XX Fluid Mechanics Conference KKMP2012,
Gliwice, 17-20 September 2012

A PARALLEL MODULAR FRAMEWORK FOR MULTI-SCALE AND MULTI-PHYSICS FINITE ELEMENT SIMULATIONS OF FLUID FLOW

Kazimierz MICHALIK[1], Krzysztof BANAŚ[1,2], Przemysław PŁASZEWSKI[1],
Paweł CYBUŁKA[1]
1 Department of Applied Computer Science and Modelling,
AGH University of Science and Technology, Cracow, Poland;
2 Institute of Computer Science,
Cracow University of Technology, Cracow, Poland
E-mail: kamich@agh.edu.pl, kbanas@pk.edu.pl

Key words: FEM, multi-scale, multi-physics, flow simulations, adaptivity, parallelism

Abstract.

We present the design and it's implementation for a flexible and robust modular finite element framework, called ModFem. The design is based on reusable modules which use narrow and well-defined interfaces to cooperate with each other. At the top of the architecture there are problem dependent modules, with the main module being an incompressible flow solver. Problem dependent modules can be additionally grouped together by "supermodules". This structure allows for applying the code to multi-physics and multi-scale problems.

1. Motivation.

Most of commercial and open-source finite element codes' architectures are based on a large part, called a finite element core, that can be extended by writing special procedures, usually for new materials or physical phenomena. Because of that, it is difficult to make deeper and extensive changes to such codes. However, when trying to solve multi-physics and multi-scale problems, such changes often become indispensable. Our goal is to develop a finite element code based on a modular architecture (Banaś, 2004), with modules smaller than in standard FEM codes. We believe that smaller modules are easier to manipulate than a large finite element core and, hence, they can modified for the purpose of special problems solved, like multiphysics and multi-scale problems.

2. Modular structure of the code.

The ModFem framework provides modular structure, with all interfaces defined and modules already implemented. Modules are independent from each other and every module can be easily exchanged with another module with the same functionality, as long as it properly implements the module interface. The whole program is split into:

- Problem module
- Problem utilities module
- Approximation module
- Parallel approximation module overlay
- Mesh manipulation module
- Parallel mesh module overlay
- Solver module

Modules interact only through strictly defined interfaces. All modules are implemented for cross platform usage and all mainstream compilers support. Process of application building is

provided by cross platform compilation and linking framework.

3. Fluid flow simulations and multi-physics capabilities.

The provided problem modules are capable of solving incompressible fluid flows using well established SUPG-type stabilization. Problem is defined in couple of input files covering meshes, fields, parameters, materials, boundary conditions, solver parameters, output format etc. The same problem module can be used for stand-alone fluid flow simulations, as well as for coupled fluid flow – heat transfer problems. For the purpose of the latter problems two problem modules – fluid flow and heat transfer – are coupled using a problem ”supermodule”. The interfaces of problem module procedures are specially designed in order to allow for such a multi-physics coupling. In near future, additional modules, such as micro-scale module for phase transitions and electro-magnetics module (to allow for solving coupled MHD (magneto-hydrodynamics) problems), are planned to be added to the framework.

4. Adaptivity.

All mesh manipulation modules used in the code are designed to support different forms of adaptivity (Banaś, Michalik, 2010). At present there are two modules, one with prismatic elements and the second for hybrid meshes, composed of prisms and tetrahedrons. Both modules support h-adaptivity. There is another module, currently being developed, for r-adaptivity with remeshing, designed to support moving meshes and fluid-structure interaction problems.

5. Parallelism.

Two levels of parallelism are exploited in the framework. First, global level parallelism with distributed memory is implemented using parallel overlays for mesh and approximation modules. It handles domain decomposition, load balance, synchronization of computational nodes etc. At the second level of parallelism we focus on a single computational node. Both levels support multiple meshes. At the global level, parallel overlay module decompose mesh into sub-meshes, which can me merged, splitted or changed. At computational node level there can be distinct meshes for different fields (for multi-physics problems) or several approximation fields can share the same mesh.

References

K. Banaś, (2004): *A model for parallel adaptive finite element software*, in: R. Kornhuber, R. Hoppe, J. P´eriaux, O. Pironneau, O.Widlund, J. Xu (Eds.), Domain Decomposition Methods in Science and Engineering, Vol. 40 of Lecture Notes in Computational Science and Engineering, Springer, pp. 159–166.

K. Banaś(2004): *A modular design for parallel adaptive finite element computational kernels*, in: M. Bubak, G. van Albada, P. Sloot, J. Dongarra (Eds.), Computational Science — ICCS 2004, 4th International Conference, Kraków, Poland, June 2004, Proceedings, Part II, Vol. 3037 of Lecture Notes in Computer Science, Springer, pp. 155–162.

K. Banaś, K. Michalik (2010): *Design and development of an adaptive mesh manipulation module for detailed FEM simulation of flows,* Procedia Computer Science ; ISSN 1877-0509. — 2010 vol. 1 iss. 1 s. 2037–2045