Solving Nonlinear Magnetic Problems Using Newton Trust Region Methods

H. Vande Sande, H. De Gersem, F. Henrotte, and K. Hameyer

***Abstract*—In this paper, a Newton trust region method is presented as an alternative to the Newton–Raphson method for solving nonlinear magnetic problems. Instead of underrelaxing the Newton step in a line search algorithm, the step is determined by minimizing a local quadratic model of the functional within a trust region. If the Newton step lies outside the trust region, a step with a smaller norm and different direction is computed. The size of the trust region plays a similar role as the relaxation factor in the line search approach. To ensure that the method converges, the trust region size is automatically adjusted from one iteration to the next one, depending on the local accuracy of the quadratic model. The trust region approach is applied for the simulation of an 8/6 switched reluctance motor.**

***Index Terms*—Magnetostatics, Newton–Raphson method, nonlinear magnetics, optimization methods.**

I. Introduction

NONLINEAR magnetostatic systems are described by

$$\nabla \times (\nu \nabla \times \vec{A}) = \vec{J} \tag{1}$$

with $\vec{A}$ being the vector potential (Vs/m), ν is the reluctivity tensor (Am/Vs), and $\vec{J}$ is the applied current density vector $(\mathrm{A/m^2})$. Equation (1) must be complemented with an appropriate gauge and appropriate boundary conditions in order to determine a unique solution [1].

A magnetostatic problem can be solved by considering its equivalent variational formulation

$$\min_{\vec{A}} \quad F(\vec{A}) \tag{2}$$

with the functional

$$F(\vec{A}) = \frac{1}{2}\int_{\Omega} \nu(\nabla \times \vec{A}) \cdot (\nabla \times \vec{A}) \mathrm{d}\Omega - \int_{\Omega} \vec{J} \cdot \vec{A} \mathrm{d}\Omega. \tag{3}$$

The functional F returns a scalar quantity, which is analogous to the enthalpy in thermodynamics. The first term represents the magnetic energy in the system. The second term represents the work done by the exciting currents imposed to the system. By discretizing the problem and interpolating the solution over all elements, (3) transforms into

$$F(\mathbf{A}) = \sum_{\mathrm{e}} \left(\int_{\Omega_{\mathrm{e}}} w_{\mathrm{m}}(\mathbf{A}) \mathrm{d}\mathbf{\Omega_e} \right) - \mathbf{A}^{\mathrm{T}}\mathbf{T} \tag{4}$$

with $\mathbf{A}$ being the vector of unknown nodal or edge values of the vector potential, w_{m} is the magnetic energy density $(\mathrm{J/m^3})$, and $\mathbf{T}$ is the source vector (A) [1]. For magnetic systems with nonhysteretic ferromagnetic materials, F is a nonlinear function of $\mathbf{A}$. This paper discusses methods for minimizing (4) and especially focuses on the trust region method.

II. Unconstrained Minimization

The direct minimization of (4) is an unconstrained minimization problem [2]. Among the gradient-based methods for solving such problems, two strategies are distinguished here: the *line search method* and the *trust region method*. Both start from an initial solution $\mathbf{A}_0$. At each subsequent nonlinear iteration k, a step $\mathbf{s}_k$ is determined, for which $F(\mathbf{A}_k + \mathbf{s}_k) < F(\mathbf{A}_k)$. The actual solution is updated from $\mathbf{A}_k$ to $\mathbf{A}_k + \mathbf{s}_k$.

For the computation of the step $\mathbf{s}_k$, line search and trust region methods both use a *local quadratic model* F_k^{qm} of F around the k^{th} nonlinear iterate $\mathbf{A}_k$. The model is defined by the second-order Taylor series expansion of F around $\mathbf{A}_k$

$$F_k^{\mathrm{qm}}(\mathbf{s}) = \mathbf{F}(\mathbf{A_k}) + \mathbf{s}^{\mathrm{T}}\nabla\mathbf{F}(\mathbf{A_k}) + \frac{\mathbf{1}}{\mathbf{2}}\mathbf{s}^{\mathrm{T}}\nabla^{\mathbf{2}}\mathbf{F}(\mathbf{A_k})\mathbf{s} \tag{5}$$

with $\mathbf{s}$ an increment for $\mathbf{A}_k$. Analytical formulas for the gradient ∇F and the Hessian $\nabla^2 F$ of magnetostatic problems are given in [1]. They are also often referred to as the residual and the Jacobian, respectively. Line search and trust region methods basically differ in the way they exploit this quadratic model.

III. Line Search Methods

A. Line Search Directions

At the beginning of a new iteration, line search algorithms compute a *descent direction* $\mathbf{d}_k$. This is a direction for which the functional value decreases close to the working point

$$\mathbf{d}_k^{\mathrm{T}} \nabla F(\mathbf{A}_k) < 0. \tag{6}$$

A graphical interpretation of the quadratic model (5) of a function of two independent variables illustrates this condition (Fig. 1). If the Hessian is positive definite, F_k^{qm} exhibits ellipsoidal isolines in a multidimensional parameter space.

Manuscript received June 18, 2002; revised November 28, 2002. This work was supported in part by the Belgian "Fonds voor Wetenschappelijk Onderzoek Vlaanderen" under Project G.0427.98, by the Belgian Ministry of Scientific Research by IUAP P5/34, and by the research council of the K.U. Leuven (the basic numerical research).

H. Vande Sande, F. Henrotte, and K. Hameyer are with the Division ELECTA, Department ESAT, Katholieke Universiteit Leuven, B-3001 Heverlee-Leuven, Belgium (e-mail: hans.vandesande@esat.kuleuven.ac.be).

H. De Gersem is with the Computational Electromagnetics Laboratory, Technische Universität Darmstadt, D-64289 Darmstadt, Germany.

Digital Object Identifier 10.1109/TMAG.2003.810212

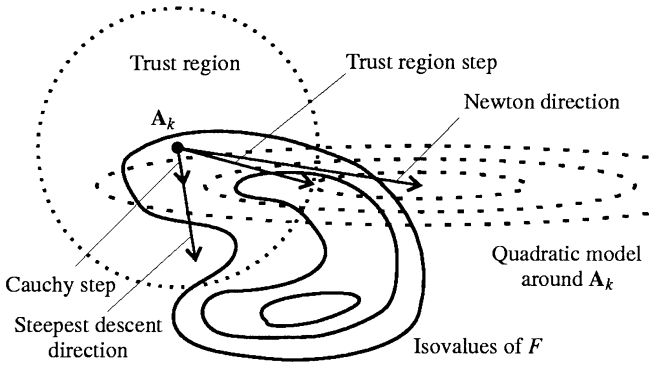

Fig. 1. Graphical interpretation of the local quadratic model, the steepest descent direction, the Newton direction, the Cauchy step, and the trust region step for a function F with two variables.

The *steepest descent direction* is perpendicular to the isoline through $\mathbf{A}_k$ and is defined by

$$\mathbf{d}_k^{\mathrm{SD}} = -\frac{\nabla F(\mathbf{A}_k)}{\|\nabla F(\mathbf{A}_k)\|}. \tag{7}$$

On the other hand, the *Newton direction* is a descent direction pointing toward the minimum of the quadratic model (Fig. 1)

$$\mathbf{d}_k^{\mathrm{N}} = -\left(\nabla^2 F(\mathbf{A}_k)\right)^{-1} \nabla F(\mathbf{A}_k). \tag{8}$$

Alternatively, the exact Hessian can be replaced by a positive definite approximation $\mathbf{B}_k$. In magnetostatics, this can be achieved, e.g., by omitting the nonlinear contribution to the Jacobian [1]. In that case

$$\mathbf{d}_k^{\mathrm{QN}} = -\mathbf{B}_k^{-1} \nabla F(\mathbf{A}_k) \tag{9}$$

is a descent direction pointing toward the minimum of a modified quadratic model. This direction is called a *quasi-Newton direction*.

B. Newton Line Search Step

The descent condition (6) is a local property. However, if the quadratic model accurately describes the functional in a sufficiently large neighborhood of $\mathbf{A}_k$, it is expected that $F(\mathbf{A}_k + \mathbf{d}_k^{\mathrm{N}}) < F(\mathbf{A}_k)$ also holds. As this cannot be ensured *a priori*, Newton (and quasi-Newton) line search methods proceed by determining an underrelaxation factor $\alpha_k \in [0, 1]$, such that $F(\mathbf{A}_k + \alpha_k \mathbf{d}_k^{\mathrm{N}}) < F(\mathbf{A}_k)$ is true. The new iterate $\mathbf{A}_{k+1}$ is then

$$\mathbf{A}_{k+1} = \mathbf{A}_k + \alpha_k \mathbf{d}_k^{\mathrm{N}} = \mathbf{A}_k + \mathbf{s}_k \tag{10}$$

with $\mathbf{s}_k$ the line search step.

C. Inexact Newton Line Search

Ideally, the underrelaxation factor is such that the new iterate $\mathbf{A}_{k+1}$ minimizes F along $\mathbf{d}_k$. However, this generally requires too many evaluations of F. Moreover, the convergence of the line search method can be guaranteed without computing the exact minimizer. By imposing a sufficient decrease on the functional value and its directional derivative, the so-called *Wolfe conditions*, convergence is obtained as well [2]

$$F(\mathbf{A}_k + \alpha_k \mathbf{d}_k^{\mathrm{N}}) \leq F(\mathbf{A}_k) + \mathrm{c}_1 \alpha_k \mathbf{d}_k^{\mathrm{N}^{\mathrm{T}}} \nabla F(\mathbf{A}_k) \tag{11}$$

$$\mathbf{d}_k^{\mathrm{N}^{\mathrm{T}}} \nabla F(\mathbf{A}_k + \alpha_k \mathbf{d}_k^{\mathrm{N}}) \geq \mathrm{c}_2 \mathbf{d}_k^{\mathrm{N}^{\mathrm{T}}} \nabla F(\mathbf{A}_k) \tag{12}$$

with $0 < c_1 < c_2 < 1$. Practical values are $c_1 = 10^{-4}$ and $c_2 = 0.9$. The *sufficient decrease condition* (11) ensures that the functional value decreases from one iteration to the next one. The *curvature condition* (12) prevents that too small underrelaxation factors are selected. Efficient algorithms for determining α_k employ interpolating polynomials between successive estimates, in order to obtain a maximum decrease with a minimum number of functional evaluations [3].

D. Practical Newton Line Search Methods

In computational electromagnetics, an underrelaxation factor α_k is often applied to the Newton step, yielding a so-called *underrelaxed Newton–Raphson method* [4]–[6]. Next to the numerical method itself, there are however several user-dependent factors influencing the convergence of the Newton–Raphson method, like the geometric description of the model and the mathematical representation of the data [7].

The Newton direction (8) is only guaranteed to be a descent direction if the Hessian is positive definite. If the Hessian is not positive definite, the Newton direction may not be defined or may not satisfy the descent condition (6). Practical Newton line search methods can be designed to ensure the descent condition without losing the second-order information in the Hessian (e.g., truncated Newton line search, modified Newton line search) [2].

IV. Trust Region Methods

A. Trust Region Step

In contrast to line search methods, trust region methods compute the search direction and the underrelaxation factor simultaneously. Trust region methods define a region around the iterate $\mathbf{A}_k$ within which the quadratic model (5) is assumed to be an adequate representation of F. The quadratic model may be constructed from the exact Hessian or from a positive definite approximation of it. In its simplest form, the *trust region step* is the step $\mathbf{s}_k$ which minimizes the quadratic model in a hypersphere of radius Δ_k centered at $\mathbf{A}_k$(Fig. 1)

$$\mathbf{s}_k \leftarrow \mathbf{s}: \quad \min_{\|\mathbf{s}\| \leq \Delta_k} \left\{F_k^{\mathrm{qm}}(\mathbf{s})\right\}. \tag{13}$$

Fig. 1 illustrates that, with this approach, the decrease of the functional may be larger than is possible along the Newton direction. Obviously, this depends on the trust region radius Δ_k. Methods for solving (13), also known as the *trust region subproblem*, are discussed later.

B. Controlling the Trust Region Radius

If the trust region radius is sufficiently large, it encloses the minimum of the quadratic model (5). Hence, the computed step equals the Newton step (8) or the quasi-Newton step (9). On the other hand, for small Δ_k, the trust region step can only realize

Given $\mathbf{A}_0$, $\Delta_{\max} > 0$, $\Delta_0 \in [0, \Delta_{\max}]$, $0 < \eta_1 < \eta_2 < 1$
For $k = 0, 1, 2, \ldots$

- Solve $\min\{F_k^{\mathrm{qm}}(\mathbf{s}_k)\}$ s.t. $\|\mathbf{s}_k\| \leq \Delta_k$
- Compute ρ_k
- Update Δ_k:

  if $\rho_k < \eta_1$
  $\longrightarrow$ decrease Δ_k e.g. $\Delta_k \leftarrow \|\mathbf{s}_k\|/4$
  else if $\rho_k > \eta_2$ and $\|\mathbf{s}_k\| = \Delta_k$
  $\longrightarrow$ increase Δ_k e.g. $\Delta_k \leftarrow 2\Delta_k$
  else if $\rho_k > \eta_2$ and $\|\mathbf{s}_k\| < \Delta_k$
  $\longrightarrow$ decrease Δ_k e.g. $\Delta_k \leftarrow \sqrt{\Delta_k \|\mathbf{s}_k\|}$
  else
  $\longrightarrow$ keep Δ_k

- Update $\mathbf{A}_k$:

  if $\rho_k > \eta_1$
  $\longrightarrow$ $\mathbf{A}_k \leftarrow \mathbf{A}_k + \mathbf{s}_k$

Fig. 2. Basic trust region algorithm.

a limited progress toward the optimal solution. In order to improve the performance of the algorithm, it is therefore necessary to adjust the trust region radius at each iteration. The adjustment is based on the ratio of the actual reduction of F to the reduction predicted by the quadratic model

$$\rho_k = \frac{F(\mathbf{A}_k) - F(\mathbf{A}_k + \mathbf{s}_k)}{F_k^{\mathrm{qm}}(0) - F_k^{\mathrm{qm}}(\mathbf{s}_k)}. \tag{14}$$

There are several strategies for updating the trust region radius. They all try to keep ρ_k close to unity while keeping Δ_k relatively large [2], [8]. If ρ_k is small or negative, the local quadratic model (5) does not accurately approximate the functional within the trust region. In that case, Δ_k is decreased and the previously computed step is rejected.

If ρ_k is close to 1, the local quadratic model seems to be a reliable representation of the functional. Two cases are then distinguished. If the computed trust region step lies on the trust region boundary, i.e., $\|\mathbf{s}_k\| = \Delta_k$, the trust region prevents a longer step to be computed and Δ_k should be increased for the next iteration. On the other hand, if $\|\mathbf{s}_k\| < \Delta_k$, the minimum of the quadratic model lies within the trust region. In that case, Δ_k should be pulled toward the Newton step (8) in order to avoid subsequent rejections in the next iterations.

If the similarity between the quadratic model and the functional is acceptable, e.g., if $\rho_k \in [\eta_1, \eta_2]$ with η_1 typically 0.25 and η_2 typically 0.75, the computed step is retained but the trust region radius is not modified. The basic trust region method described here is summarized in the algorithm of Fig. 2.

V. Trust Region Subproblem

A. Cauchy Step

The trust region algorithm has to solve a constrained minimization problem at the beginning of each iteration. It is denoted by the trust region subproblem. Similarly to line search methods, it is not required to compute the exact minimizer of this problem to ensure convergence. It can be proven [2] that the reduction of F_k^{qm} must be at least equal to the reduction obtained by the Cauchy step, which is the minimizer of the quadratic model along the steepest descent direction within the trust region around the current iterate (Fig. 1). The Cauchy step $\mathbf{s}_k^{\mathrm{C}}$ is inexpensive to compute as only matrix-vector multiplications are required [2].

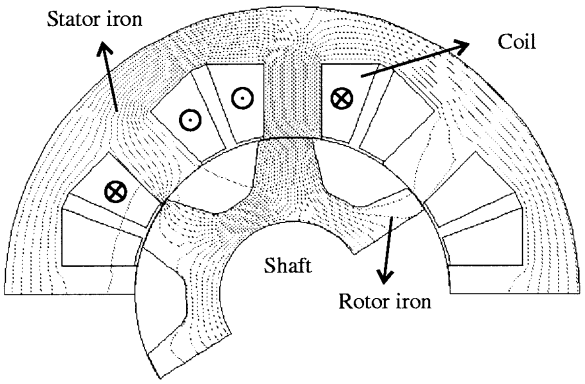

Fig. 3. Magnetostatic solution of an 8/6 switched reluctance motor.

B. Improving on the Cauchy Step

A trust region method using the Cauchy step for updating the iterates is slowly converging, as the Hessian does not play a role in the determination of the steepest descent direction. Therefore, it is imperative to improve on the Cauchy step. A very efficient method for achieving this is based on the conjugate gradient (CG) algorithm for solving linear positive definite systems of equations. It has been developed by Steihaug [9]. The Cauchy step is computed in the first linear CG iteration. All subsequent CG iterations result in a trust region step with a larger norm, for which the quadratic model has a smaller value. Next to the usual stopping criteria of the CG algorithm, two additional conditions are built in. The CG algorithm is terminated in the i^{th} linear iteration

- if a search direction $\mathbf{d}_i$ with a negative or zero curvature along the Hessian is encountered, i.e., $\mathbf{d}_i^{\mathrm{T}} \nabla^2 F(\mathbf{A}_k) \mathbf{d}_i \leq 0$, which may occur if the Hessian is not positive definite, or
- if the step $\mathbf{s}_i$ lies outside the trust region boundary, i.e., $\|\mathbf{s}_i\| \geq \Delta_k$.

VI. Implementation

For the analysis of the trust region method, the mathematical software libraries PETSc (Portable Extensible Toolkit for Scientific Computing) and TAO (Toolkit for Advanced Optimization) have been used [10], [11]. These packages are freeware and written in C/C++. They provide a rich environment for developing scientific applications in a single or multiprocessor environment.

VII. Application

The trust region method is applied for computing the magnetostatic solution of an 8/6 switched reluctance motor (Fig. 3). For this specific rotor position, two coils are simultaneously excited by the same current. The trust region method is compared with the unrelaxed and the relaxed Newton–Raphson method, in order to demonstrate the functionality of the trust region radius.

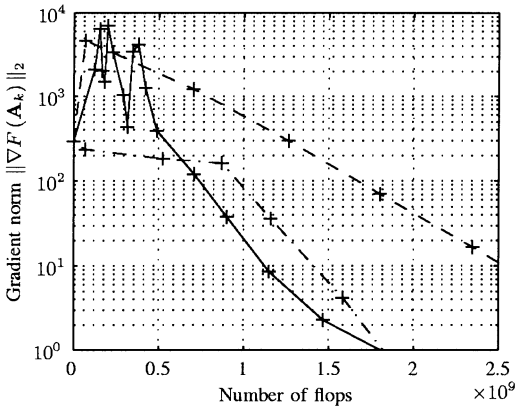

Fig. 4. The 2-norm of the gradient during the nonlinear iterations with the trust region method (solid), the unrelaxed Newton–Raphson method (dashed) and the relaxed Newton–Raphson method (dashed-dotted), for the switched reluctance motor in Fig. 3.

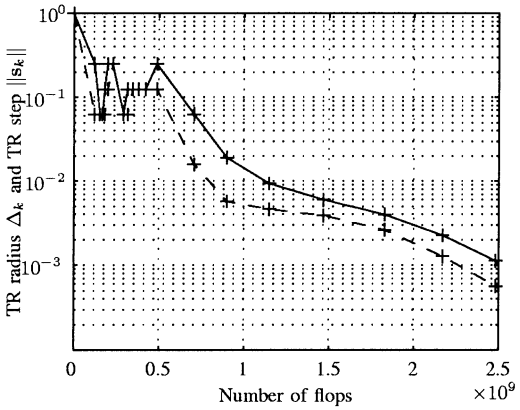

Fig. 5. The trust region radius (solid) and the 2-norm of the trust region step (dashed) during the nonlinear iterations, for the switched reluctance motor in Fig. 3.

The underrelaxation factor of the relaxed Newton–Raphson method is computed by successively dividing the Newton step by two, until the norm of the gradient is smaller than its predecessor. For the computation of the Newton step, the CG algorithm with ILU-preconditioning is applied. This allows for a fair comparison with the trust region method. The stopping criteria for the linear system solvers have been set to the same level.

For the three methods, Fig. 4 shows the Euclidean norm of the gradient as a function of the number of floating point operations. The trust region radius and the norm of the trust region step during the iterations are plotted in Fig. 5. Obviously, except for the first iteration, the unrelaxed Newton method converges smoothly. The relaxed Newton–Raphson method only requires underrelaxation in the first two iterations ($\alpha_1 = 0.5$ and $\alpha_2 = 0.25$) and converges smoothly. On the other hand, the trust region method initially performs many short nonlinear iterations, before a smooth convergence is observed. The initial behavior is a direct consequence of the additional stopping criteria in the algorithm for solving the trust region subproblem.

Fig. 5 illustrates the update algorithm of the trust region radius, as presented in Fig. 2. The trust region radius starts decreasing smoothly once the Newton step is located within the trust region bound. It can be shown that the asymptotic convergence of the presented Newton trust region method is quadratic [2].

VIII. Conclusion

The trust region method is an alternative to the relaxed Newton–Raphson method. The optimal trust region step is defined as a constrained minimization problem and is approximately computed by a variant of the conjugate gradient method. At the end of each nonlinear iteration, the trust region radius is automatically adjusted, depending on the accuracy of the quadratic model. The trust region radius initially controls the size of the step. Close to the solution, the computed step is located within the trust region boundary. The trust region method then behaves like the unrelaxed Newton method, featuring quadratic asymptotic convergence. The method is applied for the simulation of an 8/6 switched reluctance motor.

References

[1] P. P. Silvester and R. L. Ferrari, *Finite Elements for Electrical Engineers*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1996.

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

[3] J. J. Moré and D. J. Thuente, "Line search algorithms with guaranteed sufficient decrease," *ACM Trans. Math. Softw.*, vol. 20, pp. 286–307, 1994.

[4] T. Nakata, N. Takahashi, K. Fujiwara, N. Okamoto, and K. Muramatsu, "Improvements of convergence characteristics of Newton–Raphson method for nonlinear magnetic field analysis," *IEEE Trans. Magn.*, vol. 28, pp. 1048–1051, Mar. 1992.

[5] 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.

[6] 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.

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

[8] W. R. Cowell, "Sources and development of mathematical software," in *Prentice-Hall Series in Computational Mathematics*, 1st ed. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[9] T. Steihaug, "The conjugate gradient method and trust regions in large scale optimization," *SIAM J. Numer. Anal.*, vol. 20, pp. 626–637, 1983.

[10] S. Balay, W. D. Gropp, L. Curfman McInnes, and B. F. Smith, "PETSc users manual," Mathematics and Computer Science Division, Argonne National Lab., Tech. Rep. ANL-95/11-Revision 2.1.2, 2002.

[11] S. Benson, L. Curfmann McInnes, J. Moré, and J. Sarich, "TAO users manual," Mathematics and Computer Science Division, Argonne National Lab., Tech. Rep. ANL/MCS-TM-242, 2002.