Parallel aspects of Fluid-Structure Interaction

Eva Casoni*, Guillaume Houzeaux¹, Mariano Vázquez²,³,—

*Barcelona Supercomputing Center, Campus Nord UPC, 08034, Barcelona, Spain
¹IIIA-CSI, Campus Bellaterra UAB, 08193, Bellaterra, Spain

Abstract

This paper presents several parallelization aspects of Fluid-Structure Interaction (FSI) problems in computational mechanics when using an Arbitrary Lagrangian-Eulerian (ALE) scheme. The physical domain of the coupled problem is then solved on two different zones: a first zone for the fluid dynamics and the fluid mesh deformation and a second one for the solid mechanics. The idea can be further extended by adding more physics to the coupled system, such as heat transport (for fluid and solid) or excitable media, among many others. In this paper, the basic two premises are that all problems can already be solved individually in parallel with good scalability and that the coupled system is solved in a coupled way within the same code. The paper introduces the formulation, presents some parallelization issues and proposes how to attack them, presents some results and discuses them and draws some future lines.

© 2013 The Authors. Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
Selection and peer-review under responsibility of the Hunan University and National Supercomputing Center in Changsha (NSCC)

Keywords: Fluid – Structure Interaction, Computational Fluid Dynamics, Computational Solid Mechanics, Parallelization

1. Introduction

Computational Fluid Dynamics (CFD) and Solid Mechanics (CSM) are both modeled by partial differential equations, which when discretized, are transformed in algebraic linear systems. In addition, ALE-based FSI schemes require the solution of a third problem to account for the fluid mesh deformation (called here FMD). Although each equation has its features and numerical problems, their algebraic systems are very similarly created and solved. Taking profit of this fact we solve the coupled problem within one parallel multi-physics code, which allow us to explore a wide range of solving procedures. Despite we use a unique code, our approach is not monolithic; it is based on partitioned techniques that use separate solvers for the structure and the fluid.

This strategy is not the usual way of attacking FSI problems because it needs a concurrent expertise in both fields at the code conception and development. Then, instead of coupling the problems in one code, the alternative is to couple different codes (which usually cumulate years of expertise) that solve each of the problems (or at least two of them, say CFD-FMD or CSM-FMD) through some communication procedure. This second option is very reasonable when one of the problems is largely more expensive in computational effort, a usual case in most of the current FSI applications. Examples are aeroelastic flutter [1], parachutes deformation [2] or arterial flow [3]. In all these cases, CSM is by far the cheapest problem, with small meshes and/or physical models simplifications (i.e. using membranes or shells instead of 3D problems). On the other hand, CFD is always expensive, with its boundary layers, turbulent models or shocks. In between, FMD is usually as large as CFD but relatively cheap with a low number of operations per degree of freedom, which sometimes is attacked with a relatively simple mesh correction. Therefore, the usual parallelization strategy followed in this case is focused on the CFD side (see for instance [1]). However, when the complexity of the CSM problem increases, it is not enough to focus in parallelizing CFD side. The relative size of the CSM problem can grow because the simulation on the solid side requires a
richer material structure and definition. For instance, an aircraft wing simulation taking into account all its parts and materials including composites, or a hemodynamics simulation of arteries taking into account the tissue orthotropic sheet-like structure. A richer solid structure will typically mean finer (then larger) meshes, more expensive material models that include plasticity, fatigue or crack propagation, more complex geometries with different types of elements, etc. Moreover, the CSM model could depend on other groups of equations on top of it to run, like in the case of thermoelasticity, solving heat propagation on the CSM zone, piezoelectric or ferromagnetic materials, solving electromagnetic fields or biological tissue activated by electrical activity, solving excitable media equations. In this paper we present strategies to attack this kind of problems by coupling all the problems into one code with an integrated parallelization scheme. In the conference, some scalability test will be also shown.

2. The Physical problem

The proposed scheme involves the coupled solution of transient CFD+FMD+CSM+PR1+PR2+... problems, where the PRI’s are additional equations that could be present in any (or in both) of the CFD or CSM zones, for instance excitable media or thermal effect. For all the individual problems the governing equations are discretized using Finite Differences for time and Finite Elements for space. The solver used is Alya, the in-house BSC parallel multi-physics code (see for instance [4,5,6]).

FSI schemes are of three basic types: monolithic (for instance [7]), partitioned strongly coupled [1,8] and partitioned loosely [2,3] coupled. In this paper we will focus in partitioned methods, both strongly and loosely coupled. In partitioned methods, the matrix for the coupled system is solved by blocks, each block corresponding to a different problem, either with a block-Jacobi (loosely coupled) or block-Gauss-Seidel (strongly coupled) method. The main difference between both methods is the iteration scheme: in weakly coupled approaches every subproblem is only solved once per time-step but the global problem takes more time to converge since the interface conditions are not fulfilled exactly. In contrast, in strongly coupled the three subproblems are solved iteratively. From the implementation point of view, both methods are very similar, being its main difference the time integration scheme and its internal sub-iterations.

In this paper, the main ingredients are:

- CFD: Transient incompressible flows in an ALE framework (described in [5]).
- FMD: Laplacian equation in a Eulerian framework with a non-uniform diffusion tensor that depends on the size element. This strategy allows preserving regions with smaller elements, like boundary layers (described in [9]).
- CSM: Transient non-linear solid mechanics in total Lagrangian formulation (described in [6]).
- PRI’s: Transient solution of one or more PDE’s (heat, electrical activation, excitable media…) (for instance in [6])

In order to understand the parallel strategies proposed below, let us firstly describe the coupling scenario between the subproblems.

2.1. Geometrical view

In our approach the problem is geometrically divided in non-overlapping fluid and solid zones. For the sake of simplicity, we consider one zone for each problem, with the CFD and FMD problems sharing zone:
FSI interaction occurs at the interface, where fluid exerts work on the solid, which reacts deforming its zone. Conversely, as the solid’s zone deforms, it moves the interface deforming the fluid’s zone too, assuring continuity of velocity and normal forces at the interface. Then, for each coupling iteration the scheme is:

1. Solve CFD on Zone 1 in the ALE framework, considering last FMD solution. This couples CFD and FMD onto Zone 1.
2. Transfer fluid nodal forces to the structure domain at the interface (interface-load), including a push forward operation. This couples CFD to CSM on the interface.
3. Solve CSM on Zone 2 in the total-Lagrangian framework considering last PRI’s solution. This couples CSM to PRI’s onto Zone 2.
4. Solve PRI’s on Zone 2 in the Lagrangian (i.e. deformed) framework. This couples PRI’s to CSM onto Zone 2.
5. Impose the displacement to the FMD for the nodes on the interface. This couples CSM to FMD on the interface.
6. Solve FMD on Zone 1 in the Eulerian framework. This creates the new fluid-domain and mesh for the next time step.
7. Update states and go back to step 1 until convergence.

In all the problems considered here, the nodes lying at the interface are coincident for both zones, eliminating the need for interpolations. In order to impose continuity at the interface and transfer the interface-loads, structure deformation and velocity, once the original complete mesh is generated, the nodes at the interface are simply duplicated and assigned to each of the zones.

2.2. Algebraic view

On the algebraic standpoint, the coupled problem can be seen as a large system of equations of size $N_n \times DOF$, where $N_n = N_{n1} + N_{n2}$ is the number of nodal points of the complete mesh and DOF is the total number of the degrees of freedom, for instance, fluid velocity and pressure, solid and mesh displacement, temperature, electro-magnetic field, electrical activation and so on. In all the cases presented here, the system is grouped in blocks, each one corresponding to one of the problems coupled, and solved block after block in a staggered way. Depending on the time integration scheme chosen, subiterations, Aitken relaxation or both can be applied to ensure stability and acceleration for partitioned FSI-solvers.

3. The parallel strategies

A solution strategy for an FSI problem is inherently sequential. Due to the parallel characteristics of Alya, individual parallelization for each of the coupled problems is guaranteed. With these two premises in mind, we study the following parallelization strategies. Suppose that the total number of elements for the coupled problem is Nel and suppose that we can use $P$ cores. Assume also that the workload of both CFD and CSM is equal, i.e., both problems require meshes of similar sizes. To fix ideas, suppose also that time to solved the FMD problem is almost negligible compared with CFD. Then, several options for the parallel strategy can be considered:

1. The most straightforward option is to partition each zone in $P/2$ subdomains, giving each subdomain to a different core, that is, subdomain $\mathcal{N}$ is run in processor $\mathcal{N}$, for $n$ from 1 to $P$. Then, CFD and FMD are first solved in Zone 1 (with Nel/2 fluid elements) and CSM in Zone 2 (with Nel/2 solid elements). This will raise a sequential algorithm between the two zones. However, due to the parallel characteristics of Alya, the best speed-up one should expect is 50%. This is the baseline speedup.
2. Distribute the partition in subdomains and processors as in 1., but now solve the problems in parallel providing predictor estimates for both FSI forces and displacements. Scalability should be improved, but the use of predictors could result in a larger amount of sub-iterations to achieve convergence at each time step. The use of relaxation methods is here almost mandatory in order to accelerate convergence.
3. Do the same as 1., but “asynchronizing” as much as possible for matrix assembly, that is, consider two subzones within each zone: one containing the interior nodes and the other containing the interface nodes. One possible strategy would be to compute first the interface nodes of Zone 1, then, the interior nodes of Zone 1 and the interface nodes of Zone 2 can be computed at the same time and finally, compute the interior nodes of Zone 2. An improved algorithm would be to compute first all the interface nodes and then, in a second stage, the interior ones. Experimental and theoretical speedup results should be studied for this option.

4. Partition each zone in P subdomains, giving two subdomains (one on the fluid zone, one on the solid zone) to each core and solve sequentially CFD and FMD in Zone 1 (with Ne/2 fluid elements) and CSM in Zone 2 (with Ne/2 solid elements). In this case, no group of cores will be waiting for another group to finish work and the expected speedup should improve.

The strategies will be carefully studied within the Alya code, considering efficiency, viability within the parallel code and computational effort in the implementation in parallel.

4. Results

In the conference, we will present results for different large-scale problems, showing scalability results and analyzing performances in runs of thousands of cores.

![Flapping bar for the Hubner’s problem.](image)

**Hubner’s problem**

This 2D problem is incompressible flow passing a flapping bar attached to a square. The material properties of the bar are density \( \rho = 2.0 \text{ g/cm}^3 \), Young’s modulus \( E = 2.0 \times 10^6 \text{ g/(cm s}^2) \) and Poisson’s ratio \( \nu = 0.35 \). The fluid properties are density \( \rho_f = 1.18 \times 10^{-3} \text{ g/cm}^3 \) and viscosity \( \mu = 1.82 \times 10^{-4} \text{ g/(cm s)} \). The inflow velocity is taken as 31.5 cm/s, leading to a Reynolds number \( \text{Re} = 204 \) if the length of the square rigid body is taken as the characteristic length. Figure 2 shows the stress isolines for the fluid domain. The computation is done with the first strategy described in previous section. Extension to the 3D case will be studied.

**Cardiac fluid-electro-mechanical coupling**

This 3D problem is incompressible flow around a piece of cardiac tissue. The fluid is blood and the solid is an Ogden-Holzapfel tissue with passive and active stress. Electrical activation is a Fitz-Hugh Nagumo model. Electrophysiology is coupled to mechanical deformation through a Hunter-Niederer model (see [6] for a deeper description of the electromechanical coupling). Figure 3 shows several steps of the simulation for the velocity field.
References


Figure 3. A bar of cardiac tissue contracting under electrical activation coupling, surrounded by blood. Arrows are the velocity vectors of blood around the bar.