Some Industrial Challenges for Modelling Software in Solid Mechanics

Nicolas TARDIEU[1,2]

[1] EDF Lab Paris-Saclay, 7 boulevard Gaspard Monge, 91120 Palaiseau, France
[2] IMSIA, UMR 9219, 828 Boulevard des Maréchaux, 91762 Palaiseau Cedex, France

Key Words: *Solid Mechanics, legacy codes, High Performance Computing.*

Computational software have become essential tools in all sectors of the industry. They are used throughout the entire production cycle, from design, to manufacturing and to failure expertise. In the context of solid mechanics, they are often what is called "legacy codes", that is to say software initially designed and developed in the past century and who have lived several refurbishments in order to stay useable and performing. The question of performance is often a tricky one in that context where parallelism has to be considered. From a Single Program Multiple Data point of view, the data structures must be distributed among processors and the numerical treatments and their coherence must be ensured by the developer. In an old Fortran77 code, it is a difficult task. There are examples of highly parallel FEA software in mechanics that are able to scale up to a thousand cores such as Elmer°[1]. However they often cover neither the wide range of feature offered by an all-purpose software nor the specificities of specialized fields such as civil engineering or porous media. This paper is dedicated to the extension to massive parallelism of an all-purpose legacy software and to the solution we have experienced. Code_aster [2] is a legacy code, developed since 1989 by the company Electricité de France (EDF, Electricity of France in English), one of the biggest utility in the world. Its core is in Fortran77 and it enjoys a high-level programming interface in Python. It is released under the GNU GPL license since 2001. Since 2 years, we are taking up the challenge to run it in a massively parallel framework. A new object-oriented design, written in C++, has been set up. Objects and their methods have been defined to wrap the data and the numerical operations, that are both stored and implemented in the Fortran core. The main role of the C++ layer is to ensure the coherence of the data by using adequate communications. Thanks to this design, we have been able to run linear and non-linear quasi-static and linear dynamic problems with half a billion unknowns on more than 2000 processes with good scalability.
Several challenges remain through, essentially due to specific features in solid mechanics that induce linear algebra and numerical implementations difficulties. They are, for instance, the finite element mixing and the use of kinematic constraints, applied by Lagrange multipliers, the latter often being a consequence of the primer. The engineer often models parts of a structure using 3D continuum mechanics elements and others parts using shell elements and connects the different degrees of freedom using kinematics constraints. It raises several problems: the definition of an efficient preconditioner supporting the mixing of finite elements is tricky; the linear system is turned into a saddle point and the use of "discrete" Lagrange multipliers makes it difficult to define an efficient algorithm to solve it. These problems are actually addressed within a collaborative project called PAMSIM [3], involving several research teams. We shall end the presentation with an illustrating industrial problem with the case of the modelling of a containment building of a nuclear power plant that remains a big challenge, when it comes to HPC.

REFERENCES

[1] CSC. *Elmer website*, http://www.csc.fi/english/pages/elmer/
[2] EDF-R&D. *Code_Aster website*, http://www.code-aster.org/
[3] EDF-R&D. *PAMSIM website*, http://www.researchgate.net/project/PAMSIM-HPC-in-solid-mechanics-PIA