Reducing the Computation Time of Nonlinear Problems by an Adaptive Linear System Tolerance

Hans Vande Sande, Wim Deprez, Jeroen De Coster, François Henrotte, and Kay Hameyer, *Senior Member, IEEE*

***Abstract*—Within the finite-element framework, nonlinear magnetic problems are often solved by an iterative line search strategy. The efforts to achieve convergence concentrate on the selection of an adequate relaxation factor. The line search is performed along a direction obtained by solving a system of linear equations. However, it is not required to compute this intermediate solution with a high accuracy, to ensure convergence. This paper shows how the accuracy of the solver can be modified at each nonlinear iteration, in order to reduce the overall computation time.**

***Index Terms*—Nonlinear magnetics, optimization methods.**

I. Introduction

NONLINEAR problems are common in computational magnetics. When formulated in the finite-element framework, they give rise to systems of nonlinear equations, of which the solution is often obtained by an iterative procedure. Each iteration essentially consists of two steps:

- the solution of a system of linear equations, in order to determine a descent search direction;
- the line search procedure along that direction, in order to maximize the overall performance.

For reducing the computation time, one usually considers the second step [1]–[3]. However, methods that additionally consider the first step for achieving that goal have already been developed as well. These so-called *inexact Newton methods* exploit the fact that it is not required to compute the line search direction exactly for guaranteeing convergence [3]. This can be done by steadily decreasing the relative linear system solver tolerance in a particular way. Doing so may have a considerable impact on the overall computation time, especially if the initial iterate is not close to the solution of the problem. This idea is generally applicable to any type of nonlinear problem. Here, one focuses on solving a typical nonlinear time-harmonic problem. The problem is first solved by using the theoretical adaption scheme for the relative linear system solver tolerance. By additionally concentrating on the actual convergence rates, it is shown in this paper that an efficiency-based empirical approach can even improve on this.

II. Definitions

The solution procedure contains two nested loops. At the kth *nonlinear iteration* of the outer loop, the system of linear equations

$$\mathbf{J}_k \mathbf{d}_k = -\mathbf{r}_k \tag{1}$$

must be solved, with $\mathbf{J}_k$ the Jacobian of the *nonlinear residual* $\mathbf{r}_k$ (or its approximation) and $\mathbf{d}_k$ the line search direction. This system is solved by an inner loop. The initial guess for $\mathbf{d}_k$ is denoted by $\mathbf{d}_{k,0}$, the second index indicating the *linear iteration* number. Subsequent iterates from the linear system solver are indicated as $\mathbf{d}_{k,n}$. The vector

$$\mathbf{r}_{k,n} = \mathbf{J}_k \mathbf{d}_{k,n} + \mathbf{r}_k \tag{2}$$

defines the *linear residual* at the nth linear iteration of the kth nonlinear iteration. The linear system solver is terminated when

$$\frac{\|\mathbf{r}_{k,n}\|}{\|\mathbf{r}_k\|} < \epsilon_k \tag{3}$$

with ϵ_k the *(relative) linear system solver tolerance*. The sequence $\{\epsilon_k\}$ is known as the *forcing sequence* [3].

III. Fixed Linear System Solver Tolerance

To illustrate the need of an adaptive linear system solver tolerance, the simulation of the short-circuit operation of the 400-kW four-pole induction motor in Fig. 1 is simulated, using a fixed value for ϵ_k. The stator and rotor are stacked with a nonoriented ferromagnetic material, whose magnetization curve is nonlinear. The relative permeability approximately equals 1700 up to 500 A/m. At higher field strength, the material saturates, e.g., $B = 1.75$ T at $H = 15\,000$ A/m, causing the time-harmonic problem to be nonlinear. The problem is entirely solved by the Picard method, i.e., $\mathbf{J}_k$ in (1) is approximated by $\mathbf{K}_k + j\mathbf{L}$, with $\mathbf{K}_k$ the (field dependent) stiffness matrix and $\mathbf{L}$ the eddy-current matrix [4]. Alternatively, the Newton–Raphson method could be applied as well [5]. A cubic line search procedure determines the relaxation parameter [3]. The associated complex symmetric system of equations is iteratively solved using the ILU-preconditioned COCG-algorithm [6]. For the analysis, the mathematical software library PETSc (Portable Extensible Toolkit for Scientific Computing) has been used [7].

Manuscript received July 1, 2003. This paper presents research results of the Belgian Fund for Scientific Research and the Belgian programme on Interuniversity Poles of Attraction initiated by the Belgian State, Prime Minister's Office, Science Policy Programming.

H. Vande Sande is with the Flanders' Mechatronics Technology Centre, B-3001 Leuven, Belgium (e-mail: hans.vandesande@fmtc.be).

W. Deprez, J. De Coster, and F. Henrotte are with the Department ESAT, Katholieke Universiteit Leuven, B-3001 Leuven, Belgium (e-mail: francois.henrotte@esat.kuleuven.ac.be).

K. Hameyer is with the Institut für Elektrische Machinen (IEM), Rheinisch-Westfälische Technische Hochschule Aachen, D-52056 Aachen, Germany (e-mail: Kay.Hameyer@iem.RWTH-Aachen.de).

Digital Object Identifier 10.1109/TMAG.2004.825005

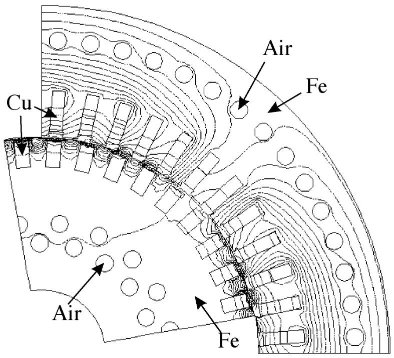

Fig. 1. Magnetic flux lines in a 400-kW induction motor under short-circuit operation.

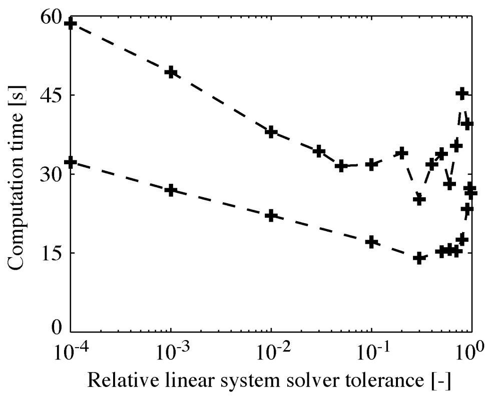

Fig. 2. Total computation time as a function of the relative tolerance ϵ_k of the linear system solver, at two different saturation levels.

Increasing ϵ_k causes a deviation of the computed direction $\mathbf{d}_k$ from the exact line search direction. In fact, the higher ϵ_k is, the smaller the number of linear iterations is, the closer $\mathbf{d}_k$ is to its initial guess $\mathbf{d}_{k,0}$. Often, $\mathbf{d}_{k,0} = -\mathbf{r}_k$ is taken, i.e., the steepest descent direction, since this direction is available without any additional computational efforts. In other algorithms, the steepest descent direction is obtained after one linear iteration. Hence, the higher ϵ_k is, the more the convergence rate reduces toward linear convergence. On the other hand, as long as the deviation between the computed $\mathbf{d}_k$ and the exact line search direction remains small, the convergence rate is hardly affected.

Fig. 2 shows the overall computation time as a function of a fixed ϵ_k, at two different current levels. The lower curve corresponds to the case with the smallest current and never requires relaxation, whereas the simulations for the upper curve require significant relaxation, due to local saturation effects. Irrespective of the observed oscillations, it is obvious that an optimal tolerance seems to exist. Moreover, it has a rather high value (≈0.3). Since the optimal value of a fixed ϵ_k is not known in advance, it is suggested to adjust ϵ_k at each nonlinear iteration in such a way that better performance may be expected.

IV. Adaptive Linear System Solver Tolerance

A. Practical Analysis

In order to elaborate an efficient update algorithm for ϵ_k, two simulations at the largest current level in Fig. 2 are studied in detail: one with $\epsilon_k = 0.5$, the other with $\epsilon_k = 0.01$. For both, the norm of the linear residual is plotted in Fig. 3 as a function of the iteration number.

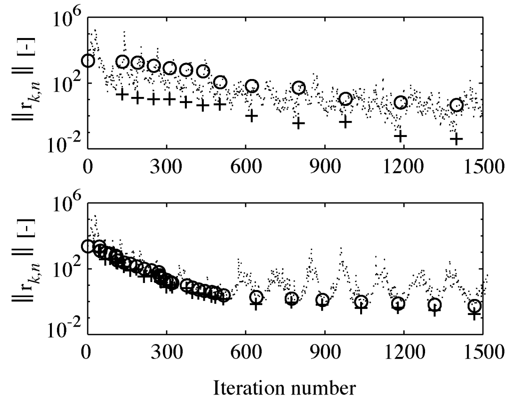

Fig. 3. The norm of the linear residual while iterating with a fixed linear system solver relative tolerance of 0.01 (top) and 0.5 (bottom).

- The circles indicate the norm of the linear residual at the beginning of a linear system solve, i.e., $\|\mathbf{r}_{k,0}\|$. Since $\mathbf{d}_k^0 = \mathbf{0}$ is taken as initial solution for the linear system, it follows that $\|\mathbf{r}_k\| = \|\mathbf{r}_{k,0}\|$. Hence, these circles represent the norm of the nonlinear residual as well and their sequence indicates the convergence of the nonlinear algorithm.
- The plus-signs indicate the norm of the linear residual at the end of a linear system solve, i.e., $\|\mathbf{r}_{k,N}\|$, with N the number of linear iterations required to decrease the linear residual by a factor ϵ_k.
- Between $+$ and $\circ$, the line search algorithm is performed and the Jacobian is updated.

The upper part of this figure, obtained for $\epsilon_k = 0.01$, illustrates that the effort for computing the descent direction is high when compared to the decrease it yields for the nonlinear residual. This defines a first type of efficiency, characterized by the ratio

$$\rho_k = \frac{\log(\|\mathbf{r}_k\|) - \log(\|\mathbf{r}_{k+1}\|)}{\log(\|\mathbf{r}_k\|) - \log(\|\mathbf{r}_{k,N}\|)}. \tag{4}$$

The nominator is a measure for the obtained reduction of the nonlinear residual norm. The denominator is a measure for the computational cost for obtaining that reduction. Obviously, ρ_k is low here. The lower part of the figure, obtained with $\epsilon = 0.5$, yields a much higher value of this ratio. Since the latter converges faster, it is concluded to increase ϵ_k if ρ_k is too low. On the other hand, if ρ_k is too high, the linear system solver is terminated at a moment that the nonlinear residual could be further decreased. Therefore, high values of ρ_k suggest a reduction of ϵ_k.

Next to these observations, the lower part of Fig. 3 shows that many short iterations are performed at the beginning. This increases the ratio of the time for building the system of linear equations to the time for solving it. The ratio

$$\sigma_k = \frac{t_{k,N} - t_{k,0}}{t_{k,N} - t_{k-1,N}} \tag{5}$$

with $t_{k,N}$ the time instant at which the Nth linear iteration of the kth nonlinear iteration begins, characterizes this second type of efficiency. It is one reason for the increased computation time at high values of ϵ_k in Fig. 2, besides the fact that increasing ϵ_k

gradually transforms the Picard method into a steepest descent method. As a consequence, it is recommended to decrease ϵ_k if σ_k is too low.

In the frame of this work, many experiments have been performed in order to estimate how ϵ_k could be adjusted. The following algorithm is suggested here.

- Neither small nor high values of ρ_k are desired. Let ρ_k^{opt} be the optimal value of this parameter.
- If $\rho_k > \rho_k^{\text{opt}}$, the weight

$$w = \frac{\rho_k - \rho_k^{\text{opt}}}{1 - \rho_k^{\text{opt}}} \tag{6}$$

is a number between 0 and 1. The higher w is, the more ϵ_k should be decreased. From Fig. 2, it is decided to restrict this decrease to a factor 2. Therefore

$$\epsilon_{k+1} = \frac{\epsilon_k}{2^w} \tag{7}$$

is proposed here.
- If $\rho_k < \rho_k^{\text{opt}}$, the weight

$$w = \frac{\rho_k^{\text{opt}} - \rho_k}{\rho_k^{\text{opt}}} \tag{8}$$

is a number between 0 and 1. The higher w is, the more ϵ_k should be increased. When ϵ_k is very small, Fig. 2 reveals that the system tolerance may be significantly increased, up to several orders of magnitude. However, to avoid a too low value of σ_{k+1}, it is suggested to limit ϵ_{k+1} to 0.9 by applying

$$\epsilon_{k+1} = \epsilon_k^{1-w} 0.9^w \tag{9}$$

here.
- If σ_k is low, it is desired to increase the linear system solving time significantly, requiring a relatively large reduction of the system tolerance. To avoid that ρ_{k+1} is too low, the reduction is restricted to a factor of 20, by setting

$$\epsilon_{k+1} = \epsilon_k^{1-w} \left[1 - 0.95\sqrt{(1-\sigma_k)}\right]. \tag{10}$$

When resimulating the short-circuit operation of the induction machine using this algorithm, with ϵ_k initiated with the same relative tolerances as in Fig. 2, the averages of the computation time become 17.0 and 32.4 s, with a standard deviation of 0.8 and 3.0 s, respectively. Compared to these, the minimal computation times in Fig. 2 are 14.1 and 25.2 s, but their average value and standard deviation is much higher. This shows that the overall computation time appears to be almost independent of the initial value of the linear system tolerance.

B. Theoretical Analysis

It is possible to analyze the problem of adjusting ϵ_k in a mathematical way. The details of this analysis are given in [3]. At this place, only the results are mentioned.

- Under the usual conditions, that the residual is continuously differentiable in a neighborhood of the solution, that the Jacobian[1] is positive definite at the solution and that no relaxation is required from a certain iteration on, one can show that the subsequent iterates at least converge linearly to the exact solution, provided $\epsilon_k < 1$.
- If in addition the forcing sequence approaches 0 for increasing values of k, the asymptotic convergence is at least superlinear. The latter can be obtained for example by setting

$$\epsilon_k = \min\left(0.5, \sqrt{\|\mathbf{r}_k\|}\right). \tag{11}$$

- If $\epsilon_k = O(\|\mathbf{r}_k\|)$, the asymptotic convergence rate is quadratic, provided $\mathbf{J}_k$ is the exact Jacobian. This can be achieved, for example, by setting

$$\epsilon_k = \min\left(0.5, \|\mathbf{r}_k\|\right). \tag{12}$$

These theorems suppose that no relaxation is required. However, the convergence rate seems to be reasonable as well in practical line search algorithms with relaxation [3], [8].

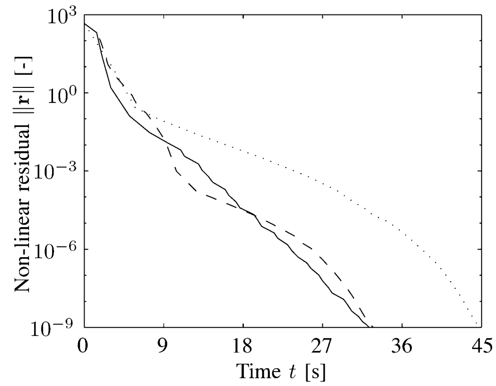

Fig. 4. Norm of the residual as a function of time, when the current level in the coils is low, for the method with fixed ϵ_k (dotted), adaptive ϵ_k (solid), and $\epsilon_k = \min(0.5, \sqrt{\|\mathbf{r}_k\|})$ (dashed).

C. Results

The convergence of three methods is compared:

- the first method uses a fixed value for ϵ_k;
- the second method adjusts ϵ_k according to the proposed algorithm;
- the third method imposes $\epsilon_k = \min(0.5, \sqrt{\|\mathbf{r}_k\|})$, for achieving superlinear convergence of the Picard method.

They are compared for two different coil currents of the induction machine. The resulting convergence characteristics are plotted in Figs. 4 and 5. The values of the linear system tolerance are plotted in Figs. 6 and 7.

Fig. 4, obtained for the smallest current in the coils, obviously reveals the increased asymptotic convergence rate of the third method, compared to the others. Unfortunately, this effect only occurs when the residual norm already decreased by a factor 10^9. Moreover, the higher the current level, the less this beneficial property is observed, as illustrated by Fig. 5. It is therefore concluded that in practical terms, (11) or (12) does

[1] In the time-harmonic case, the Jacobian is only defined if the residual is split up in its real and imaginary part [5]. However, this splitting must not be performed for computing the line search direction in the Picard approach.

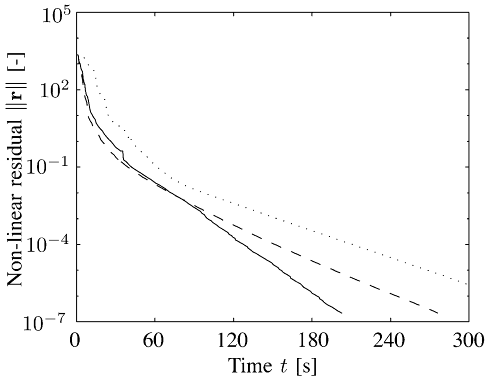

Fig. 5. Norm of the residual as a function of time, when the current level in the coils is high, for the method with fixed ϵ_k (dotted), adaptive ϵ_k (solid), and $\epsilon_k = \min(0.5, \sqrt{\|\mathbf{r}_k\|})$ (dashed).

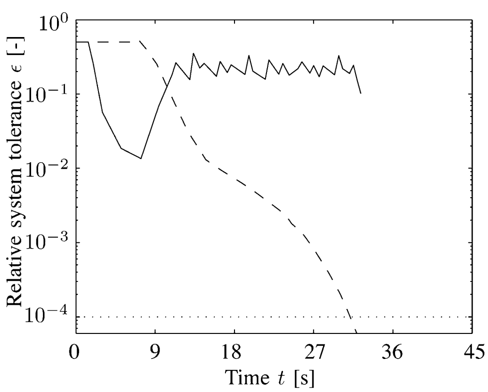

Fig. 6. Linear system solver tolerance as a function of time, when the current level in the coils is low, for the method with fixed ϵ_k (dotted), adaptive ϵ_k (solid), and $\epsilon_k = \min(0.5, \sqrt{\|\mathbf{r}_k\|})$ (dashed).

not play a significant role in decreasing the overall computation time. From Figs. 6 and 7, it follows that the proposed update algorithm for ϵ_k automatically evolves toward a relatively high value, where it slightly oscillates from one iteration to the other.

V. Conclusion

Numerically solving nonlinear problems involves two nested iterative loops. At each iteration, a linear system of equations is iteratively solved. Theoretically, the highest convergence rate is achieved when the forcing sequence steadily decreases toward zero. In practice, the convergence rate approaches its asymptotic value only when the residual norm already significantly decreased. Therefore, it is suggested to solve nonlinear problems with a relatively high value of the linear system solver tolerance. A novel algorithm which updates ϵ_k at each iteration, based on two efficiency indicators, is proposed. The overall computation times appears to be almost independent of the initial value of the linear system tolerance.

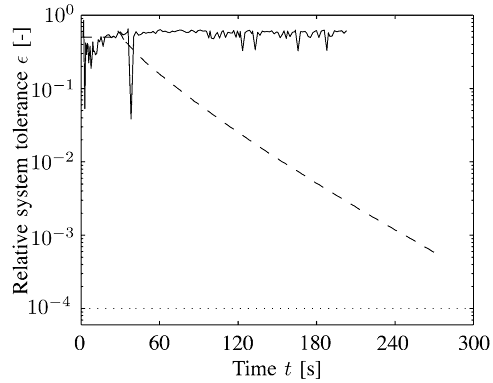

Fig. 7. Linear system solver tolerance as a function of time, when the current level in the coils is high, for the method with fixed ϵ_k (dotted), adaptive ϵ_k (solid), and $\epsilon_k = \min(0.5, \sqrt{\|\mathbf{r}_k\|})$ (dashed).

References

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

[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] J. Nocedal and S. J. Wright, *Numerical Optimization*, 1st ed, ser. Springer Series in Operations Research. New York: Springer, 1999.

[4] K. J. Binns, P. J. Lawrenson, and C. W. Trowbridge, *The Analytical and Numerical Solution of Electric and Magnetic Fields*. Chichester, U.K.: Wiley, 1992.

[5] D. Lederer, H. Igarashi, and A. Kost, "The Newton-Raphson method for complex equation systems," in *Proc. 7th Int. IGTE Symp. Numerical Field Calculation in Electrical Engineering (IGTE'96)*, Graz, Austria, Sept. 1996, pp. 391–394.

[6] 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, pp. 706–708, Mar. 1990.

[7] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, L. C.L. Curfmann McInnes, and B. F. Smith. (2001) PETSc Home Page. [Online]. Available: http://www.mcs.anl.gov/petsc

[8] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, "Inexact Newton methods," *SIAM J. Numer. Anal.*, vol. 19, pp. 400–408, 1982.