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

H. Vande Sande, W. Deprez., J. Decoster, F. Henrotte and K. Hameyer
Katholieke Universiteit Leuven, Dept. ESAT, Div. ELECTA, Kasteelpark Arenberg 10
B-3001 Heverlee-Leuven, Belgium
Email: hans.vandesande@esat.kuleuven.ac.be

***Abstract*— Within the finite element framework, non-linear 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 linear system of 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 non-linear iteration, in order to minimize the overall computation time.**

I. Introduction

NON-LINEAR problems are common in computational magnetics. When formulated in the finite element framework, they give rise to non-linear systems of equations, of which the solution is often obtained by an iterative line search procedure. Each individual cycle essentially consists of two phases: the solution of a linear system of equations in order to determine the Newton direction and the line search along that direction for a better approximation of the solution of the non-linear problem. For reducing computation time, one usually considers the second phase [1], [2], [3]. However, it is not necessary to compute the exact solution of the linear system to obtain convergence. It is rather recommended to modify the accuracy of the iterative linear solver at each non-linear step. This paper discusses how this can be done at best.

II. Test Problem

The approach to minimize the computational efforts by an adaptive system tolerance is applied to the simulation of the short-circuit operation of a 400 kW four-pole induction motor. The geometry and the computed flux lines are shown in Fig. 1. The triangular finite element mesh contains 1419 nodes and 2772 elements. The magnetic vector potential is discretized by first order nodal elements. The non-linear time-harmonic problem is solved by the Picard-method (successive substitution). The ILU-preconditioned COCG-algorithm is applied to solve the associated complex symmetric system of equations [4]. At each non-linear iteration, the relaxation parameter is determined by the cubic line search method [3]. For the analysis, the mathematical software library PETSc (Portable Extensible Toolkit for Scientific Computing) has been used [5].

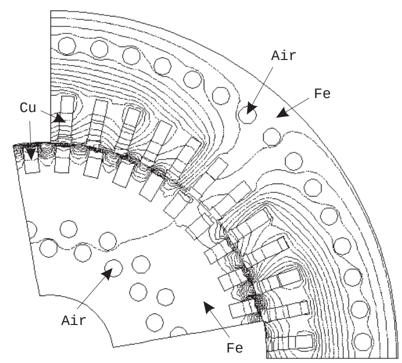

Fig. 1. The magnetic flux lines in a 400 kW induction motor under short-circuit operation.

III. Fixed Linear System Tolerance

The solution $\mathbf{d}_k$ of the linear system at the k^{th} non-linear iteration is a direction for the line search algorithm. For achieving convergence, it is only required that this direction is *descent*. This means that a relaxation parameter $\alpha_k \in [0, 1]$ can be determined such that $\|\mathbf{r}(\mathbf{A}_k + \alpha_k \mathbf{d}_k)\| < \|\mathbf{r}(\mathbf{A}_k)\|$, with $\mathbf{A}_k$ the k^{th} non-linear approximation and $\mathbf{r}$ theresidual. The exact solution $\mathbf{d}^*_k$ of the linear system is a quasi-Newton-direction here, as the Picard approach is used. Reducing the accuracy of the linear system solver causes a deviation of the computed direction $\mathbf{d}_k$ towards the steepest descent direction [3]. As long as this deviation remains small, the non-linear convergence rate is hardly affected.

Fig. 2 shows the overall computation time as a function of the linear system solver relative tolerance ϵ, at two different saturation levels. The lower curve corresponds to the case with the smallest current and requires no relaxation, whereas the upper curve requires a significant relaxation. Irrespective of the observed oscillations, it is obvious that an optimal tolerance exists. Moreover, it has a rather high value (≈ 0.3).

IV. Adaptive Linear System Tolerance

The norm of the residual, for two different values of ϵ is plotted in Fig. 3, as a function of the iteration number. Between $+$ and $\circ$ the system matrix is updated. Hence, the circles indicate the non-linear residuals $\mathbf{r}_k$ of the iteration

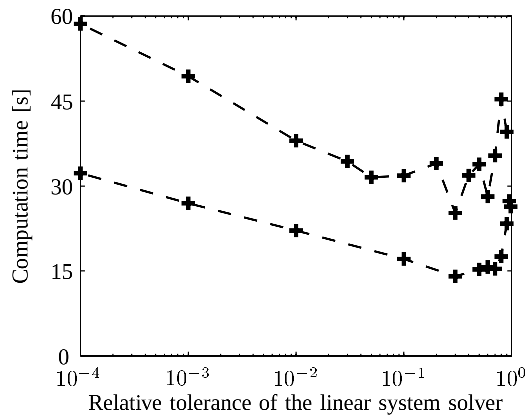

Fig. 2. The computation time as a function of the relative tolerance of the linear system solver, at two different saturation levels.

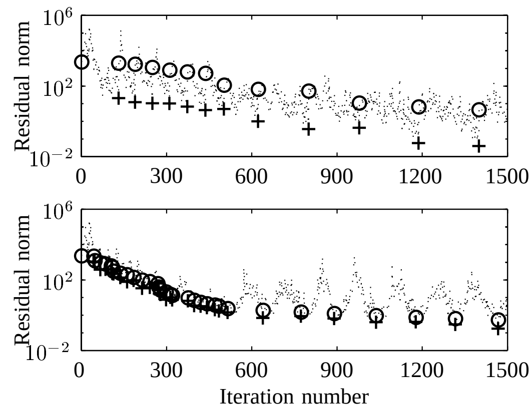

Fig. 3. The norm of the linear residual while iterating with a fixed linear system solver relative tolerance of 0.01 (top) and 0.5 (down) .

process, whereas the crosses indicate the last residual $\mathbf{r}_k^{\mathrm{lin},N}$ obtained when solving the system. The upper part of the figure, which counts for a tolerance $\epsilon = 0.01$, illustrates that the effort for solving the linear system is high, when compared to the decrease it yields for the non-linear residual. As a consequence, the ratio

$$\rho_k = \frac{\log(\|\mathbf{r}_k\|) - \log(\|\mathbf{r}_{k+1}\|)}{\log(\|\mathbf{r}_k\|) - \log(\|\mathbf{r}_k^{\mathrm{lin},N}\|)} \qquad (1)$$

is low. The lower part of the figure, obtained with $\epsilon = 0.5$, yields a much higher value of this ratio. As the latter converges faster, it is suggested to increase ϵ if ρ_k is low. However, if ρ_k is too high, the linear system solver is terminated at a moment that the non-linear residual could be further decreased. Therefore, high values of ρ_k suggest a reduction of ϵ.

Next to these observations, the lower part of Fig. 3 reveals that initially many short non-linear steps are performed. This increases the ratio of the time for building the linear system of equations to the time for solving it. Besides the fact that increasing ϵ gradually transforms the quasi-Newton method in a steepest descent method having slower convergence rates, it explains why the computation time increases at even higher values of ϵ in Fig. 2. As a consequence, it is recommended to decrease ϵ in an appropriate way if this situation occurs.

V. Results

The computations have been repeated taking the previous considerations into account. If $\rho_k > 0.6$, ϵ is divided by a factor between 1.0 and 2.0 . If $\rho_k < 0.6$, ϵ is multiplied by a factor such that ϵ cannot exceed 0.9 . The multiplication factors depend on the value of $\rho_k - 0.6$, which will be more thoroughly discussed in the full paper. Short non-linear steps are penalized by dividing ϵ by a factor between 1.0 and 2.0 . When initiating the computations with the same relative tolerances as in Fig. 2, the averages of the computation time become 17.0 s and 32.4 s, with a standard deviation of 0.8 s and 3.0 s respectively. Compared to these, the minimal computation times in Fig. 2 are 14.1 s and 25.2 s, but their average value is much higher. This obviously shows the improvement that can be obtained by modifying the relative tolerance of the linear system solver at each non-linear iteration.

VI. Conclusions

The computation time of a non-linear line search strategy is decreased by modifying the linear system solver relative tolerance at each new non-linear step. It is discussed which considerations should be taken into account. By the simulation of the short-ciruit test of an induction motor, it is illustrated that this technique yields computation times, which only slightly depend on the initial linear system solver tolerance.

Acknowledgments

The authors are grateful to the Belgian “Fonds voor Wetenschappelijk Onderzoek Vlaanderen” (project G.0427.98) and the Belgian Ministry of Scientific Research (IUAP No. P5/34).

References

[1] K. Fujiwara, T. Nakata, and N. Okamoto, “Method for determining relaxation factor for modified Newton-Raphson method,” *IEEE Transactions on Magnetics*, vol. 29, no. 2, 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 Transactions on Magnetics*, vol. 31, no. 3, pp. 1484–1487, May 1995.

[3] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer Series in Operations Research. Springer, New York, USA, 1st edition, 1999.

[4] 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 Transactions on Magnetics*, vol. 26, no. 2, pp. 706–708, Mar. 1990.

[5] S. Balay, W.D. Gropp, L. Curfman McInnes, and B.F. Smith, “PETSc users manual,” Tech. Rep. ANL-95/11 - Revision 2.1.2, Mathematics and Computer Science Division, Argonne National Laboratory, 2002, See http://www.mcs.anl.gov/petsc.