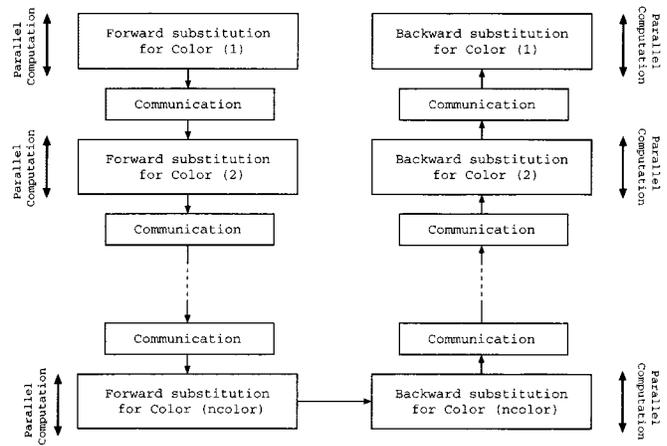


Fig. 4. Parallel processing of forward and backward substitutions with algebraic multicolor ordering method.

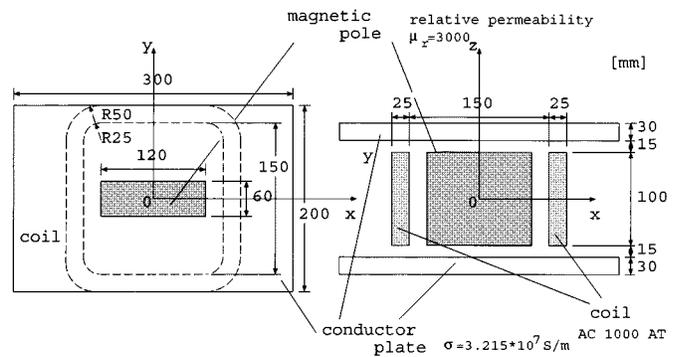


Fig. 5. IEEJ standard benchmark model (unit: [mm]).

TABLE I
DISCRETIZATION DATA

Number of volume elements	327680
Number of nodes	342225
Number of unknowns	1011920
Time step	1 msec

Data Processing Center, Kyoto University. The parallelized program code is written by using FORTRAN and MPI. The convergence criterion of the ICCG method is given by $\|\tilde{r}\|_2/\|\tilde{b}\|_2 = \|\tilde{r}\|_2/\|\tilde{b}\|_2 < 10^{-7}$, where \tilde{r} is the residual vector of the original linear system. The acceleration factor of the ICCG method is chosen to be 1.03. The parallel performance of the proposed method is evaluated in the first one time step.

B. Parallel Performance

Table II and Fig. 6 show the computation time, the number of iterations, and the convergence behavior of the proposed method compared with the Block-ICCG method where N_p is the number of processors.

In the Block-ICCG method, some sets of entries in the coefficient matrix are ignored for parallel processing in the incomplete factorization, which results in an increase of iterations due to a decline in the preconditioning [10]. Since the ignored entries increase in proportion to the number of processors, many CG iterations are generally needed for convergence when many processors are used.

TABLE II
COMPARISON OF COMPUTATION TIME AND NUMBER OF ITERATIONS

N_p	Computation time (sec)	# of iterations
1	2119	366
N_p	Computation time (sec)	# of iterations
2	1394	530
4	1028	779
8	550.5	834
16	251.6	675
32	120.4	726
N_p	Computation time (sec)	# of iterations
2	1194.4	390
4	606.3	390
8	308.4	390
16	164.3	390
32	99.83	390

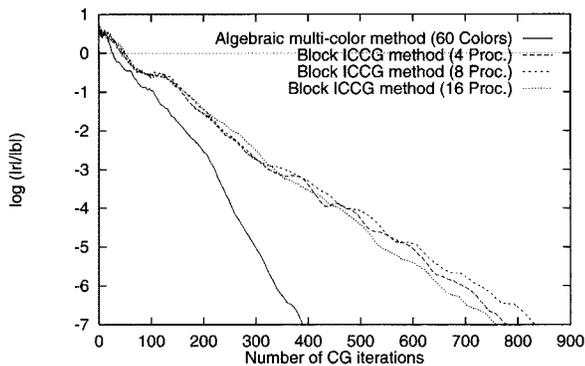


Fig. 6. Comparison of convergence behavior.

TABLE III
DEGREE OF PARALLELISM AND NUMBER OF ITERATIONS IN ALGEBRAIC MULTI-COLOR ORDERING METHOD

Number of colors	Degree of parallelism (approximate value)	Number of iterations
50	20200	426
60	16800	390
70	14500	393
100	10200	440
500	2000	383

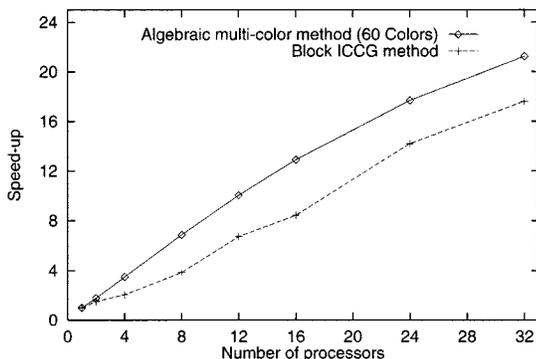


Fig. 7. Speed-up ratio of solvers compared with the ICCG on 1P.

On the other hand, in the multicolor ordering method, the convergence rate of the ICCG solver depends not on the number of processors but on the number of colors. Table III lists the degree of parallelism and number of iterations in the algebraic multicolor ordering method using various colors, showing that the proposed method achieves a high degree of parallelism with a slight increase of iterations compared with the Block-ICCG

method. While the use of many colors improves the convergence rate in general, it causes an increase in communication costs and a decline in parallelism. Accordingly, the number of colors has to be determined with consideration of this tradeoff. In previous research regarding FD analyses for a wide variety of applications, a use of from 30 to 100 colors is recommended [6], [9]. In the present analysis using edge elements, the lower boundary value of the number of colors is 33. While we have tested various numbers of colors from 33 to 100, the best solver performance is attained in the case of 60 colors. The proposed method with 60 colors attains approximately 16800 ($=1011920/60$) degrees of parallelism with only a 10% increase in CG iterations. Consequently, the proposed method can attain higher parallel performance than the Block ICCG method as shown in Fig. 7. The saturation of the speed-up is due to an increase in the rate of communication cost to computation cost. Better parallel performance is expected in a larger-scale analysis. The present result also confirms that the recommended number of colors based on FD analyses is effective in FE analyses as well. The generalized procedure to attain an optimal number of colors is being developed and will be reported in future.

IV. CONCLUSION

An algebraic multicolor ordering method is proposed for a new black-box-type parallelized ICCG solver. A set-up procedure for multicolor ordering and a renumbering process are automatically performed in the solver. The new method attains high parallelism with a small increase of CG iterations and achieves over a 20-fold speed-up by 32 processors on an eddy-current analysis with about 10^6 degrees of freedom.

REFERENCES

- [1] J. 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, pp. 148–162, 1977.
- [2] H. A. van der Vorst and T. F. Chan, "Parallel preconditioning for sparse linear equations," *ZAMM. Z. Angew. Math. Mech.*, vol. 76, pp. 167–170, 1996.
- [3] I. S. Duff and H. A. van der Vorst, "Developments and trend in the parallel solution of linear systems," *Parallel Computing*, vol. 25, pp. 1931–1970, 1999.
- [4] I. S. Duff and G. A. Meurant, "The effect of ordering on preconditioned conjugate gradients," *BIT*, vol. 29, pp. 635–657, 1989.
- [5] S. Doi and A. Lichniewsky, "A graph-theory approach for analyzing the effects of ordering on ILU preconditioning," INRIA, France, Tech. Rep. 1452, 1991.
- [6] S. Doi and T. Washio, "Ordering strategies and related techniques to overcome the tradeoff between parallelism and convergence in incomplete factorization," *Parallel Computing*, vol. 25, pp. 1995–2014, 1999.
- [7] T. Iwashita and M. Shimasaki, "Construction and ordering of edge elements for parallel computation," *IEEE Trans. Magn.*, vol. 37, pp. 3498–3502, Sept. 2001.
- [8] M. T. Jones and P. E. Plassmann, "The efficient parallel iterative solution of large sparse linear systems," *Graph Theory and Sparse Matrix Computations, the IMA Volumes in Mathematics and its Applications*, vol. 56, pp. 229–245, 1994.
- [9] T. Osoda, K. Maruyama, T. Washio, S. Doi, and S. Yamada, "Vectorization and parallelization technique of block ILU preconditioning for unstructural problems" (in Japanese), *IPJSJ Trans. HPS*, vol. 41, pp. 92–99, 2000.
- [10] T. Iwashita and M. Shimasaki, "Parallel processing of 3-D eddy-current analysis with moving conductor using parallelized ICCG solver with renumbering process," *IEEE Trans. Magn.*, vol. 36, pp. 1504–1509, July 2000.
- [11] T. Nakata, N. Takahashi, T. Imai, and K. Muramatsu, "Comparison of various methods of analysis and finite elements in 3-D magnetic field analysis," *IEEE Trans. Magn.*, vol. 27, pp. 4073–4076, Sept. 1991.