Mixing Techniques to Compute Derivatives of semi-numerical models: Application to Magnetic Nano Switch Optimization

P. Pham-Quang[1,2], B. Delinchant[1], C. Ilie[3],
E. Slusanschi[3], J.L. Coulomb[1], and B. du Peloux[2]
[1]Grenoble Electrical Engineering Laboratory, G2ELab, F-38402 St Martin d'Hères Cedex, France
[2]Cedrat SA, 15, Chemin de Malacher, F-38246 Meylan Cedex, France
[3]Computer Science and Engineering, University Politehnica of Bucharest, 060042 Bucharest, Romania
Email: phuong.pham-quang@g2elab.grenoble-inp.fr

Abstract — This paper is about derivatives techniques and their composition for semi-numerical models. Techniques such as symbolic derivation and automatic differentiation are addressed. All techniques are illustrated for the gradient based optimization of a magnetic nano switch.

I. Introduction

Sizing by optimization is nowadays of major interest of the design process of electromagnetic device. Our goal is to use optimization algorithms in conjunction with semi-analytical model in order to solve the sizing problem. Among the deterministic algorithms, the constrained gradient based optimization using Sequential Quadratic Programming (SQP) algorithms is very efficient. Such algorithms require accurate values of the objective function and constraint derivatives. Unfortunately, it is often difficult to obtain the symbolic expression of the derivatives of the sizing model. But several techniques as Automatic Differentiation (AD) can be used to compute these derivatives.

The paper highlights a generic framework of model composition using different derivation techniques. This approach is shown through the example of a magnetic nano switch optimization.

II. Derivation Techniques

The finite difference approximation leads to an approximation suffering from both truncation and cancellation errors. It is very easy to set-up, but it is very difficult to settle the adjustment parameter.

The derivatives can be computed symbolically. This method is reliable and the faster that can be found, not only with symbolic expression but also with functions resulting from numerical integration or from implicit solver (thanks to implicit theorem). The main drawback of this method is the impossibility to differentiate the computing code (with conditional instructions, loops, etc.).

Automatic differentiation (AD) exploits the natural process of source code compilation and makes use of the intermediate representation of implemented functions in different programming languages. The derivatives of each elementary operation can be obtained in a straightforward way and can be combined according to the chain rule of differential calculus in order to obtain the derivative of the initial function. Thus, depending on the chosen accumulation strategy, functions of arbitrary complexity can be differentiated in two ways, namely forward and reverse. In the forward mode one propagates derivatives of intermediate variables with respect to the independent variables, while in reverse, derivatives of the function with respect to intermediate variables – also called adjoints – are propagated.

Typically, AD can be implemented using either the operator overloading (ADOL-C [4], CppAD, etc) or the source transformation technique (ADIFOR, ADiJaC [6], etc). In operator overloading one overloads the operators which are applied on new variable types, with the routine call performing the actual derivative computation. The source transformation approach examines the source code of the original function and generates new code that computes the desired derivatives in the same time that the original function.

III. Sensitivity Propagation of Compound Model

As studied in [2] , and as shown in Fig. 1 and Fig. 2, two kinds of sensitivity propagation mechanism can be considered to build global sensitivities.

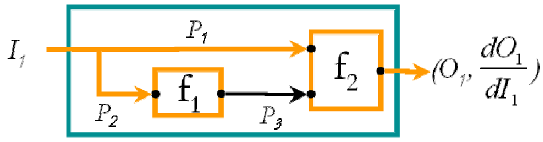

Fig. 1. A compound model based on partial derivative propagation

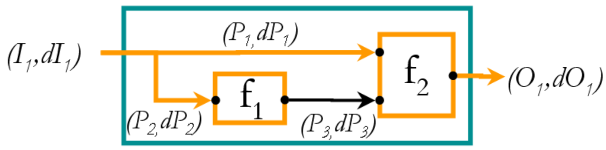

Fig. 2. A compound model based on differential propagation

Each method owns its advantages and drawbacks. The choice of partial derivative propagation has been made with interface standard in CADES framework (developed in Grenoble laboratory) to ensure interoperability between models from different derivation techniques.

IV. Derivatives Of Magnetic Nano Switch Model

The semi-numerical model of magnetic nano switch was presented in [3]. It involves a coupled magneto-mechanical deformation of a cantilever beam, and allows evaluating the contact length and the contact force. The following part details how the derivatives of each model blocks are computed.

A. *Magnetic model*

For specific configuration of MEMS/NEMS, the Coulombian approach is used for the modeling of magnet, leading to a full analytical magnetic field model. Forces and magnetic torques applied on the beam (in magnetic parts) are computed by adaptive numerical integrations. The derivatives of this magnetic model can be computed symbolically with accuracy and rapidity [5].

B. *Mechanical deformation model*

The model used to compute deformation in the presence of contact of a cantilever beam has inputs and outputs as shown in Fig. 3

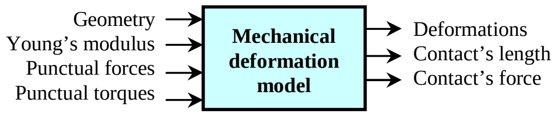

Fig. 3. Inputs and outputs of mechanical deformation model

ADiJaC [6] was used in order to manage the derivation of this complex algorithm. ADiJaC is a source transformation AD tool that implements both the forward and reverse modes on Java codes. To date it is the first useable AD tool for the Java programming language. Differentiating the NEMS model with ADiJaC was improved to deal with special cases of matrix operations, nested method calls, special functions (e.g. "length"), and special array initializations – all intra-procedural transformations. During this work, ADiJaC's inter-procedural analysis was also enhanced, so as to allow the tool to differentiate functions returning arrays, and receiving any number of vector and scalar parameters.

C. *Magneto-Mechanic coupling*

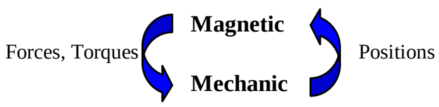

Fig. 4. Sequential magnetic-mechanical coupling

The sequential magneto-mechanical coupling described in Fig. 4. has been solved by an implicit solver with Gauss method. Its derivatives are computed symbolically thanks to a classical implicit theorem.

D. *Composition in the CADES framework*

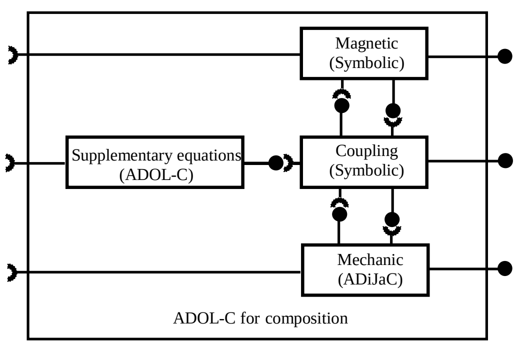

Fig. 5. Sensitivity propagation of magnetic nano switch in CADES

All these models are given with their own Jacobian using different techniques (symbolic or AD). CADES framework defines a standard to make the composition possible. Each model implementing this standard can be integrated into the framework. Then the overall is treated automatically by another AD tool (ADOL-C) to build the global model (Fig. 5).

V. Optimization Results

Working principle of magnetic nano switch was presented in [3]. Geometry and parameters to be optimized are given in Fig. 6. The optimization aims to determine dimensions of both fixed and mobile magnets and their positions on the beam. The objective is to minimize the volume of magnets, while respecting constraints such as length of contact and contact force, in order to ensure the quality of contact or the contact resistances are below a desired value.

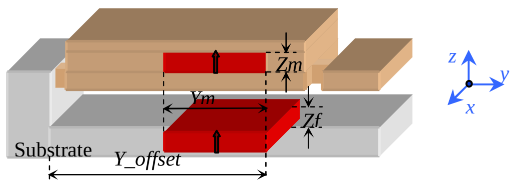

Fig. 6. Geometry and parameters to be optimized

A genetic algorithm (without gradients) and the SQP algorithm (using gradients) were used. Both gave the same results but the first algorithm needed 3 hours computation time while the second only 92 seconds.

VI. Conclusions

Deterministic optimization using gradient based algorithms is very efficient for sizing problems with constraints. However, such algorithms meet a major lack of usage because of the difficulty to obtain exactly and rapidly the derivatives values. This paper highlights the composition of different derivation techniques in a compound model through the example of magnetic nano switch.

The full paper will detail the propagation mechanism of derivatives. Each derivation techniques will be detailed, highlighting advantages and drawbacks. More details will be given on the optimization procedure and results.

VII. References

[1] J. Sobieszczanski-Sobieski, "Sensitivity of complex, internally coupled systems," AIAA J., vol. 28, no. 1, pp. 153–160, Jan. 1990.

[2] B. Delinchant, F. Wurtz and E. Atienza, "Reducing sensitivity analysis times-cost of compound model", *IEEE Transactions on Magn.*, vol. 40 no. 2, pp. 1216-1219, March 2004.

[3] P. Pham-Quang, B. Delinchant, J.L. Coulomb and B. du Peloux, "Semi-analytical magneto-mechanic coupling with contact analysis for MEMS/NEMS", to be published *IEEE Transactions on Magn.*,

[4] P. Enciu, F. Wurtz, L. Gerbaud, B. Delinchant "Automatic differentiation for electromagnetic models used in optimization", COMPEL; Vol 28, Issue: 5, pp 1313 - 1326, 2009, ISSN: 0332-1649

[5] B. Delinchant, et al. "Gradient based optimization of semi-numerical models with symbolic sensitivity", IJAEM, vol.30, pp.189-200,2009.

[6] E. Slusanschi, Algorithmic differentiation of Java programs, Ph.D. thesis, RWTH Aachen University, 2008.