CD-ROM ISSN 2247–8337
Online ISSN 2559-7841
ISSN-L 2247–8337

Effective development of a finite-element solver at the University

Romain Boman[1], Luc Papeleux[1] and Jean-Philippe Ponthot[1]

[1]*Department of Aerospace and Mechanical Engineering, University of Liège, Belgium*

Abstract

Software development is usually neglected by academic researchers in the field of computational solid mechanics. This lack of long-term strategy often leads to the loss of valuable numerical models and algorithms. In the first part of this talk, the key ideas behind the management of the source code of METAFOR, a nonlinear finite-element solver developed at the University of Liège, are presented in detail. The primary goal is to continuously integrate all the developments into a single application so that future projects can safely rely on the results from the past. The second part of the talk exhibits several applications of metal forming processes computed with METAFOR and based on the Arbitrary Lagrangian Eulerian formalism.

Keywords: Software development, Finite Element Method, Arbitrary Lagrangian Eulerian formalism.

1. Practical Management of Simulation Codes

Nowadays, conducting research in the field of computational solid mechanics and the modelling of forming processes requires the development of large simulation programs or, at least, complex libraries gathering numerical methods, models of constitutive laws and other algorithms. As long as academic researchers at the university are seldom computer enthusiasts, and since they are usually much more interested in the physical meaning of their results than the health of their code, the simulation program can become quickly unmaintainable, erroneous or simply inefficient for industrial applications. At the end of a PhD thesis, the laboratory can sometimes receive from the student as many different versions of the code as the number of applications in the manuscript. These versions are usually incompatible with each other and would require a huge amount of work to be tested and merged in a unique version. The situation is even worse for projects that are run in parallel and led by different teams.

This critical but rather common situation was experienced by our laboratory at the University of Liège 20 years ago during the early development of our in-house implicit finite-element solver METAFOR. At that time, METAFOR was written in old-style FORTRAN and simultaneously extended by several researchers in incompatible directions. In 2000, it was decided to rewrite the code using a more modern approach and to adopt a much more robust way to manage the evolving architecture of the code and the integration of its new features.

In this talk, the common issues appearing when several people, who are not necessarily computer experts, want to conduct their own research in a shared coding framework are presented. Then, the strategy used for the development of METAFOR are described in detail. The proposed methods are based on a trade-off between the robust procedures used by commercial software companies and the simplicity required by the University. The advantages of modularity and object-oriented programming will be discussed: each developer can be isolated in his own library so that major conflicts rarely occur in practice. A version control system is also used in conjunction with an extensive test suite which should be verified and enriched by everyone before each modification of the source code.

Another interesting feature of METAFOR is its python interface which is automatically generated from the source code. This second programming layer provides the researcher with a much more user-friendly environment for testing and implementation of new ideas. Python scripting is also very convenient for running coupled multiphysics model, where each physics is computed by distinct solvers [1]. A last application of this layer is the development of dedicated and highly specialised applications around a numerical model for the industry.

2. Numerical Applications

In this second part, several numerical applications are presented in the field of metal forming processes and welding. One of the main features of METAFOR is the ability to solve the equations of motion in an Arbitrary Lagrangian Eulerian (ALE) formalism. In this case, the motion of the mesh and the material can be uncoupled so that mesh distortion is avoided. The simulation of a forging operation of a thixotropic material is presented as an example of such models where the ALE method avoids expensive remeshing operations [2].

Another kind of problem which is very well suited for the ALE formalism is the modelling of stationary processes where the mesh can be fixed along the direction of the material flow. Continuous roll forming has been successfully simulated with METAFOR using this approach [3]. The model includes a basic treatment of in-line welding for closed sections, as well as the final cutting operation which partially releases the residual stresses and allows us to predict the final shape of the part after springback.

Friction stir welding can also be effectively modelled with the ALE formalism and stationary meshes [4]. Temperature fields obtained with METAFOR are compared to the results found in the literature.

Finally the very first simulations of additive manufacturing obtained with METAFOR are presented as a conclusion.

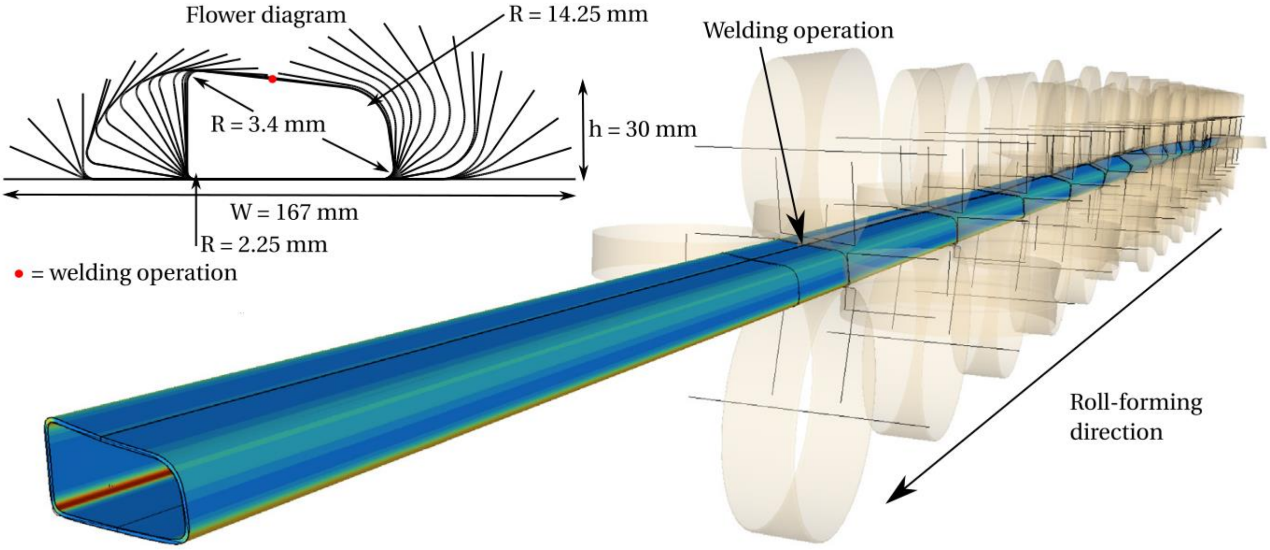

Figure 1: *Simulation of continuous roll forming with METAFOR using the Arbitrary Lagrangian Eulerian formalism.*

References

[1] Thomas D., Cerquaglia M. L., Boman R., Economon T., Alonso J., Dimitriadis G. & Terrapon V. *CUPyDO – An integrated Python environment for coupled fluid-structure simulations*. Advances in Engineering Software, 2018.
[2] Boman R., Koeune R. & Ponthot J.-P. *Numerical Simulation of Double Cup Extrusion Test Using the Arbitrary Lagrangian Eulerian Formalism*. In Numerical Simulations of Coupled Problems in Engineering, 29-53, Springer, 2014.
[3] Crutzen Y., Boman R., Papeleux L. & Ponthot J.-P. *Continuous roll forming including in-line welding and post-cut within an ALE formalism*. Finite Elements in Analysis and Design, 143, 11-31, 2018.
[4] Bussetta P., Feulvarch E., Tongne A., Boman R., Bergheau J.-M. & Ponthot J.-P. *Two 3D thermomechanical numerical models of friction stir welding processes with a trigonal pin*. Numerical Heat Transfer: Part A. Applications, 70(9), 995-1008, 2016.