An Effective Method to Reduce the Computing Time of Nonlinear Time-Stepping Finite-Element Magnetic Field Computation

W. N. Fu, S. L. Ho, H. L. Li, and H. C. Wong

***Abstract*—Time-stepping finite-element methods have been widely used to compute the magnetic field of electrical machines. Because the reluctivities of magnetic materials are nonlinear, the finite-element equations have to be solved iteratively. In this paper, an effective method for reducing the computing time of the Newton–Raphson method coupled with the incomplete Cholesky-conjugate gradient algorithm for solving time-stepping finite-element problems is presented. The proposed method is based on a proper prediction of some predefined error tolerances in the iteration processes at each time-stepping finite-element computation. The computational analysis on an induction motor shows that the proposed strategy can reduce the nominal computing time by as much as 50%.**

***Index Terms*—Computation time, finite element, ICCG method, Newton–Raphson method, nonlinear, time stepping.**

I. Introduction

THE circuit-field-motion coupled time-stepping finite-element method (FEM) is a very effective tool in the analysis of electrical machines. The method not only can include many complicated factors such as the saturation of iron materials, eddy current, and skewed rotor bars but can also produce transient solutions. Compared with the traditional FEM methods, the time-stepping FEM is more demanding on the computing time. Hence, one must refine the time-stepping FEM algorithms in order to reduce the computing time.

For a nonlinear FEM problem, the solution of the nonlinear set of equations normally takes up most of the computing time. Because the Newton–Raphson (N–R) method has the quadratic convergence characteristics, it is commonly used for solving nonlinear FEM equations. It was shown that when the vector potential is used, the N–R method could always converge. When the scalar potential is used, it might lead to a very unstable iteration process [1]. In order to ensure the convergence, a damped iterative algorithm with a proper relaxation factor has to be used [2]–[4].

At each iterative step of the N–R method, a set of linear algebraic equations is required to be solved. If the triangular decomposition method is used, a modified N–R method can be employed to reduce the computing time [5]. In the modified N–R method, it is not necessary to recalculate the global Jacobian matrix and its inverse (triangularization) at each iteration. Although the total number of iterations will be increased in the modified N–R method, the total computing time is still shorter since it is the triangularization of the Jacobian matrix, which normally takes up most of the CPU time when solving linear equations.

Compared with direct methods, the incomplete Cholesky-conjugate gradient (ICCG) algorithm is a more efficient iterative solver for large sparse linear equations. In the ICCG algorithm, only the nonzero elements are stored, and the algorithm converges very quickly. Nowadays, the N–R coupled with the ICCG has been popularly used as the solver of FEM problems.

In time-stepping FEM for time-domain solutions, the information from the last step can be successfully used to reduce the computing time when solving the current FEM equations. Several methods have also been proposed to reduce the computing time of the time-stepping FEM when using N–R and ICCG methods by the authors [6]. Those methods are used mainly to keep the topology of the FEM mesh unchanged throughout the time-stepping process, and the address vectors of the sparse matrix are then kept unchanged during the rotor movement. The predicted values of the unknowns are used conveniently as the initial data of the N–R algorithm. In this paper, a method to control the prespecified tolerances of the N–R iteration and the ICCG iteration in the time-stepping computation is further explored. A formula to predict the error tolerances of the ICCG algorithm is put forward. For the problems with which the N–R method can converge to a solution, the proposed method will reduce the computing time substantially without sacrificing the accuracy of the final FEM solutions. An induction motor with saturated iron parts is studied and reported in this paper to demonstrate the usefulness of the proposed method.

II. Basic Equations Arising From the Time Stepping FEM

The transverse electromagnetic field in induction machines can be represented by a two-dimensional (2-D) FEM formulation. If the rotor has skewed rotor bars, a multislice technique, which represents the motor with several slices of 2-D FEM, could be used to obtain the field solutions simultaneously for all the slices [7]. The electric circuit equations of the stator windings and the rotor cage can be coupled with the FEM equations

Manuscript received July 5, 2001; revised October 25, 2001. This work was supported by the Hong Kong Polytechnic University.

W. N. Fu was with the Industrial Centre, The Hong Kong Polytechnic University, Hong Kong. He is now with Ansoft Corporation, Pittsburgh, PA 15219 USA (e-mail: wfu@ansoft.com).

S. L. Ho and H. L. Li are with the Electrical Engineering Department, The Hong Kong Polytechnic University, Hong Kong (e-mail: eeslho@inet.polyu.edu.hk; eehlli@polyu.edu.hk).

H. C. Wong is with the Industrial Centre, The Hong Kong Polytechnic University, Hong Kong (e-mail: ichcwong@polyu.edu.hk).

Publisher Item Identifier S 0018-9464(02)02366-X.

together. If the interbar currents between the rotor bars have to be found, the rotor cage can be represented by a network [8]. The FEM equations of the magnetic field, together with the external circuit equations and the torque balance equation, will lead to the following large system of equation sets [7]:

$$CX + D\frac{\partial X}{\partial t} = P \tag{1}$$

where X is the unknown column vector to be evaluated. It includes the magnetic vector potentials, currents, voltages, rotor position, and rotor speed. C and D are sparse coefficient matrices. P is a column vector associated with, respectively, the input voltages and input currents for the voltage and current driven problems. Here, (1) becomes an initial value problem only. The backward Euler formula can be used readily because it has a good convergence rate. If the solution at the $(k-1)$th step is known, then X^k can be obtained from

$$\left[C^k + \frac{D^k}{(\Delta t)^k}\right]X^k = P^k + \frac{D^k}{(\Delta t)^k}X^{k-1} \tag{2}$$

where $(\Delta t)^k = t^k - t^{k-1}$ is the time step length. The coefficient matrix of (2) is a symmetrical one.

III. Solution Methods

A. N–R Method for the Nonlinear Problem

In the time-stepping method, the system equations will be solved iteratively. Because one has to consider the nonlinear characteristics of the reluctivity of the magnetic materials, the set of (2) is solved by using the N–R method at each time step:

$$J_n^k(\delta X)_n^k = P_{n-1}^k + \frac{D_{n-1}^k}{(\Delta t)^k}X^{k-1} - \left[C_{n-1}^k + \frac{D_{n-1}^k}{(\Delta t)^k}\right]X_{n-1}^k \tag{3}$$

where

$$X_n^k = X_{n-1}^k + (\delta X)_n^k \tag{4}$$

J is the Jacobian matrix of (2), and n is the iteration number of the N–R method.

The initial values of the solution X^k can be predicted by the Euler formula

$$X_{(0)}^k = X^{k-1} + \left(\frac{dX}{dt}\right)^{k-1}(\Delta t)^k. \tag{5}$$

Because the backward Euler formula at the $(k-1)$th step is

$$\left(\frac{dX}{dt}\right)^{k-1} = \frac{X^{k-1} - X^{k-2}}{(\Delta t)^{k-1}} \tag{6}$$

the $(dX/dt)^{k-1}$ in (5) can be easily obtained from the results of the $(k-1)$th and $(k-2)$th steps.

The iteration of the N–R method will stop if

$$(\varepsilon_{NR})_n = \frac{1}{N_X}\left|P_{n-1}^k + \frac{D_{n-1}^k}{(\Delta t)^k}X^{k-1} - \left[C_{n-1}^k + \frac{D_{n-1}^k}{(\Delta t)^k}\right]X_{n-1}^k\right| \le \varepsilon_{NR\_\text{allowed}} \tag{7}$$

where N_X is the number of the total unknowns, $(\varepsilon_{NR})_n$ is the error of the N–R method at the nth iteration cycle, and $\varepsilon_{NR\_\text{allowed}}$ is the error tolerance of the N–R method.

The error tolerance of the N–R method $\varepsilon_{NR\_\text{allowed}}$ is specified in advance, but it will be adjusted during the time-stepping process. If $(\varepsilon_{NR})_n$ cannot be reduced below the initial $\varepsilon_{NR\_\text{allowed}}$, it means that an increase in the iteration number of the N–R method will not reduce the value of ε_{NR} of the N–R method. Hence, $\varepsilon_{NR\_\text{allowed}}$ can be increased automatically according to the information of the last step.

B. ICCG Method for Solving the Algebraic Equations

At each iteration cycle of the N–R method, the ICCG algorithm is used to solve the large algebraic equation set. For iterative solvers, only the upper triangular coefficient matrix with nonzero elements is needed to be stored. Hence, the computing time and the computer storage can be greatly reduced. In the ICCG method, one should first obtain an approximate factorization of the coefficient matrix. To reduce the storage and reduce the computing time when factorizing the matrix, the incomplete Cholesky factorization is used. The coefficient matrix of the algebraic equation set can be expressed as

$$J = LL^T + E \tag{8}$$

where E is the error matrix defined by $(e_{ij})_{N_x \times N_x}$. $L = (l_{ij})_{N_x \times N_x}$ is the lower triangular matrix of the incomplete Cholesky factorization. It has the same sparse characteristics of the coefficient matrix J, which means that if $a_{ij} = 0$, then $l_{ij} = 0$.

Because the nonzero elements of the coefficient matrix $L^{-1}AL^{-T}$ will be distributed mainly near the diagonal, then if the conjugate gradient algorithm is used to solve the equation

$$\begin{aligned} &L^{-1}J_n^kL^{-T}L^T(\delta X)_n^k \\ &= L^{-1}\left\{P_{n-1}^k + \frac{D_{n-1}^k}{(\Delta t)^k}X^{k-1} - \left[C_{n-1}^k + \frac{D_{n-1}^k}{(\Delta t)^k}\right]X_{n-1}^k\right\} \end{aligned} \tag{9}$$

it will have good convergence. The iteration formulas are [9]

$$Q_1 = \left(LL^T\right)^{-1} r_0 \tag{10}$$

$$\beta_{m-1} = \frac{r_{m-1}^T\left(LL^T\right)^{-1} r_{m-1}}{r_{m-2}^T\left(LL^T\right)^{-1} r_{m-2}} \tag{11}$$

$$Q_m = \left(LL^T\right)^{-1} r_{m-1} + \beta_{m-1}P_{m-1} \tag{12}$$

$$\alpha_m = \frac{r_{m-1}^T\left(LL^T\right)^{-1} r_{m-1}}{Q_m^T J Q_m} \tag{13}$$

$$(\delta X)_m = (\delta X)_{m-1} + \alpha_m P_m. \tag{14}$$

The iteration is controlled by a vector r defined as

$$\begin{aligned} r = {} & P_{n-1}^k + \frac{D_{n-1}^k}{(\Delta t)^k}X^{k-1} \\ & - \left[C_{n-1}^k + \frac{D_{n-1}^k}{(\Delta t)^k}\right]X_{n-1}^k - \left[J_n^k\right](\delta X)_n^k. \end{aligned} \tag{15}$$

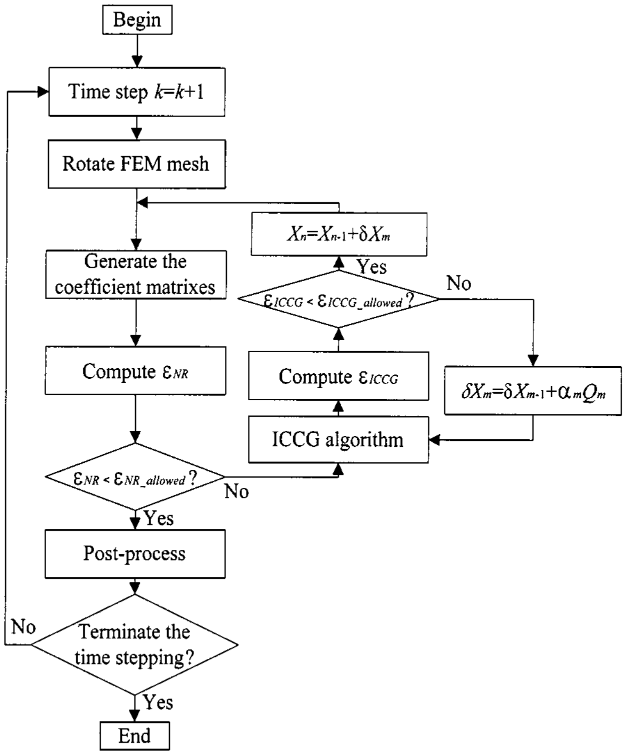

Begin
Time step $k=k+1$
Rotate FEM mesh
Generate the coefficient matrixes
Compute ε_{NR}
$\varepsilon_{NR} < \varepsilon_{NR\_allowed}$?
No
Yes
Post-process
Terminate the time stepping?
No
Yes
End
$X_n=X_{n-1}+\delta X_m$
Yes
$\varepsilon_{ICCG} < \varepsilon_{ICCG\_allowed}$?
No
Compute ε_{ICCG}
$\delta X_m=\delta X_{m-1}+\alpha_m Q_m$
ICCG algorithm

Fig. 1. Block diagram of the solver.

The iteration error $\varepsilon_{\mathrm{ICCG}}$ of the ICCG algorithm can be determined by the value of r. If

$$(\varepsilon_{\mathrm{ICCG}})_n = \frac{1}{N_X}\,|r| \le \varepsilon_{\mathrm{ICCG\_allowed}} \tag{16}$$

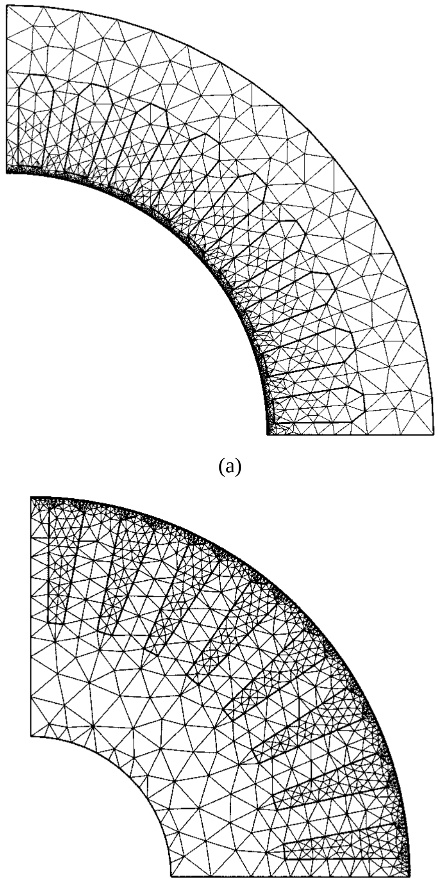
the iteration of the ICCG algorithm is terminated. The block diagram of the solver is shown in Fig. 1.

The iteration number of the ICCG algorithm is sensitive to a prespecified error tolerance $\varepsilon_{\mathrm{ICCG\_allowed}}$. If $\varepsilon_{\mathrm{ICCG\_allowed}}$ is very small, the iterative process of the ICCG algorithm will take a long time. Moreover, the aforementioned set of algebraic equations need not be solved precisely at the initial iteration steps of the N–R method. Therefore, the $\varepsilon_{\mathrm{ICCG\_allowed}}$ can be relatively large at the beginning of the N–R method, although it will be reduced later. If the total iteration number of the N–R method is N_{NR} (as predicted from the last step of the time step), the error tolerance of the ICCG algorithm at the nth iterative step of the N–R method can be adaptively controlled by

$$(\varepsilon_{\mathrm{ICCG\_allowed}})_n = (\varepsilon_{\mathrm{ICCG\_allowed}})_{n-1}\left[\frac{\varepsilon_{\mathrm{ICCG\_allowed}}}{(\varepsilon_{\mathrm{ICCG\_allowed}})_1}\right]^{1/(N_{NR}-1)} \tag{17}$$

where $(\varepsilon_{\mathrm{ICCG\_allowed}})_1$ is the error tolerance of the ICCG algorithm at the first step of the N–R iteration, and $\varepsilon_{\mathrm{ICCG\_allowed}}$ is the error tolerance of the ICCG algorithm at the last step of the N–R iteration. Because the last solution of the set of the nonlinear equations should still satisfy (7), the proposed method will not sacrifice the accuracy of the final solution.

The above-method for controlling the error tolerance of the ICCG algorithm is based on the assumption that at each time step, the number of the N–R iteration is basically the same. Usually, this assumption can be satisfied because physical quantities are generally changing very smoothly. Of course, there could be exceptions, such as in the case of a sudden change in field excitations. In those cases, the predicted error tolerance of the ICCG may be incorrect, and the computing time cannot be reduced. Since the field excitation normally changes smoothly and it is very rare to have sudden changes, the total computing time will nonetheless be reduced significantly by using the proposed method.


(a)

(b)

Fig. 2. (a) Stator mesh (1976 nodes, 3334 elements). (b) Rotor mesh (1884 nodes, 3152 elements).

IV. Numerical Experiments

The proposed method has been implemented to simulate the operation of a 2.2–kW induction motor with skewed rotor bars. The motor is fed by sinusoidal voltages. Multislice FEM is used to model the motor. For the sake of simplicity, the slice number is fixed at 4 for the following numerical experiments. The stator mesh and the rotor mesh are shown in Fig. 2. The total number of unknowns is 13 087. The step size is 38 μs. All the computations are run on a Pentium III/450 MHz computer. A typical computed flux plot is shown in Fig. 3.

The comparison between the normal method and the proposed method ($(\varepsilon_{\mathrm{ICCG\_allowed}})_1 = 10^3$) in obtaining the solution of one step of the time-stepping FEM is listed in Tables I –III. The corresponding computing time is given in Table IV.

TABLE I
ITERATION NUMBERS OF THE ICCG WHEN $\varepsilon_{ICCG\_allowed}$ IS FIXED. (INITIAL DATA OF THE N–R METHOD ARE ZERO)

N-R iteration no.	ε_{NR}	$(\varepsilon_{ICCE\_allowed})_n$	ε_{ICCG}	ICCG iteration no.
0	2.66×10^{7}			
1	1.04×10^{1}	3.00×10^{-4}	1.48×10^{-4}	631
2	1.47×10^{0}	3.00×10^{-4}	2.75×10^{-4}	863
3	2.55×10^{-1}	3.00×10^{-4}	2.52×10^{-4}	716
4	3.33×10^{-2}	3.00×10^{-4}	1.97×10^{-4}	678
5	4.12×10^{-3}	3.00×10^{-4}	2.24×10^{-4}	520
6	2.81×10^{-3}	3.00×10^{-4}	1.91×10^{-4}	376

TABLE II
ITERATION NUMBERS OF THE ICCG WHEN $\varepsilon_{ICCG\_allowed}$ IS FIXED. (INITIAL DATA OF THE N–R METHOD ARE BEING PREDICTED)

N-R iteration no.	ε_{NR}	$(\varepsilon_{ICCE\_allowed})_n$	ε_{ICCG}	ICCG iteration no.
0	2.88×10^{4}	-	-	-
1	2.74×10^{1}	3.00×10^{-4}	2.80×10^{-4}	650
2	3.49×10^{-2}	3.00×10^{-4}	2.75×10^{-4}	646
3	2.19×10^{-3}	3.00×10^{-4}	2.39×10^{-4}	467

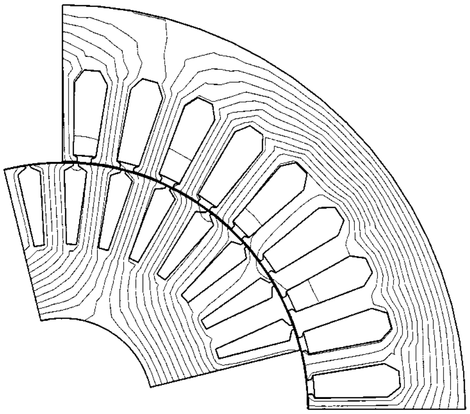


Fig. 3. Computed flux distribution during on-load operation.

The eventual error tolerance of the N–R method is 3.0×10^{-3}, and that of the ICCG is 3.0×10^{-4}.

Usually, it is difficult to choose the initial ICCG error tolerance. However, the choice of the error tolerance will affect the computing time when using the proposed method. The computing time required for different values of $(\varepsilon_{ICCG\_allowed})_1$ is shown in Fig. 4. One can see that the range of the initial ICCG error tolerances that can be chosen is quite large and that the proposed method is suitable for many applications.

From Table IV and Fig. 4, one can see that the proposed methods can effectively reduce the CPU time by up to 50% of that required to obtain the solutions of the time-stepping FEM model at each time step.

TABLE III
ITERATION NUMBERS OF THE ICCG OF THE PROPOSED METHOD

N-R iteration no.	ε_{NR}	$(\varepsilon_{ICCE\ allowed})_n$	ε_{ICCG}	CG iteration no.
0	2.05×10^{4}	-	-	-
1	5.91×10^{2}	1.00×10^{3}	5.90×10^{2}	3
2	4.88×10^{-1}	5.48×10^{-1}	4.87×10^{-1}	50
3	2.61×10^{-3}	3.00×10^{-4}	2.25×10^{-4}	623

TABLE IV
COMPARISON OF THE CPU TIMES

	CPU time
Normal method (X_0=0)	114.8 s
Normal method (X_0 being predicted)	57.4 s
Proposed method	23.7 s

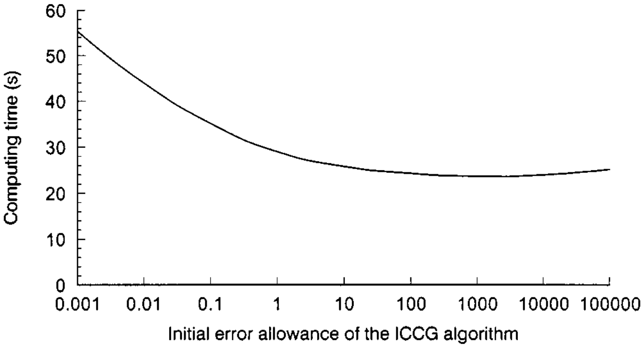


Fig. 4. Computing time versus initial error tolerance of the ICCG algorithm.

V. CONCLUSION

When using iterative methods to solve nonlinear equations and linear algebraic equations, the prespecified error tolerance will greatly affect the computing time. In the proposed time-stepping FEM computation, the error tolerance of the N–R method can be predicted from the last step, thereby avoiding the use of an unduly small error tolerance. At each iteration for the N–R method, the error tolerance of the ICCG can be suitably adjusted during the N–R iteration. Hence, the total computing time for the solution of the system equations can be reduced significantly, whereas the accuracy of the solution can be kept unchanged. The proposed method can also be used to deal with similar problems if nonlinear system equations are to be solved by the N–R and the ICCG methods.

REFERENCES

[1] C. Neagoe and F. Ossart, “Analysis of convergence in nonlinear magnetostatic finite elements problems,” *IEEE Trans. Magn.*, vol. 30, pp. 2865–2868, Sept. 1994.

[2] J. O’Dwyer and T. O’Donnell, “Choosing the relaxation parameter for the solution of nonlinear magnetic field problems by the Newton–Raphson method,” *IEEE Trans. Magn.*, vol. 31, pp. 1484–1487, May 1995.

[3] K. Fujiwara, T. Nakata, N. Okamoto, and K. Muramatsu, “Method for determining relaxation factor for modified Newton–Raphson method,” *IEEE Trans. Magn.*, vol. 29, pp. 1962–1965, Mar. 1993.

[4] L. Janicke and A. Kost, “Convergence properties of the Newton–Raphson method for nonlinear problems,” *IEEE Trans. Magn.*, vol. 34, pp. 2505–2508, Sept. 1998.

[5] O. A. Mohammed, “Fast solution techniques in 3D electromagnetic field problems,” in *Proc. IEEE Southeastcon*, Apr. 1988, pp. 322–326.

[6] S. L. Ho, W. N. Fu, and H. C. Wong, “Reduction of computing time in time-stepping finite element model of magnetic field computation of electric machines,” in *Proc. Ninth Biennial IEEE Conf. Electromagn. Field Comput.*, Milwaukee, WI, June 2000.

[7] S. L. Ho and W. N. Fu, “A comprehensive approach to the solution of direct-coupled multi-slice model of skewed induction motors using time-stepping eddy-current finite element method,” *IEEE Trans. Magn.*, vol. 33, pp. 2265–2273, May 1997.

[8] S. L. Ho, H. L. Li, and W. N. Fu, “Inclusion of inter-bar currents in a network-field coupled time-stepping finite element model of skewed rotor induction motors,” *IEEE Trans. Magn.*, vol. 35, pp. 4218–4225, Sept. 1999.

[9] D. S. Kershaw, “The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations,” *J. Comput. Phys.*, vol. 26, pp. 43–65, 1978.