IMPROVING THE OVERALL SOLVER SPEED: A FAST, RELIABLE AND SIMPLE ADAPTIVE MESH REFINEMENT SCHEME

R. Mertens, U. Pahner, H. De Gersem, R. Belmans and K. Hameyer
Katholieke Universiteit Leuven, Dep. EE (ESAT), Div. ELEN
Kardinaal Mercierlaan 94, B-3001 Leuven, Belgium
tel. +32 16 32 10 20 - fax +32 16 32 19 85

Abstract

Design and numerical optimization of electromagnetic devices using the finite element method require fast computations. The overall solver speed is improved using an adaptive mesh refinement algorithm that is fast, reliable and simple. Starting from a classic edge based refinement algorithm, supplementary rules and moving the nodes are added to guarantee quality meshes. The generated high quality meshes with an average aspect ratio close to unity, improve the convergence rate and thus reduce the overall computation time. Less time is spent on the initial mesh generation, while the time needed for refinement is insignificant. Special data structures are used to implement the edge based refinement and to avoid quadratic search algorithms over nodes and elements.

INTRODUCTION

The efficiency of an adaptive mesh refinement algorithm is determined by the quality of the generated mesh. This quality is characterized by

- the size of the elements,
- an average element aspect ratio close to unity,
- a low worst element aspect ratio, and
- the ability to restore the original geometry.

The last three characteristics are determined by the mesh refinement algorithm itself, while an a posteriori error estimator selects the elements to be refined.

Adaptive mesh refinement based on an error estimator results in lower memory requirements and higher solver speed for a given accuracy [1]. As an example, fig. 1 shows the initial mesh and the field plot of a C-core inductor. The local error based on the magnetic flux density in a node, weighted with the energy in an element, is used as error estimator. The difference in slope of the global error e in fig. 2 indicates a better convergence rate and thus a higher overall solver speed when an error estimator is used.

$$e = \left| \frac{W - W_\infty}{W_\infty} \right| \tag{1}$$

with

energy W stored in the model, and
energy W_∞ stored in the model for an infinite or significantly higher number of nodes.

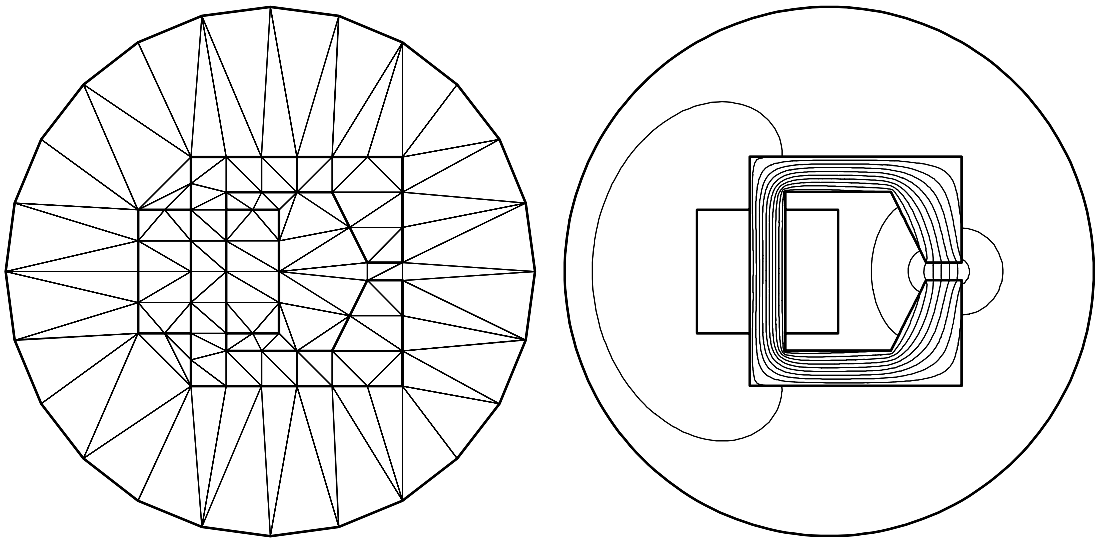

Fig. 1. Initial mesh and field plot of a C-core inductor.

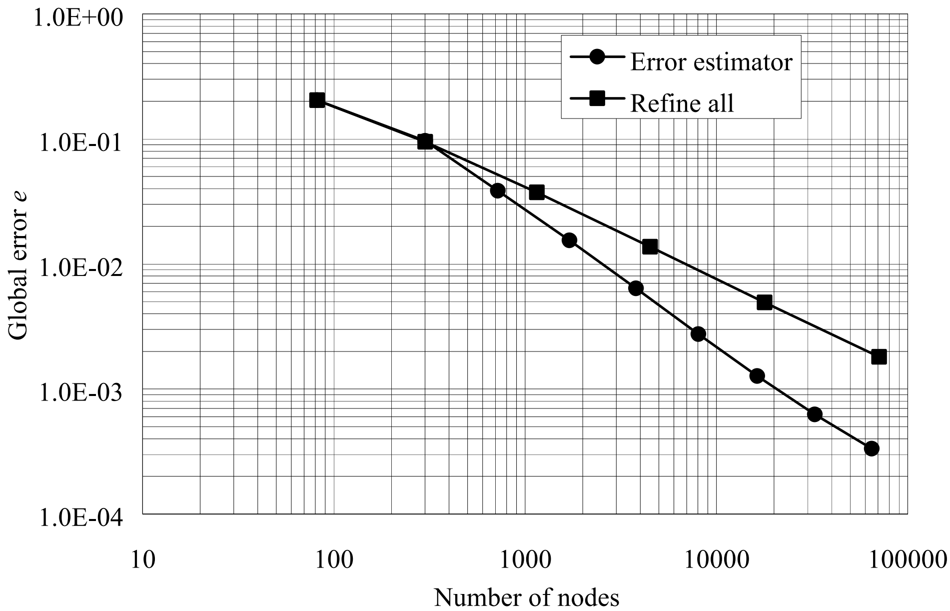

Fig. 2. Global error versus number of nodes: convergence rate.

ADAPTIVE MESH REFINEMENT SCHEME

Data Structures

Two special data structures are built to avoid quadratic search algorithms over nodes and elements [2]. The node-to-element matrix contains all elements surrounding a node (fig. 3a) and is built in a straightforward way. The neighbouring-element matrix contains the neighbouring elements along the three edges of an element (fig. 3b). As the maximum number of surrounding elements is almost a constant for quality meshes, and the rows of the node-to-element matrix are sorted, a very effective and fast search algorithm can be used to build the neighbouring-element matrix. Binary constraints are dealt with in such a way that elements connected through binary constraints are treated as neighbouring elements.

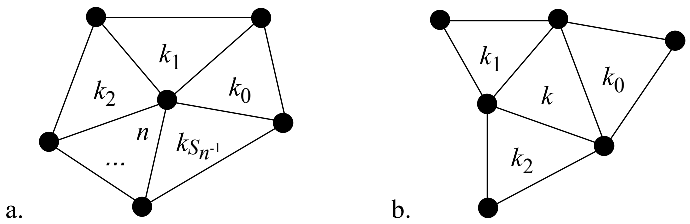

Fig. 3. Illustrating the node-to-element matrix and the neighbouring-element matrix.

Marking Edges for Refinement

The refinement of the elements is edge based, i.e. new nodes are inserted at the three edges of the elements. This method preserves the initial element aspect ratio compared to an element based refinement method. The marking of an edge is done by putting the new node number in the to-be-refined-edges matrix. At the same time, the appropriate edge of the neighbouring element is marked with the same node number. When two elements are connected through binary constraints, the next node number is used because a new node has to be inserted. Supplementary rules are added to improve the quality of the generated mesh. Thus the marking of the edges for refinement is done in four steps:

- not all edges of an element with a bad aspect ratio (e.g. > 5.0) are marked to improve the initial average element aspect ratio (fig. 4),
- all edges of a selected element are marked,
- all non-outline edges touching two outlines are marked (only in the first refinement step) (fig. 5), and
- all edges touching a selected outline edge are marked (fig. 6).

Fig. 7a shows the result of a classical edge based refinement algorithm after 3 refinement steps for the air gap of the C-core inductor of fig. 1. Fig. 7b shows the effect of the supplementary refinement rules, resulting in a better quality of the generated mesh. To improve the quality of the mesh further, the nodes are moved.

Fig. 4. Refinement of elements with an aspect ratio > 5.0.

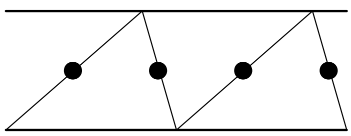

Fig. 5. Refinement of non-outline edges that touch two outlines.

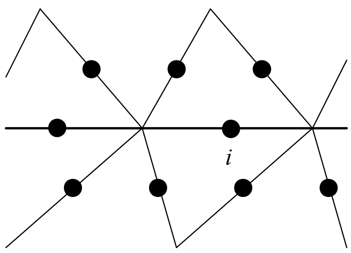

Fig. 6. Refinement of all edges that touch a to be refined outline edge *i*.

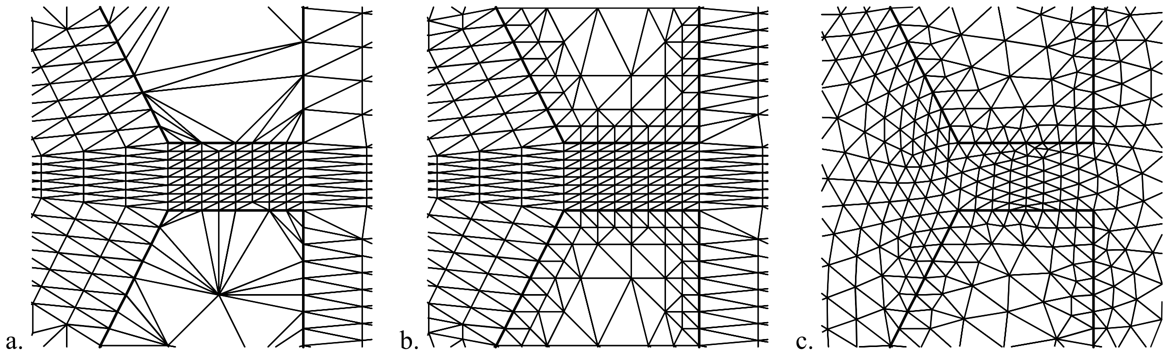

Fig. 7. Edge based refinement of the air gap of a C-core inductor: starting from a classic edge based refinement algorithm (a), supplementary rules are added (b) and finally the nodes are moved (c).

Moving Nodes

Each node is moved to the centre of gravity of the surrounding elements and all touching edges are checked if swapping can improve the element aspect ratio [3]. This is done for all nodes and repeated until the number of swapped edges is a fraction (e.g. 10 %) of the initial number of swapped edges. Moving the nodes results in a lower average element aspect ratio (fig. 7c).

PERFORMANCE OF THE REFINEMENT SCHEME

Average Element Aspect Ratio

The aspect ratio of a triangular element is calculated as the ratio of radius of the circumscribed circle to twice the radius of the inscribed circle. An equilateral triangle has an aspect ratio of 1.0. Fig. 8 shows the evolution of the average element aspect ratio for the three different refinement schemes: starting from a classic edge based refinement algorithm, supplementary rules are added and finally the nodes are moved. The complete refinement scheme generates quality meshes with an average element aspect ratio less than 1.1 in a few refinement steps.

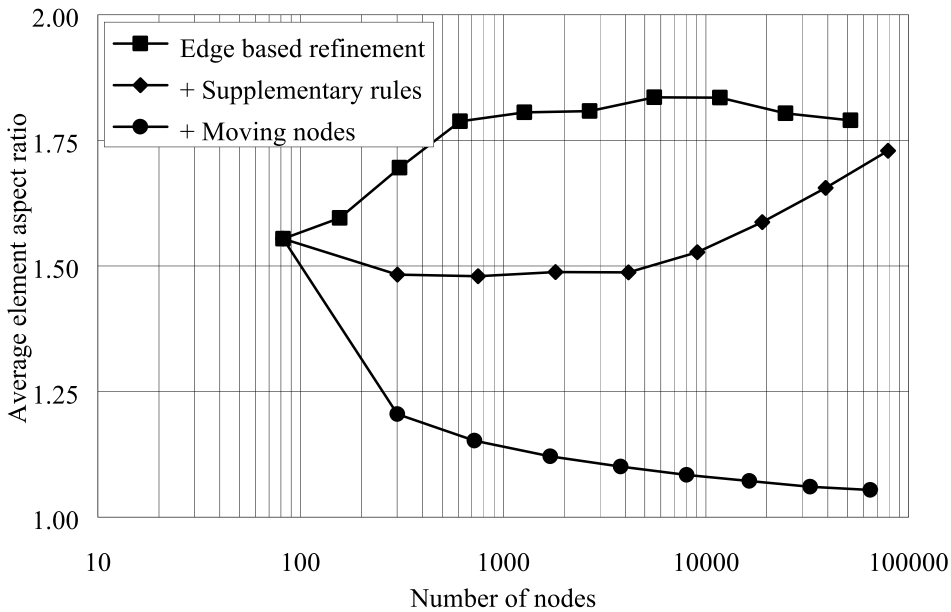

Fig. 8. Evolution of the average element aspect ratio.

Improved Convergence Rate

Fig. 9 shows the convergence rate for the three different refinement schemes. Starting from an initial discretisation, i.e. only nodes on the outline of the model, a lot more new nodes are inserted due to the supplementary rules. This is almost immediately compensated by moving the nodes. The difference in final slope of the global error indicates a better convergence rate.

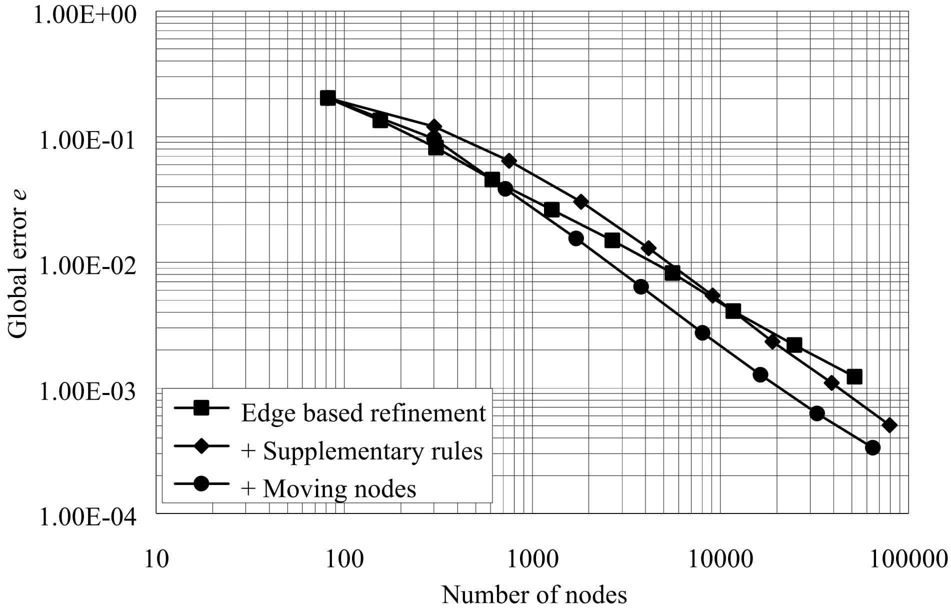

Fig. 9. Global error versus number of nodes: convergence rate.

Time Needed for Refinement

Fig. 10 shows the total computation time of the C-core inductor split in error estimation, refinement and solving the system of equations with a SSORCG iterative solver (matrix assembly included). Refinement takes only 2.8 % (104 s) of the total computation time (3753 s). Saving the final solution to a local hard disk needs an extra 27 s. The calculations were performed on a HP C200 workstation.

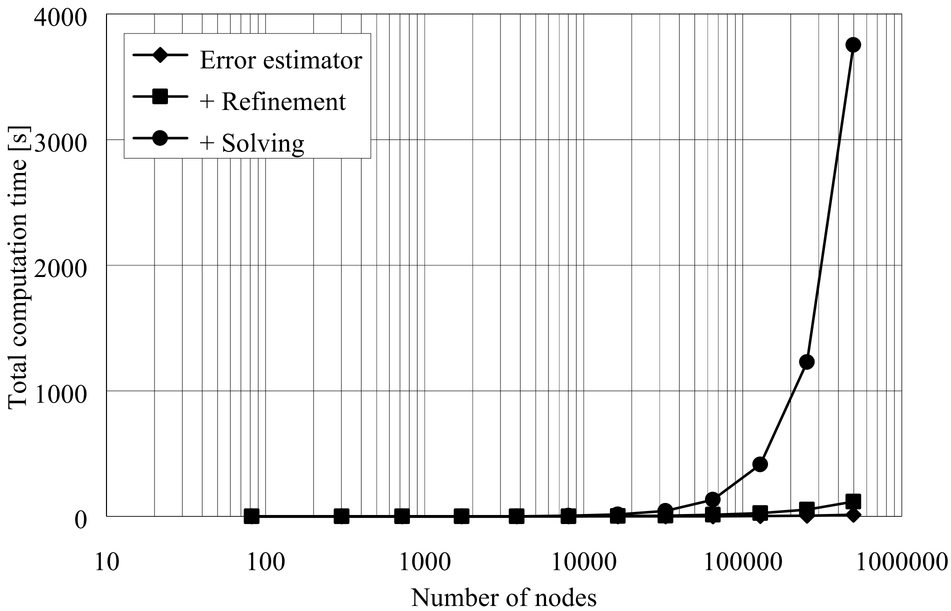

Fig. 10. Time spent on error estimation, refinement and solving the system of equations.

CONCLUSION

The adaptive mesh refinement scheme is fast, reliable and simple. The refinement scheme with its supplementary rules results in a quality mesh with an average element aspect ratio close to unity (a typical value of 1.1). This results in an improved overall solution speed, while the time needed for refinement is insignificant (fig. 10). Special data structures are built and very efficient quasi linear search algorithms replace the quadratic search algorithms over nodes and elements. In case of more realistic models, i.e. electromagnetic devices in which saturation plays an important role, the time needed for the solution proces becomes more significant because several Newton steps have to be performed. Because refinement takes only a few percent of the total solution time, a lot more can be gained by using more effective iterative solvers for magnetostatic problems such as multigrid methods or domain decomposition [4].

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

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

[2] E. Bank, H. Sherman and A. Weiser, "On the regularity of local mesh refinement", in *10th IMACS World Congress on System Simulation and Scientific Computation*, August 8-13, 1982, Montréal, Canada.

[3] S.W. Sloan, “A Fast Algorithm for constructing Delaunay triangulations in the plane”, *Software for Engineering Problems III, Progress in Engineering Vol. 11*, Computational Mechanics Publications, 1993.

[4] R. Mertens, H. De Gersem, R. Belmans, K. Hameyer, D. Lahaye, S.Vandewalle and D. Roose, “An Algebraic Multigrid Method for Solving Very Large Electromagnetic Systems“, in *Proceedings of the XIth Conference on the Computation of Electromagnetic Fields* (COMPUMAG), Rio de Janeiro, Brazil, Nov. 2-6, pp. 531-532.