SELECTED PROJECTION METHODS TO IMPROVE THE CONVERGENCE OF NON-LINEAR PROBLEMS

Ronny Mertens, Uwe Pahner, Ronnie Belmans and Kay Hameyer

Katholieke Universiteit Leuven, Dep. EE (ESAT), Div. ELEN,
Kardinaal Mercierlaan 94, B-3001 Leuven, BELGIUM E-mail: Ronny.Mertens@esat.kuleuven.ac.be

Abstract - **Short computation times required for the design and numerical optimisation of electromagnetic devices with the finite element method are obtained using an adaptive mesh refinement algorithm. Less time is spent on the initial mesh generation, while the time needed for refinement is negligible when special data structures are used. But an even more significant reduction in total computation time is achieved with the initialisation of the solution on the generated mesh. Less Newton steps are required to solve non-linear problems compared to a zero initial solution, while the time needed for the projection of the solution is far less than the time needed for a Newton step.**

Keywords: electromagnetic analysis, finite element methods, adaptive mesh refinement, projection methods, non-linear problems.

I. INTRODUCTION

Design and numerical optimisation of electromagnetic devices using the finite element method require fast computations. The overall solver speed is improved using an adaptive mesh refinement algorithm (Zienkiewicz, 1988). The edge based refinement, i.e. new nodes are inserted at the three edges of an element, is easily implemented by using two special data structures (Bank, 1982, Mertens, 1998). The node-to-element matrix contains all elements surrounding a node (Figure 1a). The neighbouring-element matrix contains the neighbouring elements along the three edges of an element (Figure 1b). Binary constraints are treated in such a way, that elements connected through binary or periodic constraints are neighbouring elements. An a-posteriori error estimator selects the elements to be refined. Afterwards the nodes are moved to the centre of the surrounding elements, resulting in a lower average element aspect ratio and thus improving the quality of the generated mesh. The time needed for refinement is insignificant. But an even more significant reduction in total computation time is achieved with the initialisation of the solution on the generated mesh.

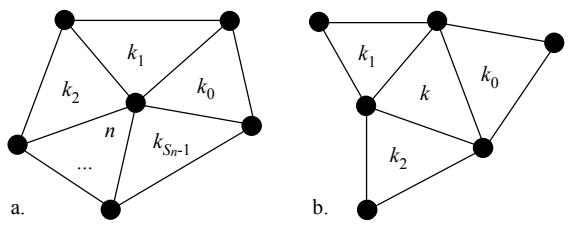

Fig. 1. Illustrating the node-to-element matrix and the neighbouring-element matrix

II. INITIALISATION OF THE SOLUTION

Poisson's equation in two dimensions leads to a linear system of equation of the form

$$\boldsymbol{K}\,\boldsymbol{A} = \boldsymbol{T} \tag{1}$$

where $\boldsymbol{K}$ is the element matrix, $\boldsymbol{A}$ the vector of unknown magnetic vector potentials and $\boldsymbol{T}$ the source vector. Non-linear problems are solved with an outer Newton iteration.

$$\boldsymbol{A}^{k+1} = \boldsymbol{A}^{k} + \delta\boldsymbol{A}^{k} \tag{2}$$

$$\boldsymbol{P}\,\delta\boldsymbol{A}^{k} = \boldsymbol{T} - \boldsymbol{K}\,\boldsymbol{A}^{k} \tag{3}$$

where $\delta\boldsymbol{A}$ is the vector of residues and $\boldsymbol{P}$ the Jacobian matrix. Both the element matrix $\boldsymbol{K}$ and the Jacobian matrix $\boldsymbol{P}$ depend on the approximation $\boldsymbol{A}^k$ of the exact solution. Therefore a start solution $\boldsymbol{A}^0$ is required to assemble the system of equations in the first Newton step. The same start solution can also be used as start solution for the iterative solver, e.g. a symmetric successive overrelaxation (SSOR) preconditioned conjugate gradient (CG) method.

A. Zero initial solution

The most simple start solution is the zero solution.

$$\boldsymbol{A}^0 = \boldsymbol{0} \tag{4}$$

The initial slope of the non-linear material characteristic is used for the reluctivity.

B. Average of the refined edge

Because new nodes are inserted along the edges of an element, i.e. an edge based refinement, the average of the solution on the two nodes i and j of the refined edge is taken as initial solution on the new node n.

$$A_n = \frac{A_i + A_j}{2} \tag{5}$$

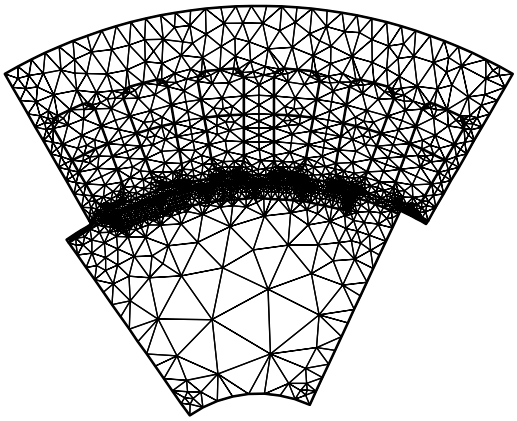

Fig. 2. Generated mesh after one refinement step (1790 nodes)

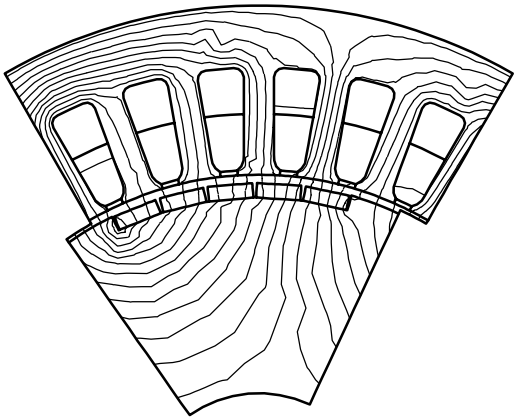

Fig. 3. Projection of the previous solution on the generated mesh by taking the average of the refined edge

This method corresponds to a full projection if the nodes are not moved to improve the quality of the generated mesh. Figure 2 shows the generated mesh after one refinement step for a 6-pole synchronous machine with inset permanent magnets (rated power 3kW). Only one pole is modelled and periodic boundary conditions are applied. Values of the nodal flux densities weighted by the energy in an element are used as an a posteriori error estimator. 7 refinement steps were calculated. Figure 3 shows the field plot of the projection of the previous solution on the generated mesh. Moving the nodes to improve the quality of the mesh, results in disturbances in the equipotential lines.

C. Moving & averaging

Each time a node is moved, the initial solution of the node is recalculated as average of the solution at the surrounding nodes (Figure 1a).

$$A_n = \frac{\sum_{i=0}^{S_n-1} A_i}{S_n} \tag{6}$$

Figure 4 shows the field plot of the projection of the previous solution on the generated mesh. No disturbances in the equipotential lines due to movement of nodes can be seen.

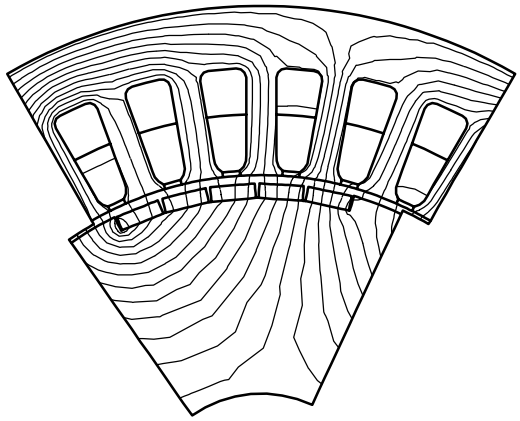

Fig. 4. Projection of the previous solution on the generated mesh by moving & averaging

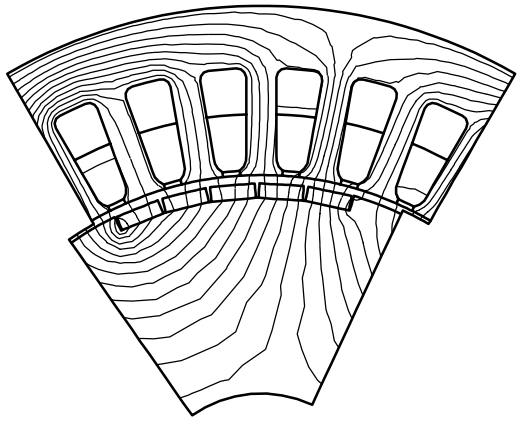

Fig. 5. Full projection of the previous solution on the generated mesh

D. Full projection

The most accurate method is a full projection of the previous solution on the generated mesh. A search over the elements of the previous mesh for each interior node, i.e. a node not lying on an outline, of the generated mesh is performed. The initial solution of each interior node is the interpolated value using the shape functions N_i of the finite element method.

$$A_n = \sum_{i=0}^{2} A_i \, N_i(x,y) \tag{7}$$

Due to the restoration of the original geometry during refinement, problems arise when a node lies on an outline arc. The new node falls inside the neighbouring region of the previous mesh or can even fall outside the model. The result is in both cases a jump in the magnetic vector potential along the outline. Because outline nodes are not moved after refinement, the initial solution of each outline node is calculated as the average of the solution at the two nodes of the refined outline edge (Eq. 5). Figure 5 shows the field plot of the full projection of the previous solution on the generated mesh.

III. SEARCH ALGORITHMS

A. Simple search

The simplest search method is an ordinary loop over the elements of the previous mesh for each interior node of the generated mesh and each time is tested if the node lies inside the element. This search algorithm is quadratically dependent on the number of nodes. Figure 6 shows the average number of elements searched per node for the all refinement steps. The overall computation time including 7 refinement steps, of the permanent magnet synchronous machine of Figure 2 is 80 seconds on a HP-C200 workstation.

B. Geometric search

An element k is taken as start element of the geometric search and checked if node n lies inside. The element is marked as being searched. The neighbouring element with its centre of gravity lying closest to node n is taken as the next element. A check is done and the element is marked as being searched. This is repeated until node n lies inside the element or the element was already being searched. In the latter case, the algorithm switches to a more guaranteed method as the simple search method. This switching happened only for 14 of the 7838 interior nodes during the last refinement step.

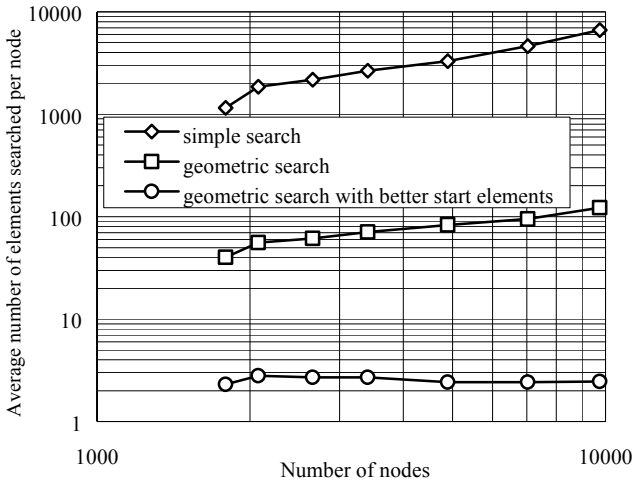

Fig. 6. Average number of elements searched per node for the different search algorithms

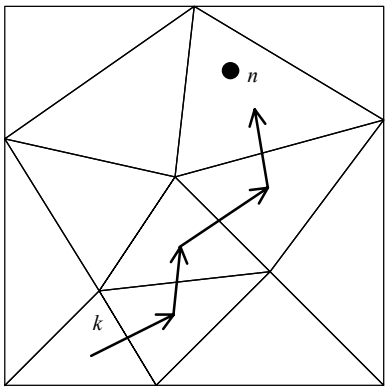

Fig. 7. Principle of the geometric search method

Figure 7 shows the principle of the geometric search method as walking through the mesh. This method uses the same data structure, i.e. the neighbouring-element matrix, as the refinement algorithm and follows immediately after it. The start element for all nodes should be determined in an easy and fast way. As this method can be seen as walking through the mesh, an obvious choice of a start element is located somewhere in the middle of the model. A loop over the nodes is made and a surrounding element (based on the node-to-element matrix) of the first node found lying inside a circle with radius R and with the centre of the extent as centre point is taken as start element (Figure 8). The radius R is a fraction of the extent of the mesh.

$$R = \frac{\sqrt{\left(x_{\max} - x_{\min}\right)^2 + \left(y_{\max} - y_{\min}\right)^2}}{10} \tag{8}$$

Figure 6 shows the average number of elements searched per node for all refinement steps. The slope in the figure indicates that the geometric search method is not a quadratic search algorithm. The non-linear problem was solved in 43 seconds.

C. Geometric search with better start elements

Instead of using one start element for all nodes, a start element per projected node can be used as long as it does not take too much time to find a better start element. When a solution is projected on a generated mesh by refinement, start elements can be determined during the refinement algorithm itself. Two categories of interior nodes of the generated mesh can be distinguished. Old nodes get as start element one of the surrounding elements in the previous mesh. Due to the creation of new elements and swapping of edges, the start element is not necessary a surrounding element of the old node in the generated mesh, but it is close. New nodes get as start element the element from which they are created by refinement. Figure 6 shows again the average number of elements searched per node for all refinement steps. In average 2 or 3 elements are searched per interior node, making the geometric search method linear and very fast for generated meshes due to refinement. 33 seconds were needed to solve the non-linear problem.

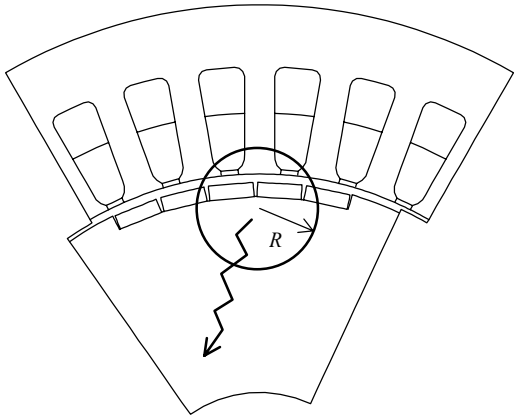

Fig. 8. Determining the start element for the geometric search method

IV. IMPROVED CONVERGENCE

A. Iterative solver

Figure 9 shows the typical convergence of an SSOR-CG iterative solver starting from a zero solution for the first Newton step of the last refinement step. When the projection of the previous solution on the generated mesh is used as start solution, the curve shifts down. As in most cases local values of the solution are required, e.g. the calculation of the torque, the stopping criterion of the iterative solver is low. Only a few iterative steps are gained when a linear problem is solved.

B. Outer Newton iteration

Figure 10 shows the number of Newton steps for the different projection methods. The Newton tolerance is set at 10^{-4}. The projection of the previous solution on the generated mesh by taking the average of the refined edge is rather unpredictable. Moving the nodes and averaging the solution results in less Newton steps, because the field solution is more smooth. Still one or two Newton steps can be gained by the full projection of the previous solution on the generated mesh. In average 2 or 3 Newton steps are needed after a few refinement steps to solve a non-linear problem in which saturation plays an important role, decreasing the overall computation time significantly.

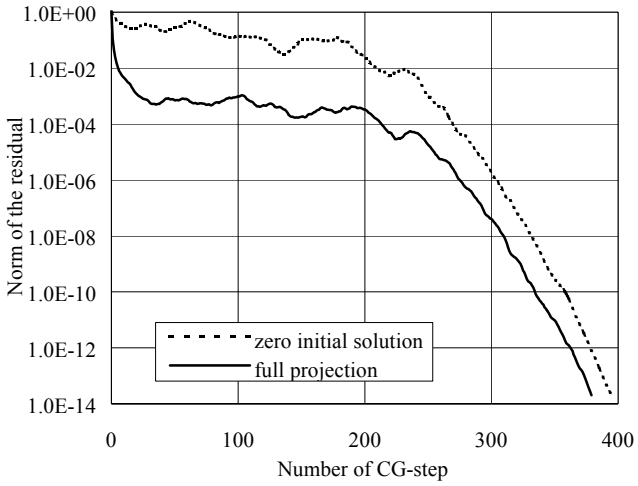

Fig. 9. Convergence of a SSOR-CG iterative solver

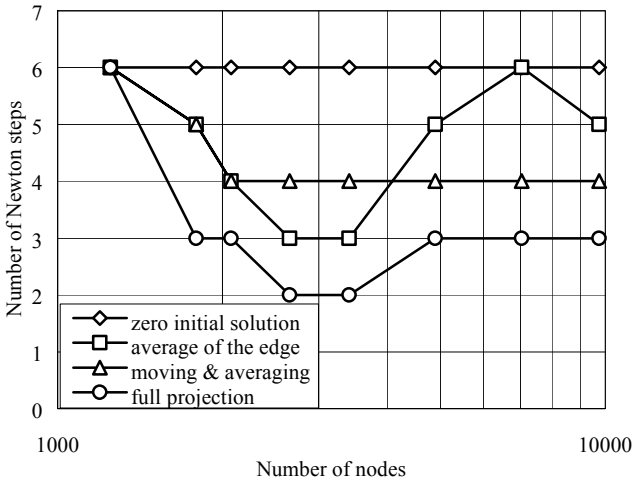

Fig. 10. Number of Newton steps for the different projection methods

V. CONCLUSION

A fast adaptive mesh refinement algorithm is a first step to improve the overall solver speed. A significant reduction in total computation time is further gained with the initialisation of the solution on the generated mesh. In the full projection of the previous solution, a geometric search method using the same special data structure as the refinement algorithm is implemented. When a solution is projected on a mesh generated by refinement, the start elements of the geometric search method are determined during the refinement and the method becomes linear. In average 2 or 3 elements are searched per interior node of the generated mesh, making the method very fast. In case of non-linear magnetostatic problems where each Newton step means solving a system of linear equations, the total computation time is significantly reduced because in average only 2 or 3 Newton steps are needed after a few refinement steps. Because refinement and projection takes only a few percent of the total computation time, more can be gained by using more effective iterative solvers for magnetostatic problems such as multigrid methods or domain decomposition (Mertens, 1997).

ACKNOWLEDGEMENT

The authors are grateful to the Belgian "Fonds voor Wetenschappelijk Onderzoek Vlaanderen" for its financial support of this work and the Belgian Ministry of Scientific Research for granting the IUAP No. P4/20 on Coupled Problems in Electromagnetic Systems. The Research Council of the K.U.Leuven supports the basic numerical research.

REFERENCES

O.C. Zienkiewicz, R.L. Taylor, *The Finite Element Method Volume 1: Basic Formulation and Linear Problems*, Fourth Edition, McGraw-Hill, 1988.

E. Bank, H. Sherman, A. Weiser, „On the regularity of local mesh refinement", *Proc. of 10th IMACS World Congress on System Simulation and Scientific Computation*, Montréal, Canada, August 8-13, 1982, pp. 61-64.

R. Mertens, U. Pahner, H. De Gersem, R. Belmans, K. Hameyer, „Improving the overall solver speed: a fast, reliable and simple adaptive mesh refinement scheme", *4th International Workshop on Electric and Magnetic Fields*, Marseille, France, May 12-15, 1998, pp. 385-390.

R. Mertens, H. De Gersem, R. Belmans, K. Hameyer, D. Lahaye, S.Vandewalle, D. Roose, „An Algebraic Multigrid Method for Solving Very Large Electromagnetic Systems", *IEEE Transactions on magnetics*, Vol.34, No.5, Sept. 1998, pp.3327-3330.