Parallel Hierarchical Architectures: The Automatic Building of High Performance Simulators

Felix C. G. Santos[⋆], Eduardo R. R. Brito Jr., José Maria B. Silva, and
Manassés N. Monteiro

Universidade Federal de Pernambuco - Department of Mechanical Engineering
Rua Acadêmico Hélio Ramos, s/n - Recife - PE 50740-530 - Brazil
flxcgs@yahoo.com.br
errbj@yahoo.com.br
zemaria@ufpe.br
http://www.demec.ufpe.br

Abstract. MPhyScas - Multi-Physics and Multi-Scale Solver Environment - is a computational system aimed at supporting the automatic development of simulators for coupled problems, developed at the Department of Mechanical Engineering of the Federal University of Pernambuco - Brazil. It provides a framework, which is flexible enough to accommodate representations for all levels of computation that can be found in simulators based on the finite element method. MPhyScas is built on a set of a powerful language of patterns supporting abstractions for solution algorithms; phenomena, geometric entities; phenomenon-phenomenon and phenomenon-geometry relationships and others, together with a library of low level entities - like finite elements, reference finite elements, numerical integration tools, and so on. It also defines interfaces to other software components like linear solvers, mesh generation among others. In despite of its completeness in what regards all stages of a multi-physics simulation, the current version of MPhyScas produces sequential simulators only. Thus, it does not support any kind of communication between its computational entities besides those defined by direct references (pointers). In this work we present the architecture of an improvement of MPhyScas, called MPhyScas-P (MPhyScas Parallel), which can be used for the automatic development of either sequential or parallel simulators. We take an advantage of the architecture in layers of MPhyScas in order to define a hierarchical parallel computational scheme in such a way that communication and synchronization procedures are automatically identified, localized and built. That hierarchy also provides a natural way of defining data structures and access dynamics for all memory levels, providing simpler ways of dealing with non-uniform memory access patterns. Furthermore, since its workflow has a DAG (direct acyclic graph) structure, the architecture of MPhyScas-P allows for dynamic workflow analysis, balancing and scheduling. Some preliminary results are presented and analyzed.

[⋆] This work was supported by fundings from FINEP and Cenpes/Petrobrás

Key words: High performance computing, simulators, hierarchical parallelization, finite element method

MPhyScas (Multi-Physics Multi-Scale Solver Environment) is an environment dedicated to the automatic development of simulators based on the finite element method. The term multi-physics can be defined as a qualifier for a set of interacting phenomena, in space and time. These phenomena are usually of different natures and may be defined in different scales of behavior (macro and micro mechanical behavior of materials). A multi-physics system is also called a system of coupled phenomena. If two phenomena are coupled, it means that a part of one phenomenon's data depends on information from other phenomenon. Such a dependence may occur in any geometric part, where both phenomena are defined. Multi-physics and multi-scale problems are difficult to simulate and the building of simulators for them tend to be very demanding in terms of time spent in the programming of the code. The main reason is the lack of reusability. A detailed discussion can be found in [2], [3]. Usually, simulators based on the finite element method can be cast in an architecture of layers. In the top layer global iterative loops (for time stepping, model adaptation and articulation of several blocks of solution algorithms) can be found. This corresponds to the overall scenery of the simulation. The second layer contains what is called the solution algorithms. Each solution algorithm dictates the way linear systems are built and solved. It also defines the type of all operations involving matrices, vectors and scalars, and the moment when they have to be performed. The third layer contains the solvers for linear systems and all the machinery for operating with matrices and vectors. This layer is the place where all global matrices, vectors and scalars are located. The last layer is the phenomenon layer, which is responsible for computing local matrices and vectors at the finite element level and assembling them into global data structures. The definition of those layers is important in the sense of software modularization. But it does not indicate neither how entities belonging to different layers interact nor what data they share or depend upon. That is certainly very important for the definition of abstractions, which could standardize the way those layers behave and interact. The architecture of MPhyScas presents a language of patterns in order to define and represent not only a set of entities in each layer - providing the needed layer functionalities - but also the transfer of data and services between the layers. Thus, MPhyScas is a framework that binds together a number of computational entities, which were defined based on that language of patterns, forming a simulator. Such a simulator can easily be reconfigured in order to change solution methods or other types of behavior, [1]-[3]. Almost every single piece of code that constitute MPhyScas computational entities in a simulator can be reused in the building of other different simulators. This makes the simulators produced by MPhyScas strongly flexible, adaptable and maintainable. However, the original architecture of MPhyScas only provides support for the automatic building of sequential simulators. For instance, it does not have abstractions that could automatically define the distribution of data and processes and their relationships across a cluster of PC's. We will briefly present MPhyScas architecture for sequential simulators (called MPhyScas-S) in order to proceed with the main part of the work, i.e., the definition of a new architecture. This new architecture, MPhyScas-P, should satisfy a number of new requirements, including the support for the parallel execution of the simulators in clusters of PC's. MPhyScas-S is a framework with the support of an extended library and a knowledge management system. MPhyScas-P uses the same extended library and knowledge management system from MPhyScas-S. It also makes use of the concept of layers already used in MPhyScas-S in order to define a hierarchical parallel structure. Such a hierarchy is useful to define synchronization schemes; data partition and distribution procedures; inter-process communication patterns and data management across several levels of memory. Processes are automatically specialized during the pre-processing, during the simulation and during the post-processing phases, depending on the characteristics of the hardware being used. Also, two types of communication between processes during a simulation are identified and patterns are defined for their representation. All those aspects will be defined with some detail and preliminary results will be shown and analyzed. The model problem used as a test case is the quasi-static problem of elasticity coupled to the diffusion of hydrogen in the sense that diffusion is driven by the stress state and the mechanical load is coupled to the concentration of hydrogen. Suitable boundary conditions are prescribed on the geometric parts of the contour in order to comply with a manufactured analytic solution, to which the numeric solutions will be compared. The following aspects were considered: (i) simplicity and time costs of coding tasks; (ii) reusability of software components, when changing solution algorithms; (iii) speed up obtained. Since we use an SPMD scheme, we can not expect that the speed up to be different from other more traditional software packages. However, the intention with this work is to show that same speed ups can be obtained with much less reprogramming, even when changing radically the solution algorithms, due to the architecture of MPhyScas-P and to its high reusability. Besides that, due to the DAG (direct acyclic graph) structure of the workflow of MPhyScas distributed among four layers, the possibilities for execution of dynamic MPMD schemes are being currently pursued. One negative aspect is that the preprocessing becomes a little complex in order to maintain the same workflow structure of MPhyScas-S. However, the extra time needed for that is not large.

References

1. Lencastre, M., *Conceptualisation of an Environment for the Development of FEM Simulators*, Recife, Pernambuco, Universidade Federal de Pernambuco, Doutorado em Ciências da Computação, 2004
2. Santos, F. C. G. and Brito, E. R. R. JR. and Barbosa, J. M. A., *Phenomenon Computational Coupling: relationship between phenomena on multi-physics simulation*, 20th European Conference on Modelling and and Simulation, Bonn, Germany, 2006
3. Santos, F. and Brito, E. R. R. JR. and Barbosa, J. M. A. and Silva, J. M. B. and Santos, I. H. F., *Toward the Automatic Development of Simulators for*

Multi-physics Problems, International Journal of Modelling and Simulation for the Petroleum Industry, vol. 1(1), 2007