

A Parametric Finite Element Environment tuned for Numerical Optimization

Uwe Pahner, Ronny Mertens, Herbert De Gersem, Kay Hameyer and Ronnie Belmans

KATHOLIEKE UNIVERSITEIT LEUVEN, DEPT. E.E. (ESAT), DIV. ELEN

Kardinaal Mercierlaan 94, B-3001 Leuven, BELGIUM

Abstract - Nowadays, numerical optimization in combination with finite element (FE) analysis plays an important role in the design of electromagnetic devices. To apply any kind of optimization algorithm, a parametric description of the FE problem is required and the optimization task must be formulated. Most optimization tasks described in literature, feature either special developed algorithms for the particular optimization task, or extensions to standard finite element packages. Here a 2D parametric FE environment is presented, which is designed to be best suited for numerical optimization while maintaining its general applicability. Attention is paid to the symbolic description of the model, minimized computation time and the user friendly definition of the optimization task.

INTRODUCTION

The finite element analysis is widely accepted for its general application range regarding geometry and problem type (electromagnetic, thermal, motion, coupled problems). This makes it a desirable tool for the optimization of electromagnetic devices, regardless of its computational expense [1]. An optimization problem is formulated by defining an objective (quality) function with a number of design parameters as the variables. The purpose of an optimization is to find the best possible solution to a given problem by the simultaneous variation of the design variables. This requires many objective function evaluations. Two main streams in the combination of FE analysis with different optimization algorithms can be pointed out:

- the development of computer codes that are designed to solve a particular optimization problem as efficient as possible (accepting the loss of the general applicability range of the final code) [2],
- the development of add-on tools to standard finite element packages that are originally not designed to perform repetitive analyses [3].

This paper focuses on the development of a 2D FE code, which is user-friendly, optimized for repetitive analyses while still offering a general application range (Fig. 1).

SYMBOLIC DESCRIPTION OF FINITE ELEMENT MODELS

To allow a repetitive analysis of a FE problem, a thorough symbolic description is a prerequisite. Geometry, excitations as well as material data are defined symbolically. The syntax of these definitions is *Matlab*-like, and can be entered by the user, or using macro-commands. Thorough in terms of FE analysis includes the possible symbolic description of

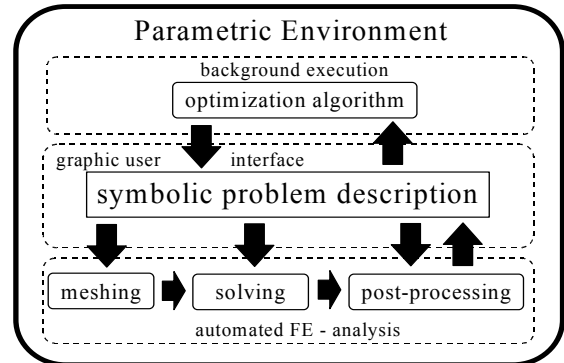


Fig. 1: All analysis and optimization steps are defined by the symbolic description of the problem.

boundary conditions, external electric circuits, initial discretizations on geometric boundaries, parameter consistency checks, solver settings and post-processor computations. A special feature is the definition of analysis procedures, that define the sequence of steps towards the desired result in case of a non-standard analysis. Third party programs can be linked into the analysis. Weak coupled problems, such as time-harmonic/thermal can be defined. The symbolic problem description is stored in ASCII format. This allows access to all information if it is intended to link other routines or codes into the analysis process (e.g. specialized post-processor tools). The problem file does not contain information about the FE nodes, elements etc.. This ensures low storage requirements. The mesh generation and the solution process are fully automated.

ACCELERATION OF FINITE ELEMENT ANALYSIS

Accuracy and convergence rate of the FE analysis are strongly depending on the discretization of the model and the appropriate solution algorithm of the system of equations.

Initial Discretization and Mesh Refinement

It is often impossible to determine a good discretization of the model before starting the optimization, as the optimized shape of the device can be totally different. A fast and reliable mesh refinement, in combination with various *a posteriori* error estimators is implemented here. First a minimum discretization of the symbolic description of the model is constructed. It is based on a very fast constrained Delaunay algorithm [4]. A sequence of refinement steps is performed until the desired accuracy is reached. The error estimators based on interpolation theory can be applied for different

regions in the model. The intersection of the elements is followed by a quality enhancement of the discretization. Nodes are moved towards the center of gravity of the surrounding nodes and a local, Delaunay based edge swapping algorithm is applied. This assures an average aspect ratios of the elements close to 1 (typical < 1.2) in the final discretization. Particular attention is paid to fast refinement algorithms. Error estimation and refinement requires less than 10% of the overall computation time.

Solution of the System of Equations

The choice of the appropriate solution algorithm depends on the properties of the coefficient matrix, which in term depends on the problem type. Several pre-conditioners can be combined with either CG (Conjugate Gradient) or BiCG (Bi-directional Conjugate Gradient): SOR (Successive Over-Relaxation), SSOR (Symmetric Successive Over-Relaxation), IC (Incomplete Cholesky Decomposition) and AMG (Algebraic Multigrid). The Algebraic Multigrid preconditioner is especially well suited for large systems of equations, as it reduces the computation time significantly (up to 11 times) compared to classical pre-conditioned CG methods [5]. After the mesh refinement, all solvers start with an interpolated solution at the newly imposed nodes. In a non-linear analysis, the number of Newton steps can be reduced down to 25% of the steps with the initial mesh.

DEFINITION AND EXECUTION OF OPTIMIZATION TASKS

The pre-processor of the FE package provides all tools for the set-up of an optimization task. Most optimization problems are constrained, often feature multiple objectives and it is desired to find the global optimum. Stochastic algorithms (Evolution Strategy, Genetic Algorithms, Simulated Annealing) are chosen, due to their simplicity in the preparation of this type of problem. The optimization algorithms are implemented in an external optimizer. The three reasons for this separation are:

- Non-FE optimizations are performable.
- The execution of the optimization task defined by the pre-processor can be parallelized.
- Additional optimization algorithms can be linked to the package without changing the main code.

Based on the symbolic description of the model, consistency checks are formulated by the user, using the same syntax. The violation of one of these logical statements causes the non-acceptance of the set of design variables. While the user defines design variables in their physical units, the optimizer requires normalized variables. The pre-processor allows to define a normalization function for each design variable. During the optimization, the design objectives are calculated for each intermediate design (local field values, forces, inductances, volumes etc.). They have to be combined

to return a scalar valued quality, describing the acceptance of the design in terms of the objectives. As every optimization task might involve non-standard calculations, especially during the post-processing, a general data transfer algorithm is provided. It allows the extraction of data from external ASCII files (e.g. LOG-files) into user defined variables inside the FE-package. It is now possible to formulate a single valued quality function (e.g. weighted sum) using the same syntax as in the definition of the model. The execution of the optimization is performed either from within a graphical interface, or more efficient in background, involving no graphics. A stand-alone program allows to monitor the progress of the optimization at run-time. The optimization can safely be stopped and restarted at any time. This feature is required, as settings of the optimization algorithm are to be adjusted in the starting phase of an optimization or hardware (network) maintenance requires the interruption of the process.

CONCLUSION

A powerful 2D FE-package is presented which is tuned for numerical optimization. Emphasis is put on the combination of various measures to accelerate the FE-analysis, while remaining its applicability to different types of field and optimization problems. A thorough symbolic description of the problem serves as the basis for the optimization of the device.

ACKNOWLEDGMENT

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] K. Hameyer, "Numerical Optimization of Finite Element Models in Electromagnetics with Global Evolution Strategy", Adaptive Computing in Engineering Control, ed. by I.C.Parmee, pp.61-66, Plymouth, 1994
- [2] I.H. Park, H.B. Lee, I.G. Kwak, S.Y. Hahn, "Design Sensitivity Analysis for Steady State Eddy Current Problems by Continuum Approach" *IEEE Transactions on Magnetics*, vol. 30, no. 5, pp. 3411-3414, September 1994
- [3] B. Trowbridge, "The Role of Optimisation Techniques used for Magnet Design", Proceedings of the 3rd International Workshop on Electric and Magnetic Fields, pp. 15-23, Liege, Belgium, May 6-9 1996.
- [4] J.R. Shewchuk, "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator", First Workshop on Applied Computational Geometry, pp. 124-133, Philadelphia, USA, May 1996.
- [5] R. Barrett et al., Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM Philadelphia, 1994