43

An overview on current multiphysics software strategies for coupled applications with interacting physics on parallel and distributed computers

P. CHOW[1] and C. ADDISON[2]

INTRODUCTION

Computational Multiphysics is generally defined as the combining/coupling of technologies in various developed fields and disciplines, e.g. fluid dynamics, heat transfer, solid and structure mechanics, and electromagnetic, to accurately model the physical phenomena and the interacting processes.

Previously, it was the simplifying assumptions used - because of past numerical algorithms and computing technology - that had constrained the models. Now with advances in computer technology, modelling techniques and numerical methods (e.g. domain decomposition methods - applied with good success in linear and non-linear solvers, exchange of quantities between non-matching grids, and data partitioning in parallel computations), it is no longer the case. Today's engineers and scientists have the tool, in Multiphysics software, to study and tackle tough problems with interacting physics, such as:

1 Fujitsu European Centre for Information Technology Ltd, 2 Longwalk Road, Stockley Park, Uxbridge, Middlesex UB11 1AB, U.K. email: chow@fecit.co.uk
2 Address same as above.
Eleventh International Conference on Domain Decomposition Methods
Editors Choi-Hong Lai, Petter E. Bjørstad, Mark Cross and Olof B. Widlund

- Fluid-structure interaction in bridges and high-rise structures;
- Metal shape casting of high-quality engineering components.

Employing it on parallel computers ensures the solution is deliverable in a practical timeframe.

The big question is how to combine the different disciplines in order to solve and obtain accurate and efficient solutions to these types of multidisciplinary applications. In this paper, we give an overview of current mesh-based computational software technology for Multiphysics simulations. Software strategies such as Single-Code and Multiple-Codes models, coupling libraries, and agents request brokers, are covered.

MODELLING & COMPUTING PERSPECTIVE

Constructing computational "*Analysis-Engines*" for Multiphysics applications is NOT simple. Commonly involving a multidisciplinary team that can include engineers, physicists, chemists, mathematicians, computer and computational scientists, if not all and more. In larger projects it can span to multiple groups and organizations that may geographically be located miles apart. For international ones it could mean different continents of the world.

Often each discipline has their preferred methods. In numerical discretization, the Finite Element Method is commonly associated with Solid and Structure Dynamics. While the Finite Volume Method is popular in Computational Fluid Dynamics. In programming languages, Fortran and C are commonly associated with mature or legacy codes, with C mostly in visualization and Fortran in analysis. Of the new object-oriented languages, C++ is popular with the scientific community. Language inter-operability is difficult in both code integration and reuse of libraries when they are coded in different programming languages. Inter-operability between Fortran and C is included in the next revision of the language standard. But no such standard is near for object-oriented languages, primarily becasue of difficulties with structures and objects.

Multiphysics applications generally involve solving highly non-linear system of equations, commonly in three-dimensional space, time-dependent and complex geometry with unstructured meshes. Meshes into millions of elements and 10 or more unknowns in applications such as transport vehicles, engines, and electronic packaging are anticipated. Parallel processing is a core element to running and to delivering the solution in a practical time.

SOFTWARE STRATEGIES

Today, the software strategies used to put together multiphysics analysis-engines can be categorized into two types of models, Single-Code and Multiple-Codes. In scalar processing settings, Multiple-Codes models would have a number of independent analysis codes with a coordinating program to oversee controls. Often there is a common database and the utilities for mapping the data between the analysis codes and/or the database. In Single-Code models, the analysis-engine is a single analysis code.

The coming subsections go into each model in more detail and the parallel processing paradigms. Also, where possible, giving some examples of the software in each category.

Single-Code Model

The Single-Code model is more common with newer analysis codes for addressing multiphysics problems. This is no surprise as new methods and algorithms are developed for coupled analysis, it is natural to implement them under a single-code model (single program). For example, codes such as SPECTRUM [spe] from Centric Engineering System, Inc. and PHYSICA [phy] from University of Greenwich come under this category.

There are two approaches in which multiphysics are generally achieved under this model:

- *One fully coupled model*: typically using every specific numerical technique to solving the entire coupled non-linear system of equations in a single step. The matrix system is in general ill-conditioned, as a result of the difference in stiffness of the various physical models. Currently, it is very application specific;
- *Build on and combine existing models*: solve each discipline using established methods and using an iterative scheme, such as Newton, to couple the interacting processes between the various phase of the analysis. In most cases some kind of coupling algorithm is required for time integration, solution stability and speed of convergent. The approach shares a lot of the common coupling elements with the Multiple-Code model.

One major advantage of the model is all the data are in one Single Data Space, i.e. no out-of-memory data transfer between various phase of analysis. Consequently, the overall solution procedure is far more efficient than with Multiple-Code models. A possible downside is that these analysis programs are generally a closed system, with only provision for adding boundary conditions, material models, and possibly calling of user routines, in parts of the code. The ability to change internal elements of the code is not possible. One exception is the PHYSICA toolkit. It has an open software analysis-engine framework based on a component-based paradigm to support component exchanges [PCK+98].

Multiple-Codes Model

The Multiple-Codes model is primarily using several independent analysis codes, in an integrated way and frequently with an element of domain decomposition, to solve Multiphysics and coupled problems. Commonly, these codes are well established in a particular field, such as FLUENT [flu] for Computational Fluid Dynamics and NASTRAN [nas] for Computational Structural Dynamics. Often, industries such as automotive and aerospace would have most, if not all, the analysis packages for engineering and design. Running them in a coordinated way and with proper transfer of information between the analysis codes, a framework for solving coupled problems is formed. This is the Multiple-Codes model towards Multiphysics analysis.

The two approaches in which this kind of analysis-engine is usually put together are:

- *From Single Source*: as a fully integrated product supplied by a single software source/vendor, customarily using vendor's own analysis codes. An example is the ANSYS/Multiphysics [ans] suite. Generally these are closed systems.
- *From Multiple Sources*: commonly a purpose built open integration framework based on component paradigm to support a suite of analysis packages from various sources/vendors. Supporting utilities such as data translation, files and databases manager, solution and job control, are essential tools. So is a common application interface for the framework. The ACTS Toolkit [act] and JULIUS [jul] are two examples that come under this category.

These models often have multiple data sets, such as databases, of the same model case for each analysis package (generally not so with single vendor situation). Duplication is common on data such as geometry, material properties and, in some cases the mesh. The flow of data between the packages is frequently done via files or handles. Often there is an extra data translation process for non-matching meshes and variables on different spatial locations (e.g. values from vertex to element). For transient problems with large number of time-steps this can be very inefficient.

With the Multiple-Codes model the integration of the different disciplines' analysis codes is usually done with a loose coupling algorithm [RCJ+95][JR97], as tight coupling is generally viewed not possible without modification to source code of analysis solver. The upside of loose coupling is the parts in the analysis-engine, the analysis codes, can interchanges to cater for application needs.

PARALLEL PROCESSING

Parallel processing add an extra dimension to the analysis-engine software strategy. In the Single-Code model, it very much follows the standard parallel strategies found in conventional analysis codes. The most popular being Single Program Multiple Data (SPMD) paradigm.

The Multiple-Codes approach is generally more complicated but does open up several interesting possibility and models, such as intelligent agent object request broker paradigm [JW98][cor92]. The big question here with the Multiple-Codes model is how to inter-operate with efficient data transfer? For whichever model domain decomposition technology is a key element.

In parallel and distributed environment, some challenging issues for the Multiple-Codes approach are:

- Different codes have different parallel models;
- Communication between codes for data transfer is necessary - a potential bottleneck;
- Dynamic load balancing made even harder in Multiple Instruction Multiple Data (MIMD) setting.

Tools and libraries are being developed to address these points. For example, the COupled COmmunication LIBrary (COCOLIB) being developed in a European Esprit project CISPAR [cis] is addressing these via a library approach. The purpose is to enable matching and exchange of surface interface data on parallel computers, where the analysis codes themselves are parallel codes. The message passing interface (MPI) standard [mpi94] is use in the implementation of COCOLIB, MPI has the facilities to keep communicate separate between the individual components. In the integrated framework side ACTS Toolkit [act], JULIUS [jul] and TOOLSHED [too], are few examples that are in this category.

In the Metacomputing environment, there are several interesting developments and potential models on the horizon. The Internet could potentially be the ultimate compute engine if it can be harnessed and made viable for computational mechanics applications. The inter-operability standard for this kind of environment, such as CORBA and JAVA, are they sufficient for scientific applications such as multiphysics? Are the infrastructure in-placed? What about security matters? These and other infrastructure related questions are being addressed, largely driven by needs of business, commerce, and entertainment sectors. One potential model from this for the Multiphysics analysis-engine is the intelligent agent object broker computing paradigm [JW98][cor92]. One such example is the SciAgents [J.R98][T.T96] from Purdue University. It is agent-based and uses a loose coupling algorithm for distributed components (analysis codes/models) with interface relaxation method for solving composite problems. As the current commercial investments continue to pour heavily into Internet and Web-based solutions for business, commerce, and entertainment, and encouraging an economy of commodity components and services. It is highly that this and related models will develop more rapidly in the near future than the other computing paradigms.

CONCLUDING REMARKS

The demand is increasing in computational mechanics to have the ability to simulate engineering processes with interacting physics. Putting together such an *Analysis-Engine* for Multiphysics simulations is not simple, often requiring a multidiscipline team that commonly involves different numerical methods and codes written in different programming languages. With large models anticipated, meshes into millions of elements and 10 or more unknowns, parallel processing is a core part of the software strategy to running and to delivering the solution in a practical time.

Here, an overview is given on current, and near future, software strategies and models for mesh-based analysis-engines to address Multiphysics problems. The strategies can be categorized into Single-Code and Multiple-Codes models. Each model has a number of approaches depending on the degree of openness, parallel processing, and application needs. Future models maybe more Internet and Web-based related (e.g. intelligent agent object broker computing paradigm) as economy of commodity components (both hardware and software) and services becomes more readily available. But for whichever model domain decomposition technology is a key element.

REFERENCES

[act] ACTS toolkit: Advanced computational testing and simulation toolkit. Department of Energy 2000 Project. U.S.A.

[ans] ANSYS. ANSYS Inc. (http://www.ansys.com). Canonsburg, PA, U.S.A.

[cis] CISPAR - open interface for coupling of industrial simulation codes on parallel computers. ESPRIT Project 20161. E.U.

[cor92] (1992) The common object request broker: Architecture and specification. The Object Management Group and X/Open, John Wiley and Sons.

[flu] FLUENT. Fluent Inc. (http://www.fluent.com). Lebanon, NH, U.S.A.

[JR97] J.R.Cebral and R.Lohner (1997) Fluid-structure coupling: Externsions and improvements. *AIAA-97-0858*.

[J.R98] J.R.Rice (1998) An agent-based architecture for solving partial differential equations. *SIAM news* 31(6).

[jul] JULIUS: Joint industrial interface for end-user simulations. ESPRIT Project 25050. E.U.

[JW98] Jennings N. R. and Wooldridge M. J. (eds) (1998) *Appnt Technology: Foundations, Applications, and Markets.* Springer-Varlag.

[mpi94] (1994) M.P.I. forum, MPI: a message-passing interface standard. Journal of Supercomputer Applications.

[nas] NASTRAN. The MacNeal-Schwendler Corporation (http://www.macsch.com). Los Angeles, California, U.S.A.

[PCK+98] P.Chow, C.Baily, K.McManus, M.Cross, and C.Addison (1998) A single-code software model for multiphysics analysis-engine on parallel and distributed computers with the physica toolkit. In Lai C.-H., rstad P. E. B., Cross M., and Widlund O. B. (eds) *Proceedings of the 11th international domain decomposition conference.* London, U.K. submitted.

[phy] PHYSICA. University of Greenwich (http://physica.gre.ac.uk). London, U.K.

[RCJ+95] R.Lohner, C.Yang, J.R.Cebral, J.D.Baum, H.Luo, D.Pelessone, and C.Carman (1995) Fluid-structure interaction using a loose coupling algorithm and adaptive unstructured grids. In M.Hafez and K.Oshimna (eds) *Computational Fluid Dynamics Review 1995.* John Wiley.

[spe] SPECTRUM. Centric Engineering System Inc. (http://www.centric.com). Sunnyvale, CA, U.S.A.

[too] TOOLSHED: Tools for high productivity engineering design. ESPRIT Project 20231. E.U.

[T.T96] T.T.Drashansky (1996) *An Agent Based Approach to Building Multiple Disciplinary Problem Solving Environments.* PhD thesis, Department of Computer Science, Purdue University.