8th. World Congress on Computational Mechanics (WCCM8)
5th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008)
June 30 –July 5, 2008
Venice, Italy

FLECS: A FLExible Coupling Shell

*** Aukje de Boer[1], Margreet Nool[2], Erik Jan Lingen[3] and Hester Bijl[1]**

[1] Delft Univ. of Technology
P.O. Box 5058, 2600 GB
Delft, The Netherlands
{a.deboer, h.bijl}@tudelft.nl

[2] CWI
P.O. Box 94079, 1090 GB
Amsterdam, The Netherlands
margreet.nool@cwi.nl

[3] Habanera
Kluyverweg 6, 2629 HT
Delft, The Netherlands
e.lingen@habanera.nl

Key Words: *Coupled problems, Partitioned schemes, Coupling shell, Interface coupling, Fluid-Structure Interaction.*

ABSTRACT

Numerical simulations involving multiple, physically different domains can be solved effectively by coupling simulation programs, or solvers, in a partitioned approach. This means that it is possible to use sophisticated solvers for the individual domains. The coordination of the different solvers is commonly handled by a coupling shell. A coupling shell synchronizes the execution of the solvers and handles the transfer of data from one physical domain to another.

The majority of coupling shells used today are embedded subprograms that have been developed for coupling two specific solvers. This makes it hard, if not impossible, to replace one solver by another, or to experiment with new transfer algorithms. One exception is MpCCI, a generic coupling shell that can be used both as an embedded sub-program and as a separate program. Although MpCCI is relatively easy to use and provides many advances features, it is less suitable for research that is aimed at developing new data transfer algorithms and accelaration techniques. The primary reason is that MpCCI is a commercial product of which the source code is not accessible.

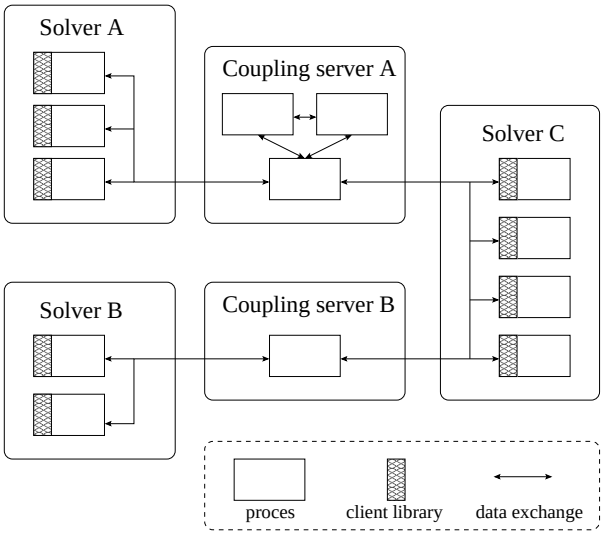

Figure 1: Schematic representation of FLECS.

We aim for a more generic approach and therefore developed FLECS, a flexible coupling shell, designed for implementing and applying an interface for multidisciplinary simulations. The intend is not to achieve the best possible efficiency or to support a large feature set, but to provide a flexible platform for developing new data transfer algorithms and coupling schemes.

The design of FLECS is based on a client-server model in which two solvers communicate with a separate program called the coupling server that is responsible for transferring data from one physical domain to another, see Figure 1. This design is minimally intrusive in the sense that there is no need to change the structure of the solver programs, only some subroutine calls have to be made. This design also makes it possible to run the solvers on two different computers that are located at different research institutes. FLECS uses MPI-2 to transfer data between the solver processes and the coupling server.

If a solver program comprises multiple parallel processes, then each process will be linked to its own copy of the client library, and will communicate with the coupling server through one of the solver processes, see Figure 1. To limit the complexity of the coupling server, it can only couple two solver programs at a time. However, one can couple a solver program to two other solver programs by starting a second coupling server, see Figure 1.

FLECS is open-source software and currently limited to couple sequential solvers. At the moment FLECS is extended to support solvers that run on parallel computers, in order to make FLECS suitable for large applications. In particular, FLECS will be able to deal with data sets that have been distributed over multiple parallel processes. In addition, FLECS will support the implementation of parallel data transfer algorithms.

In principle the subroutines for transferring data from one domain to the other must be implemented by the user. At the moment a data transfer algorithm based on radial basis function interpolation is already provided. This method gave good results in a comparison in [1] and is currently adapted to be able to work in parallel.

As the research in our group is focussing on efficiently solving fluid-structure interaction (FSI) problems the next step will be to implement accelaration techniques. Higher order time integration methods have proven to be very efficient for FSI simulations [2]. For strongly coupled problems the larger time step allowed by the higher order methods gives the need to use more subiterations within one time step. To reduce the number of subiterations Newton-Krylov methods [3] or reduced order models [4] can be used. Both methods can be implemented as subroutines within FLECS without having to change the separate solvers involved. The computational costs of a simple subiteration can be reduced by multilevel accelaration techniques [5]. The use of these techniques is simplified by FLECS, as FLECS supports multiple grids on a single interface. In this way the infomation transfer over the interface at both coarse and fine grid levels can be handled.

REFERENCES

[1] A. de Boer, A.H. van Zuijlen and H. Bijl, “Review of Coupling Methods for non-matching meshes”, *Comput. Methods in Appl. Mech. and Eng.*, Vol. **196**, 1515–1525, 2007.

[2] A.H. van Zuijlen, A. de Boer and H. Bijl, ”Higher-order time integration through smooth mesh deformation for 3D fluid-structure interaction simulations”, *J. Comp. Phys.*, Vol. **224**, 414–430, 2007.

[3] C. Michler, E. H. van Brummelen and R. de Borst, “An Interface Newton-Krylov Solver for Fluid-Structure Interaction”, *Int. J. Num. Methods in Fluids*, Vol. **47**, 1189–1195, 2005.

[4] J. Vierendeels, L. Lanoye, J. Degroote and P. Verdonck, ”Implicit coupling of partitioned fluid-structure interaction problems with reduced order models”, *Computers and Structures*, Vol. **85**, 970–976, 2007.

[5] A.H. van Zuijlen, S. Bosscher and H. Bijl, ”Two level algorithms for partitioned fluid-structure interaction computations”, *Comput. Methods. in Appl. Mech. and Eng.*, Vol. **196**, 1458–1470, 2007.