11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
July 20 - 25, 2014, Barcelona, Spain

A DYNAMIC APPROACH FOR AUTOMATING FINITE ELEMENT CODE DEVELOPEMENT

D. Eyheramendy[1] and R. Saad[2]

[1] Centrale Marseille, LMA, CNRS UPR 7051, Aix Marseille Université, 38 rue F. Joliot-Curie, 13451 Marseille Cedex 20, dominique.eyheramendy@centrale-marseille.fr
[2] Université Saint-Esprit de Kaslik, Jounieh, Lebanon, roysaad@usek.edu.lb

Key Words: *Symbolic computations, dynamic binding, Finite Elements, Java.*

Since the origin in computational mechanics, different kinds of approaches have been developed in the design of simulation tools. The most important one remains the hand implementation of mathematical forms and algorithms derived from the physical problems lying on a traditional programming language. Since the middle of the 80's, with the increasing complexity of the problems to be solved, the object-oriented programming has gained more and more attention. Since the first works published on the subject (see e.g. Rehak [1], Miller [2], Zimmermann [3] and references therein), almost every sector of computational mechanics has been addressed, and the approach is now widely adopted. In the same time, the use of algebraic manipulation software has always been a focus for finite element developments. Among the precursors are Luft [4] and Noor [5], in which is described a methodology to automatically generate finite element matrices. Later on, many solutions were presented for solving different kind of finite element problems including some symbolic computations, e.g. in magnetic Yagawa [6]. Nonlinear finite element problems have also been considered such as in Eyheramendy-Zimmermann [7] (see also references therein). In the same time, Korelc [8] has presented an original and complete set of tools in MATHEMATICA providing a global framework for deriving FE models. The first tool aims at deriving symbolic derivation, and the second aims at automatically interfacing the finite element code and the symbolic environment. Automatic differentiations techniques are used to derive complex nonlinear problems. Similarly, Logg et al. [9] have presented the project FEniCS with the explicit goal of developing software for the automation of computational mathematical modeling, including an automation of the finite element method. At last, Saad and Eyheramendy (see [10][11][12][13]) have extended the approach proposed in [7] to full tensor analysis of the discretization process of continuum equation. The approach is proposed for finite elements but is general enough to be extended to alternative discretization schemes. The mathematical formalism is based on tensor algebra in which the discretization process of a variational formulation is introduced. The generic character of the approach is preserved through the object-oriented approach in Java. In most of these works, code is automatically produced and then compiled. The whole process may be considered as a static process, which necessitates a compilation phase, even if today compiler can provide dynamic binding capabilities. In the present work, we explore a track in which the symbolic framework is closely integrated into the classical numerical one, providing a kind of dynamic binding of the newly symbolically created formulations.

REFERENCES

[1] Rehak D.R. and Baugh Jr. J.W., Alternative Programming Techniques for Finite Element Programming Development, *Proceedings IABSE Colloquium on Expert Systems in Civil Engineerings*, Bergamo, Italy. IABSE, (1989).

[2] G.R Miller., A LISP-Based Object-Oriented approach to structural analysis, Engr. with Comp., vol. 4 (1988) pp. 197-203.

[3] Th. Zimmermann, Y. Dubois-Pèlerin and P. Bomme, Object-oriented finite element programming : I. Governing principles, *Comput. Methods Appl. Mech. Engrg.*, vol. 98 (1992) pp. 291-303.

[4] R.W. Luft, J.M. Roesset and J.J. Connor, Automatic generation of finite element matrices, *Struct. Div., Proceedings of ASCE*, January 1971, (1971) pp. 349-361.

[5] Computerized symbolic manipulation in nonlinear finite element analysis, A.K. Noor, C.M. Andersen, *Computers & Structures* vol. 13 (1981) pp. 379-403.

[6] G. Yagawa, G.-W. Ye and S. Yoshimura, A numerical integration scheme for finite element method based on symbolic manipulation, *Internat. J. Numer. Methods Engrg.*, vol. 29 (1990) pp. 1539-1549.

[7] D. Eyheramendy, Th. Zimmermann, The Object-oriented finite elements: II. A symbolic environment for automatic programming, *Comput. Methods Appl. Mech. Engrg.*, vol. 32 (1996) pp. 259-276.

[8] J. Korelc, Multi-language and Multi-environment Generation of Nonlinear Finite Element Codes, *Engineering with Computers*, vol. 18 (2002) pp. 312-327.

[9] A. Logg, Automating the Finite Element Method, *Arch Comput Methods Eng.*, vol. 14 (2007).

[10] R. Saad and D. Eyheramendy, Generalized finite element discretization: an object-oriented analysis, *ECCM 2010*, Paris, May 21-26 2010.

[11] R. Saad, Sur une approche à objets généralisée pour la mécanique non linéaire, *PhD Thesis report, Université de Provence*, 2011.

[12] R. Saad and D. Eyheramendy, An advanced Java approach for the development of FE codes, *ECT2012, Dubrovnik, Croatia*, September 2012.

[13] D. Eyheramendy and R. Saad, An object-oriented Java tensor approach for finite elements derivation, *ECCOMAS 2012*, Vienna, Austria, Sept. 10-14 2012.