A COMPUTATIONAL STRATEGY FOR FLUID-STRUCTURE INTERACTION COMBINING IMMERSED DOMAINS AND A BODY-FITTED APPROACH

Enzo A. Dari[*,a,c], **Pablo J. Blanco**[b,c] **and Raúl A. Feijóo**[b,c]

[a]*Centro Atómico Bariloche, Av. Bustillo Km. 9.5, San Carlos de Bariloche, 8400, Rio Negro, Argentina, darie@cab.cnea.gov.ar http://www.cab.cnea.gov.ar*

[b]*Laboratório Nacional de Computação Científica LNCC/MCT, Av. Getúlio Vargas 333, 25651-075 Petrópolis, Brazil, {pjblanco,feij}@lncc.br, http://www.lncc.br*

[c]*Instituto Nacional de Ciência e Tecnologia em Medicina Assistida por Computação Científica, Petrópolis, Brazil, http://macc.lncc.br/*

Keywords: ALE method, mesh adaptation, embedded objects, cardiac valves

Abstract. There are many alternatives for treating the problem of a solid moving completely inside a fluid. One of them is using the fluid equations strictly in the space occupied by the fluid, and adjusting the mesh in each temporal step. If the displacements are limited a node-movement strategy without topology changes may be used to keep low the computational costs. On the other hand, for large displacements and distortion of the computational domain (as usual when modeling cardiac valves), a full remeshing may be needed. Another approach to the problem is using immersed domains: in this case the fluid mesh is fixed and the solid imposes boundary conditions on the fluid through Lagrange multipliers. In this work we present a hybrid approach: we use a single mesh for the fluid, containing the solid, without remeshing. However, we allow the movement of its nodes to adjust to the solid boundaries. In this manner, we avoid the high cost of the remeshing step, while reducing the interpolation errors produced by elements that contain partially portions of the solid. Hence, the strategy is such that the body is always fitted by the same mesh. The main difficulty of this approach is the generation of a valid mesh efficiently (without largely distorted elements), in transferring information between solid and fluid meshes, and in the interpolations needed when re-adjusting. We have obtained some preliminary results working with a parallel Navier-Stokes solver, using geometric searching techniques that allow us to keep bounded the computational costs involved.

---

[*]Member of CONICET, Argentina

1 INTRODUCTION

Throughout the last decade there has been notorious efforts and advances in the use of immersed-like methods to tackle the fluid-structure interaction problem Baaijens (2001); Hart et al. (2003); Glowinski et al. (1998, 1999a,b); Löhner et al. (2004); Patankar et al. (2000); Peskin (2002); Roma et al. (1999). The main advantage of such an approach is that the fluid mesh remains fixed, no needing neither mesh movement nor remeshing. Specifically in the case of immersed solid bodies, several approaches that employ such fixed meshes for the fluid problem *move* the solid on top of them Blanco et al. (2008); Liu et al. (2006, 2007); Wang and Liu (2004); Wang (2006); Zhang et al. (2004). The main issue in this approach is the lack of an accurate representation of the solid/fluid interface, which leads to poor solution at the solid-fluid layers. This is also put in evidence when computing the forces of the fluid acting over the immersed solid. Although some interesting results can be obtained using this approach (see the afore referred literature), an accurate representation of the literature remains an open problem. In this sense, some approaches has proposed to make use of hybrid continuous/discontinuous Galerkin interpolation in order to have optimal convergence of the solution when the interfaces do not match the edges (or faces) of a given mesh Lew and Buscaglia (2008). However, the complexity in this case is that the approach is out of the scope of pure continuous Galerkin interpolations, which are the ones in which we are interested.

In order to remedy this the most popular approach has been to employ an ALE formulation in which the body is accutarely represented because the mesh *follows* the movement of the solid Deparis et al. (2006); Heil (2004); Tallec and Mouro (2001); Tallec et al. (2005); Matthies and Steindorf (2003). In cases with large mesh distortions, this must be accompanied by remeshing techniques. This ingredient makes the ALE approach a not so interesting method, in contrast with the appealing aspects related to the immersed methods.

In this context, the goal of the present work is to combine both techniques in order to retrieve the most interesting features of each one. Thus, we develop an hybrid immersed/ALE approach that, on one hand, accounts for the fluid-structure interaction using the ideas borrowed from the immersed methods (it is an immersed method in that sense), computing the forces over the solid as Lagrange multipliers dual to the imposition of the velocity (of the solid) onto the region occupied by the solid. On the other hand, instead of keeping the mesh fixed it is fitted to the body all the time (it accompanies the solid body), for which an ALE method is then employed. Clearly, at a certain number of time steps the mesh is readjusted in order to avoid bad-quality elements. This is carried out by resorting to a reference mesh (the original non-body-fitted mesh) and performing the adjustment to the solid boundaries allowing the nodes near to the solid/fluid interface to move. This two step procedure will be called the mesh reset. Thus, the topology of the mesh never changes, that is, throughout the whole simulation the methodology is such that it preserves number of nodes, number of elements and connectivity.

The strategy developed here is valid for either deformable solids or rigid bodies. We make use of the finite element method for the numerical approximation of the problem. The preliminary numerical results are obtained for problems involving the interaction of incompressible flows with rigid bodies. Nevertheless, in order to get the desired generality in the algorithm the solid is also represented using a finite element mesh.

Some comments regarding some difficulties of this approach are in order. In the first place, the fitting process must be such that the final body-fitted mesh is a valid mesh (without largely distorted elements). In the second place, the present approach lies in the transfer of information between solid and fluid meshes to determine whether a fluid point is immersed in the body or

not. In the third place, there is need for interpolation of quantities when re-setting the mesh.

This work is organized as follows. Section 2 briefly describes the variational context for the immersed method in the fluid-structure interaction problem. In Section 3 we describe the combined immersed/ALE algorithm, while in Section 4 some preliminary results are provided. The final remarks are outlined in Section 5.

2 IMMERSED APPROACH TO FLUID-STRUCTURE

Let us start with a brief review about immersed methods.for immersed rigid body objects. In this regard we follow Blanco et al. (2008), and the reader interested in the more general formulation for a deformable (compressible or incompressible) solid is directed to the aforereferred work. Then, we provide the linearization used in the present work.

2.1 Variational principle using immersed domains

Let us introduce some notation: Ω is the fluid domain, with Dirichlet and Neumann boundaries denoted by Γ_{f_D} and Γ_{f_N}, Ω_s^t is the region occupied by the solid at time t. Indexes f and s make reference to fluid and solid quantities, respectively. Velocity and pressure fields are denoted by $\mathbf{v}$ and p, and ρ and μ are the density and viscosity, respectively.

Recall that the rigid body velocity is given by

$$\mathbf{v}_s = \mathbf{v}_s^o + \boldsymbol{\omega}_s^o \times \mathbf{r}^o, \tag{1}$$

where $\mathbf{v}_s^o$ is the velocity of the center of mass, $\boldsymbol{\omega}^o$ is the angular velocity of the body and $\mathbf{r}^o$ is the distance from any point within the solid and its center of mass. Equivalently we can write

$$\mathbf{v}_s = \mathbf{v}_s^o + \boldsymbol{\Omega}_s^o \mathbf{r}^o, \tag{2}$$

where $\boldsymbol{\Omega}_s^o$ is skew–symmetric. The accelerations can be written as follows

$$\frac{D\mathbf{v}_s}{Dt} = \frac{D\mathbf{v}_s^o}{Dt} + \frac{D\boldsymbol{\Omega}_s^o}{Dt}\mathbf{r}^o + \boldsymbol{\Omega}_s^o\boldsymbol{\Omega}_s^o\mathbf{r}^o. \tag{3}$$

In deriving this variational problem we use the fact that stresses are not determined during a rigid motion, and they do not contribute to the virtual power balance. As well, since $\mathbf{r}^o$ is the distance to the center of mass, then it is verified that $\int_{\Omega_s^t} \rho_s \mathbf{r}^o \, \mathrm{d}\mathbf{x} = 0$. Hence, we obtain the following variational formulation: for each $t \in (0, T)$ find $(\mathbf{v}, p, \boldsymbol{\psi}) \in \boldsymbol{\mathcal{U}} \times \mathcal{P} \times \boldsymbol{\Psi}$ such that

$$\begin{aligned}
&\int_\Omega \left[\rho_f \frac{D\mathbf{v}_f}{Dt} \cdot \hat{\mathbf{v}}_f + \boldsymbol{\sigma}_{f_D} \cdot \boldsymbol{\varepsilon}(\hat{\mathbf{v}}_f) - p_f \operatorname{div} \hat{\mathbf{v}}_f - \hat{p}_f \operatorname{div} \mathbf{v}_f - \rho_f \mathbf{g} \cdot \hat{\mathbf{v}}_f \right] \mathrm{d}\mathbf{x} - \int_{\Gamma_{f_N}} \bar{\mathbf{t}}_f \cdot \hat{\mathbf{v}}_f \, \mathrm{d}\Gamma \\
&\quad + m_s \frac{D\mathbf{v}_s^o}{Dt} \cdot \hat{\mathbf{v}}_s^o + \frac{D\boldsymbol{\Omega}_s^o}{Dt} \mathbb{I}_s \cdot \hat{\boldsymbol{\Omega}}_s^o + \boldsymbol{\Omega}_s^o \boldsymbol{\Omega}_s^o \mathbb{I}_s \cdot \hat{\boldsymbol{\Omega}}_s^o \\
&\quad - \left[\int_{\Omega_s^t} \rho_f \frac{D\mathbf{v}_f}{Dt} \, \mathrm{d}\mathbf{x} \right] \cdot \hat{\mathbf{v}}_s^o - \left[\int_{\Omega_s^t} \rho_f \frac{D\mathbf{v}_f}{Dt} \otimes \mathbf{r}^o \, \mathrm{d}\mathbf{x} \right] \cdot \hat{\boldsymbol{\Omega}}_s^o \\
&\quad - (m_s - m_f)\mathbf{g} \cdot \hat{\mathbf{v}}_s^o + \int_{\Omega_s^t} \hat{\boldsymbol{\psi}} \cdot (\mathbf{v}_s^o + \boldsymbol{\Omega}_s^o \mathbf{r}^o - \mathbf{v}_f) \, \mathrm{d}\mathbf{x} \\
&\quad + \left[\int_{\Omega_s^t} \boldsymbol{\psi} \, \mathrm{d}\mathbf{x} \right] \cdot \hat{\mathbf{v}}_s^o + \left[\int_{\Omega_s^t} \boldsymbol{\psi} \otimes \mathbf{r}^o \, \mathrm{d}\mathbf{x} \right] \cdot \hat{\boldsymbol{\Omega}}_s^o - \int_{\Omega_s^t} \boldsymbol{\psi} \cdot \hat{\mathbf{v}}_f \, \mathrm{d}\mathbf{x} = 0 \\
&\qquad \forall (\hat{\mathbf{v}}, \hat{p}, \hat{\boldsymbol{\psi}}) \in \boldsymbol{\mathcal{V}} \times \mathcal{P} \times \boldsymbol{\Psi}, \quad (4)
\end{aligned}$$

where

$$m_s = \int_{\Omega_s^t} \rho_s \, \mathrm{d}\mathbf{x} \qquad m_f = \int_{\Omega_s^t} \rho_f \, \mathrm{d}\mathbf{x} \qquad \mathbb{I}_s = \int_{\Omega_s^t} \rho_s (\mathbf{r}^o \otimes \mathbf{r}^o) \, \mathrm{d}\mathbf{x}, \tag{5}$$

and

$$\begin{aligned} \boldsymbol{\mathcal{U}} &= \left\{ (\mathbf{v}_f, \mathbf{v}_s^o, \boldsymbol{\Omega}_s^o) \in \mathbf{H}^1(\Omega) \times [\mathbb{R}]^{n_d} \times ([\mathbb{R}]^{n_d \times n_d})^a ; \, \mathbf{v}_{f_{|\Gamma_{f_D}}} = \bar{\mathbf{v}}_f \right\}, \\ \mathcal{P} &= L^2(\Omega), \\ \boldsymbol{\Psi} &= \mathbf{H}^{-1}(\Omega_s^t), \end{aligned} \tag{6}$$

being $n_d = 2, 3$, $(\cdot)^a$ denotes the skew–symmetric component of a second order tensor and $\boldsymbol{\mathcal{V}}$ is the space of admissible variations of elements in $\boldsymbol{\mathcal{U}}$.

The Euler–Lagrange equations of this formulation are the classical equations of the fluid flow problem, that is the Navier-Stokes equations for a Newtonian fluid like in the present work, together with the equations that describe the rigid body motion. Such equations are not given here for the sake of brevity. It is worth noticing that the Lagrange multiplier $\boldsymbol{\psi}$ is distributed all over the solid region Ω_s^t.

Now, we make use in advance of the continuity of the velocity field in Ω_s^t and replace the following terms in (4)

$$\begin{aligned} \int_{\Omega_s^t} \rho_f \frac{D\mathbf{v}_f}{Dt} \, \mathrm{d}\mathbf{x} \quad &\rightarrow \quad \int_{\Omega_s^t} \rho_f \frac{D\mathbf{v}_s}{Dt} \, \mathrm{d}\mathbf{x} = m_f \frac{D\mathbf{v}_s^o}{Dt}, \\ \int_{\Omega_s^t} \rho_f \frac{D\mathbf{v}_f}{Dt} \otimes \mathbf{r}^o \, \mathrm{d}\mathbf{x} \quad &\rightarrow \quad \int_{\Omega_s^t} \rho_f \frac{D\mathbf{v}_s}{Dt} \otimes \mathbf{r}^o \, \mathrm{d}\mathbf{x} = \frac{D\boldsymbol{\Omega}_s^o}{Dt} \mathbb{I}_f + \boldsymbol{\Omega}_s^o \boldsymbol{\Omega}_s^o \mathbb{I}_f, \end{aligned} \tag{7}$$

and then, being $\mathbf{r}^o = \mathbf{x} - \mathbf{u}_s^o$ the distance to the center of mass $\mathbf{u}_s^o$, (4) results

$$\begin{aligned} \int_\Omega \left[\rho_f \frac{D\mathbf{v}_f}{Dt} \cdot \hat{\mathbf{v}}_f + \boldsymbol{\sigma}_{f_D} \cdot \boldsymbol{\varepsilon}(\hat{\mathbf{v}}_f) - p_f \operatorname{div} \hat{\mathbf{v}}_f - \hat{p}_f \operatorname{div} \mathbf{v}_f - \rho_f \mathbf{g} \cdot \hat{\mathbf{v}}_f \right] \mathrm{d}\mathbf{x} - \int_{\Gamma_{f_N}} \bar{\mathbf{t}}_f \cdot \hat{\mathbf{v}}_f \, \mathrm{d}\Gamma \\ + \Delta m_{sf} \frac{D\mathbf{v}_s^o}{Dt} \cdot \hat{\mathbf{v}}_s^o + \frac{D\boldsymbol{\Omega}_s^o}{Dt} \Delta\mathbb{I}_{sf} \cdot \hat{\boldsymbol{\Omega}}_s^o + \boldsymbol{\Omega}_s^o \boldsymbol{\Omega}_s^o \Delta\mathbb{I}_{sf} \cdot \hat{\boldsymbol{\Omega}}_s^o - \Delta m_{sf} \mathbf{g} \cdot \hat{\mathbf{v}}_s^o \\ + \int_{\Omega_s^t} \hat{\boldsymbol{\psi}} \cdot (\mathbf{v}_s^o + \boldsymbol{\Omega}_s^o (\mathbf{x} - \mathbf{u}_s^o) - \mathbf{v}_f) \, \mathrm{d}\mathbf{x} \\ + \left[\int_{\Omega_s^t} \boldsymbol{\psi} \, \mathrm{d}\mathbf{x} \right] \cdot \hat{\mathbf{v}}_s^o + \left[\int_{\Omega_s^t} \boldsymbol{\psi} \otimes (\mathbf{x} - \mathbf{u}_s^o) \, \mathrm{d}\mathbf{x} \right] \cdot \hat{\boldsymbol{\Omega}}_s^o - \int_{\Omega_s^t} \boldsymbol{\psi} \cdot \hat{\mathbf{v}}_f \, \mathrm{d}\mathbf{x} = 0 \\ \forall (\hat{\mathbf{v}}, \hat{p}, \hat{\boldsymbol{\psi}}) \in \boldsymbol{\mathcal{V}} \times \mathcal{P} \times \boldsymbol{\Psi}, \end{aligned} \tag{8}$$

with

$$\Delta m_{sf} = m_s - m_f \qquad \Delta\mathbb{I}_{sf} = \mathbb{I}_s - \mathbb{I}_f. \tag{9}$$

where $\mathbb{I}_f$ is the same as $\mathbb{I}_s$ (see (5)) but defined in terms of the fluid density.

2.2 Fixed point iterations

Problem (8) is solved using fixed point iterations. Basically, we segregate the solution into two main problems, the fluid and the solid problems. The solution of the coupling problem is obtained by iterating between these two problems using a relaxed Gauss-Seidel method.

The fluid problem is the following: given $(\mathbf{u}_s^o, \mathbf{v}_s^o, \boldsymbol{\Omega}_s^o)$ find $(\mathbf{v}_f, p_f, \boldsymbol{\psi}) \in \boldsymbol{\mathcal{U}}_f \times \mathcal{P} \times \boldsymbol{\Psi}$ such that

$$\int_\Omega \left[\rho_f \frac{D\mathbf{v}_f}{Dt} \cdot \hat{\mathbf{v}}_f + \boldsymbol{\sigma}_{f\,D} \cdot \boldsymbol{\varepsilon}(\hat{\mathbf{v}}_f) - p_f \operatorname{div} \hat{\mathbf{v}}_f - \hat{p}_f \operatorname{div} \mathbf{v}_f - \rho_f \mathbf{g} \cdot \hat{\mathbf{v}}_f \right] \mathrm{d}\mathbf{x} - \int_{\Gamma_{f\,N}} \bar{\mathbf{t}}_f \cdot \hat{\mathbf{v}}_f \,\mathrm{d}\Gamma + \int_{\Omega_s^t} \hat{\boldsymbol{\psi}} \cdot (\mathbf{v}_s^o + \boldsymbol{\Omega}_s^o(\mathbf{x} - \mathbf{u}_s^o) - \mathbf{v}_f) \,\mathrm{d}\mathbf{x} - \int_{\Omega_s^t} \boldsymbol{\psi} \cdot \hat{\mathbf{v}}_f \,\mathrm{d}\mathbf{x} = 0 \quad \forall (\hat{\mathbf{v}}_f, \hat{p}, \hat{\boldsymbol{\psi}}) \in \boldsymbol{\mathcal{V}}_f \times \mathcal{P} \times \boldsymbol{\Psi}, \tag{10}$$

where

$$\boldsymbol{\mathcal{U}}_f = \left\{ \mathbf{v}_f \in \mathbf{H}^1(\Omega);\ \mathbf{v}_{f|\Gamma_{f\,D}} = \bar{\mathbf{v}}_f \right\}. \tag{11}$$

Then, the solid problem reads: given $\boldsymbol{\psi}$ find $(\mathbf{v}_s^o, \boldsymbol{\Omega}_s^o) \in \boldsymbol{\mathcal{U}}_s$ such that

$$\Delta m_{sf} \frac{D\mathbf{v}_s^o}{Dt} \cdot \hat{\mathbf{v}}_s^o + \frac{D\boldsymbol{\Omega}_s^o}{Dt} \Delta \mathbb{I}_{sf} \cdot \hat{\boldsymbol{\Omega}}_s^o + \boldsymbol{\Omega}_s^o \boldsymbol{\Omega}_s^o \Delta \mathbb{I}_{sf} \cdot \hat{\boldsymbol{\Omega}}_s^o - \Delta m_{sf} \mathbf{g} \cdot \hat{\mathbf{v}}_s^o + \left[\int_{\Omega_s^t} \boldsymbol{\psi} \,\mathrm{d}\mathbf{x} \right] \cdot \hat{\mathbf{v}}_s^o + \left[\int_{\Omega_s^t} \boldsymbol{\psi} \otimes (\mathbf{x} - \mathbf{u}_s^o) \,\mathrm{d}\mathbf{x} \right] \cdot \hat{\boldsymbol{\Omega}}_s^o = 0 \qquad \forall (\hat{\mathbf{v}}_s^o . \hat{\boldsymbol{\Omega}}_s^o) \in \boldsymbol{\mathcal{V}}_s, \tag{12}$$

where

$$\boldsymbol{\mathcal{U}}_s = \left\{ (\mathbf{v}_s^o, \boldsymbol{\Omega}_s^o) \in [\mathbb{R}]^{n_d} \times ([\mathbb{R}]^{n_d \times n_d})^a \right\}. \tag{13}$$

All the steps involved in the iterative algorithm will be explained in detail in Section3.

2.3 Approximation by the finite element method

Problem (10) implies solving the Navier-Stokes equations with a constraint in the velocity field distributed all over Ω_s^t. For the fluid problem a $\mathbb{P}_1$-$\mathbb{P}_1$ equal order interpolation is employed with Newton linearization for the convective term. Since our approach involves moving meshes the ALE formulation is employed, and therefore the material derivative $\frac{D\mathbf{v}_f}{Dt}$ in (10) is defined as

$$\frac{D\mathbf{v}_f}{Dt} = \frac{\partial \mathbf{v}_f}{\partial t} + \nabla \mathbf{v}_f (\mathbf{v}_f - \mathbf{v}_m), \tag{14}$$

where $\mathbf{v}_m$ is the velocity of the mesh consistent with the ALE frame of reference. Whenever the mesh needs to be moved we employ a Laplace problem.

In the context of the finite element method the constraint over the velocity field is imposed node-wise, and then the Lagrange multipliers are nodal quantities. Therefore, in problems (10) and (12) we replace the integrals in Ω_s^t as follows

$$\begin{aligned} \int_{\Omega_s^t} \boldsymbol{\psi} \cdot \mathbf{v}_f \,\mathrm{d}\mathbf{x} \quad &\rightarrow \quad \sum_{j \in \mathcal{N}_f} \boldsymbol{\psi}_j \cdot \mathbf{v}_f(\mathbf{x}_j), \\ \int_{\Omega_s^t} \boldsymbol{\psi} \,\mathrm{d}\mathbf{x} \quad &\rightarrow \quad \sum_{j \in \mathcal{N}_f} \boldsymbol{\psi}_j, \end{aligned} \tag{15}$$

where $\mathcal{N}_f$ represents the set of fluid nodes that are in the solid region, and $\mathbf{x}_j$ the corresponding coordinates. Note now that $\boldsymbol{\psi}_j$ is a quantity (a vector) defined in each node.

In turn, problem (10) entails integrating the rigid body equations for the computation of the velocity of the center of mass and the rotational degrees of freedom. An implicit second order method is used for this purpose.

For the hybrid approach we will need to reset the mesh in order to avoid high distortions due to the solid movement. Then, we make use of two meshes (with the same topology): the reference mesh, denoted by $\mathcal{T}_f^0$, that is the original mesh used for the fluid domain that does not necessarily is body fitted, and the adjusted (body-fitted) mesh, denoted by $\mathcal{T}_f^t$, which is the same mesh after fitting the solid boundaries through some procedure that will be detailed in the next section. Whenever we need to make reference to a previous stage of the body fitted mesh we refer to it as $\bar{\mathcal{T}}_f^t$. As already said, for the sake of generality the solid is described using a second mesh, which is denoted by $\mathcal{T}_s^t$ (this mesh moves -as a rigid body- with time as a result of the fluid-structure interaction problem).

In order to identify the points in the fluid domain that occupy the solid region we compute a level set function ϕ_f, with its zero level describing the boundary of Ω_s^t. With this, we are able to compute the distance of the fluid nodes to the solid in order to determine whether a fluid node is inside or outside the solid domain Ω_s^t.

3 IMMERSED/BODDY-FITTED ALGORITHM FOR FLUID-STRUCTURE

3.1 Hybrid immersed/body-fitted algorithm

This section is devoted to describing the steps used in the algorithm. Whenever we need to make reference to solution at the previous time step we employ the index n, while the solution at the current time step and at the previous iteration is denoted by k, and the current iteration is denoted by $k+1$.

For each time step the algorithm is the following.

1. Perform the mesh reset and readjust (this is always done in the first time step) as follows
   a. reset the previous body fitted mesh $\bar{\mathcal{T}}_f^{t_n}$, that is, go back to the reference mesh $\mathcal{T}_f^0$ (this is not necessary in the very first time step),
   b. adjust the fluid mesh to fit the solid body
      i. for the solid domain $\Omega_s^{t_n}$ compute the level set function $\phi_{f,0}^n$ in $\mathcal{T}_f^0$;
      ii. build the set $\mathcal{N}_{f,0}^{\text{int},n}$ of fluid nodes which are in the solid domain $\Omega_s^{t_n}$ and that pertain to elements which are crossed by the solid interface $\partial\Omega_s^{t_n}$ given by the condition $\phi_{f,0}^n = 0$, and determine the distance $\mathbf{d}_{f,0}^n$ of these nodes to such interface;
      iii. move the mesh solving a Laplace problem in $\mathcal{T}_f^0$ with Dirichlet boundary conditions for the nodes in $\mathcal{N}_{f,0}^{\text{int},n}$ equal to $\mathbf{d}_{f,0}^n$, obtaining the mesh $\mathcal{T}_f^{t_n}$;
   c. interpolate linearly the fluid velocity field from the previous body-fitted mesh $\bar{\mathcal{T}}_f^{t_n}$ onto the current body-fitted mesh $\mathcal{T}_f^{t_n}$;
   d. analyze the mesh quality of the mesh $\mathcal{T}_f^{t_n}$;
   e. for the solid domain $\Omega_s^{t_n}$ compute the level set function ϕ_f^n in $\mathcal{T}_f^{t_n}$, this function will be used as indicator of the solid position, and is not computed again until this very Step 1 takes place again;
   f. do $(\cdot)^k = (\cdot)^n$ as the initial guess for the fluid-structure iterations.
2. Start of a single fluid-structure iteration (go through Steps 3 to 8).
3. Compute the fluid mesh velocity $\mathbf{v}_m^k$ as follows

a. build the set of nodes $\mathcal{N}_f^k$ of fluid nodes in the solid region $\Omega_s^{t_{n+1},k}$ in order to compute the level set ϕ_f^k (not needed in the first iteration) and retrieve the velocity of the solid body $\mathbf{v}_{s,j}^k = \mathbf{v}_s^k(\mathbf{x}_j), \forall j \in \mathcal{N}_f^k$ from the solid mesh $\mathcal{T}_s^{t_{n+1},k}$;

b. compute an extension for the mesh velocity $\mathbf{v}_m^k$ solving a Laplace problem in $\mathcal{T}_f^{t_{n+1},k}$ with Dirichlet boundary conditions for the nodes in $\mathcal{N}_f^k$ equal to $\mathbf{v}_{s,j}^k, \forall j \in \mathcal{N}_f^k$.

4. Compute $(\mathbf{v}_f^{k+1}, p_f^{k+1}, \boldsymbol{\psi}^{k+1})$ solving the fluid problem (10) in $\mathcal{T}_f^{t_{n+1},k}$, that is, solve the Navier-Stokes equations in ALE form with a constraint over the set of nodes $\mathcal{N}_f^k$ such that the velocity of those nodes is the velocity of the mesh, that is $\mathbf{v}_f^{k+1}(\mathbf{x}_j) = \mathbf{v}_{s,j}^k, \forall k \in \mathcal{N}_f^k$.

5. From the fluid mesh $\mathcal{T}_f^{t_{n+1},k}$ retrieve $\sum_{j \in \mathcal{N}_f^k} \boldsymbol{\psi}_j^{k+1}$ to be given to the solid problem.

6. Solve the solid problem (12), computing $\mathbf{v}_s^{o,k+1}$ and $\boldsymbol{\Omega}_s^{o,k+1}$, and transfer the solid velocity and position to the solid mesh $\mathcal{T}_s^{t_{n+1},k}$, also compute the corresponding displacement $\mathbf{u}_s^{k+1}$ in each point of the solid mesh such that from $\mathcal{T}_s^{t_{n+1},k}$, that is from $\Omega_s^{t_{n+1},k}$, we get the mesh at the next iteration $\mathcal{T}_s^{t_{n+1},k+1}$, that is $\Omega_s^{t_{n+1},k+1}$.

7. With the solid displacement $\mathbf{u}_s^{k+1}$ computed in Step 6 move the fluid mesh solving a Laplace problem in $\mathcal{T}_f^{t_{n+1},k}$ with Dirichlet boundary conditions equal to $\mathbf{u}_f^{k+1}(\mathbf{x}_j) = \mathbf{u}_s^{k+1}(\mathbf{x}_j)$, $j \in \mathcal{N}_f^k$, such that we get here $\mathcal{T}_f^{t_{n+1},k+1}$.

8. Check convergence using the solid degrees of freedom $\mathbf{v}_s^o$ and $\boldsymbol{\Omega}_s^o$

a. if convergence was achieved then do $(\cdot)^{n+1} = (\cdot)^{k+1}$ and go to Step 9;

b. otherwise compute a proper relaxation parameter ω^k, do $(\cdot)^k = \omega^k(\cdot)^{k+1} + (1-\omega^k)(\cdot)^k$ and go to Step 2.

9. Check mesh reset and readjust condition

a. if it is not necessary to reset and readjust then do $(\cdot)^k = (\cdot)^n = (\cdot)^{n+1}$ and go to Step 2;

b. if it is necessary to reset and readjust the go to Step 1.

For this algorithm some remarks are in order.

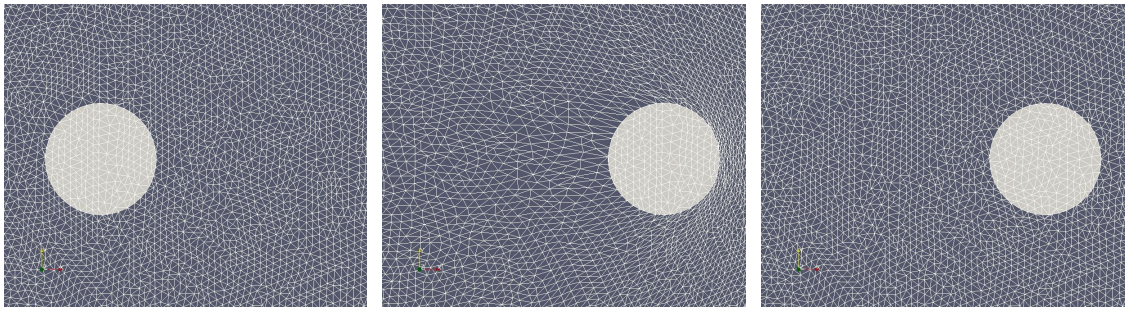
- Index 0 in Step 1 denotes quantities defined in the reference mesh $\mathcal{T}_{f,0}^{t_n}$
- In Step 1 two level set functions play a role: $\phi_{f,0}^n$ is the level set corresponding to the solid domain $\Omega_s^{t_n}$ with respect to the reference mesh $\mathcal{T}_f^0$, while ϕ_f^n is the level set for the same solid domain with respect to the body-fitted mesh $\mathcal{T}_f^{t_n}$.
- The adjustment procedure in Step 1b does not ensure a good quality of the mesh $\mathcal{T}_f^{t_n}$. Some elements inside the solid region $\Omega_s^{t_n}$, near the boundary $\partial\Omega_s^{t_n}$, may be distorted. This is not an issue indeed, since the velocity field is going to be constrained over the nodes pertaining to those elements. However, we consider this an open issue that should be further investigated.
- Step 3b is necessary in order to propagate the mesh velocity to the entire fluid domain to be used by the ALE formulation.

- The solid mesh $\mathcal{T}_s^{t_{n+1}}$ present in Steps 3a and 6 is necessary for the sake of generality in the algorithm. It is not strictly necessary for the case of rigid bodies, but it will be necessary for deformable bodies.

- In Step 8 the relaxation parameter is computed using Aitken acceleration as in Borazjani et al. (2008).

- The reset condition in Step 9 may based on a given criterion upon mesh distortion or may be fixed, that is, after a certain number of time steps we reset the mesh. In the present work we employ the second alternative.

In order to further exemplify the reset and adjustment implied by Step 1 we present Figure 1. In Figure 1(a) we observe the position of the solid in a given time t_n step for which the fluid mesh is denoted by $\mathcal{T}_f^{t_n}$. Then, Figure 1(b) shows the fluid mesh after m (several) time steps, which is denoted by $\mathcal{T}_f^{t_{n+m}}$. Here the fluid mesh accompanied the solid movement from t_n to t_{n+m} (no reset nor readjustment was performed in that period of time). Now this mesh is going to be the instance which is previous to the mesh reset and readjustment, so we change names $\bar{\mathcal{T}}_f^{t_{n+m}} = \mathcal{T}_f^{t_{n+m}}$. Finally, Figure 1(c) presents the new body-fitted mesh $\mathcal{T}_f^{t_{n+m}}$ after reset and readjustment of the reference mesh $\mathcal{T}_f^0$. Thus, notice that we need to interpolate information from $\bar{\mathcal{T}}_f^{t_{n+m}}$ onto $\mathcal{T}_f^{t_{n+m}}$ according to Step 1c.



(a) Body-fitted mesh obtained from the reference mesh.

(b) Deformed body-fitted mesh after some time steps.

(c) New boddy-fitted mesh obtained from the reference mesh after performing the reset and readjustment step.

Figure 1: Reset and adjustment of the mesh to fit solid boundaries.

3.2 Remarks on the parallel computational implementation

The entire implementation of this algorithm has been caried out on the flexible PAR-GPFEP system for generating high performance finite element softwareBuscaglia et al. (1999b,a), in such a way that we are able to make use of parallel computing.

The solid mesh, and therefore all the fields related to the solid body are in all the processors, so the searches in the solid mesh of information needed by the fluid mesh can be carried out independently in each processor, in a completely parallel manner.

The PAR-GPFEP system provides means to access all the unknown fields at the elemental (gauss-point) level, in order to build the elemental matrices at each sub-step and manages the mesh distribution between different processes so that each process deals with a part of the global mesh, which we call the local mesh. The main difficulty in our approach is in Step 1c, where we

need to transfer information from one mesh into another. Depending on the mesh displacements it is common that a value required in a processor needs to be searched in the mesh assigned to another one, thus generating a communication operation.

In order to avoid interruptions and minimize the parallel overhead, the interpolation step was implemented as a collective operation involving the following steps:

i. each processor has the position of the local nodes and searches this position in the local mesh, building a list with the positions not found in the local mesh;

ii. this lists of points not found in each process is broadcasted to all processes (an all-gather collective operation);

iii. each processor searches in the local mesh the positions required by other processes;

iv. the results of this search is sent to the requiring process (in practice, due to the mesh distribution method, with nodes uniquely assigned and elements assigned to more than one process, there can be multiply found points, and this step becomes a collective reduce operation).

The geometric searches in Steps 1c, 3a and 7, are performed with an auxiliary structure, (a quad-tree in 2D, an octree in 3D) and a pre-classification stage that allows for a very fast search (see Buscaglia and Dari (1992); Dari and Vénere (1991) for details).

The PAR-GPFEP system also manages the parallel assembly of the global systems derived from both the Laplace problems and the Navier-Stokes equations, and their parallel solution.

4 PRELIMINARY RESULTS

In all the cases the physical quantities are dimensionless.

4.1 Example 1: falling of a 2D ellipse

In this first case we release an ellipse in a stagnated fluid and it falls due to the gravity force. The configuration of the problem is given in Figure 2. According to that figure the dimensions are $(H, L) = (10, 13)$, $(h, l) = (4, 9)$, $a = 1$, $b = 0.15$, $\alpha = 20°$. The fluid and solid densities are $\rho_f = 1$ and $\rho_s = 2.678$, respectively, while the fluid viscosity is $\mu = 0.00462$. The gravity force is given by a gravity field $\mathbf{f} = -9.8\mathbf{e}_y$, where $\mathbf{e}_y$ is the vertical direction. The fluid mesh is composed with 208921 nodes, and the solid mesh with 1320 nodes. The time step used here is $\Delta t = 0.0005$ and the tolerance for the fluid-structure iterations is $\varepsilon = 0.01$. The number time steps between mesh resets and readjustments is 200.

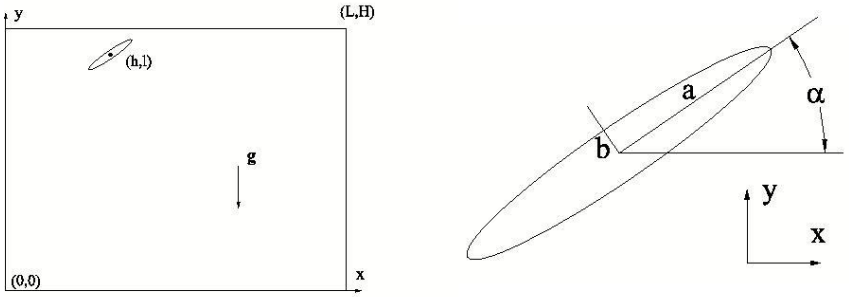


Figure 2: Ellipse released in a stagnated fluid.

In Figure 3 we summarize the results obtained for the falling ellipse. There the trajectory of the ellipse can be appreciated as well as the velocity map throughout the simulation, see top and bottom left figures, respectively. The rotation of the ellipse and the number of iterations to reach convergence are also presented in the same figure, see top and bottom right figures, respectively.

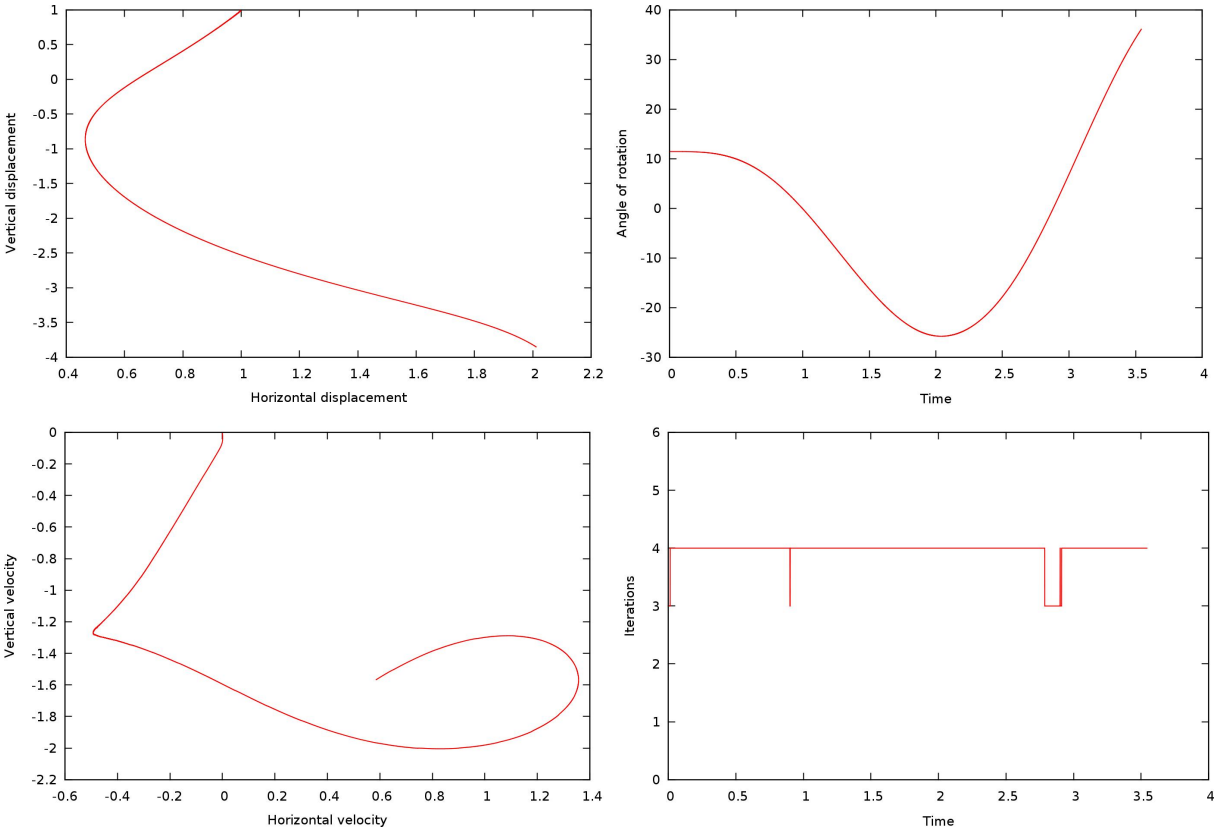


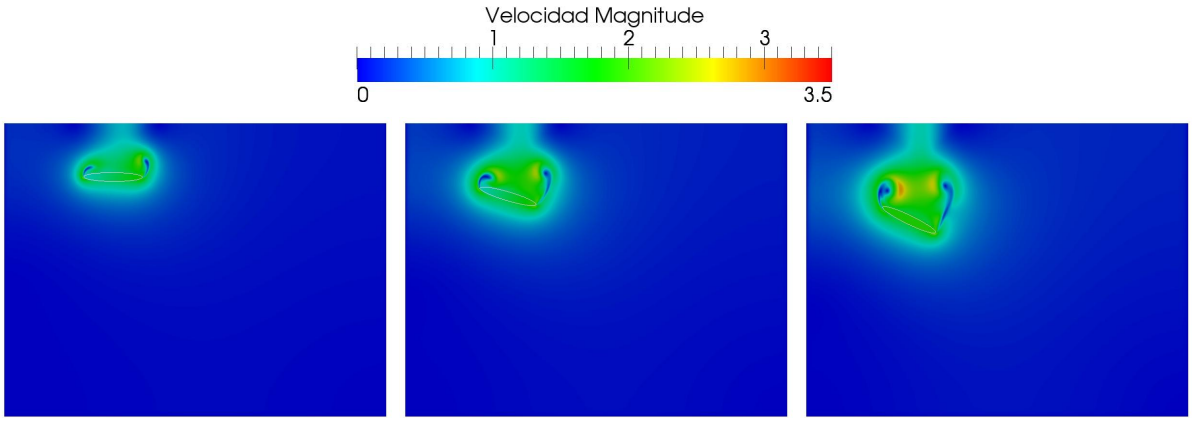
Figure 3: Results for the falling ellipse.

In Figure 4 we display some snapshots showing the magnitude of the velocity field as the ellipse falls.

In this case we analyzed the performance of the computational implementation. It must be said that in all tested cases the computational cost has been dominated by the solution of the Navier-Stokes equations. Table 1 summarizes the performance up to 240 processors.

Number of processors	Total	Speed-up	Assembling	Speed-up assembling	Solution	Speed-up solution
1	1747.84	1	39.49	1	1708.26	1
8	784.21	2.23	6.8	5.81	777.38	2.20
16	397.46	4.40	3.49	11.32	393.93	4.34
24	260.95	6.70	2.39	16.52	258.54	6.61
32	211.45	8.27	1.83	21.58	209.61	8.15
48	143.02	12.22	1.29	30.61	141.72	12.05
64	105.56	16.56	0.96	41.14	104.57	16.34
80	86.94	20.10	0.77	51.29	86.14	19.83
128	53.73	32.53	0.52	75.94	53.19	32.12
160	41.41	42.21	0.44	89.75	40.95	41.72
200	33.25	52.57	0.37	106.73	32.87	51.97
240	24.82	70.42	0.3	131.63	24.49	69.75

Table 1: Performance measured for a reference time step. Times are in seconds.



(a) $t = 1.0$ (b) $t = 1.5$ (c) $t = 2.0$

(d) $t = 2.5$ (e) $t = 3.0$ (f) $t = 3.5$

Figure 4: Magnitude of the velocity field at different time instants.

It can be seen, also from Figure 5 that the speed-up is linear up to 240 processors. To run this simulations we employed a cluster with Sun Blade x6250 modules and InifiBand conection with operating system CentOS 5.3.

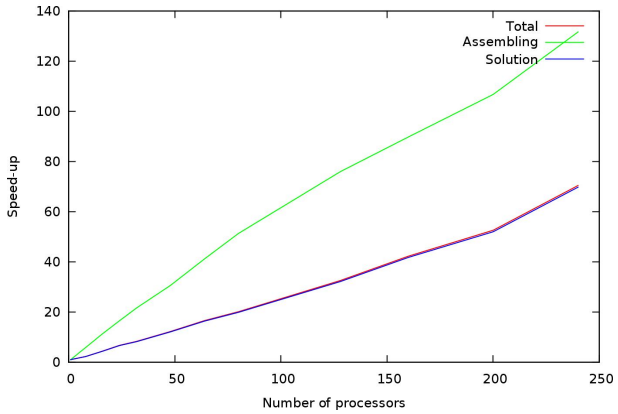


Figure 5: Speed-up up to 240 processors.

4.2 Example 2: lifting a 3D sphere

In this case the configuration of the problem is given in Figure 6. A sphere of radius $R = 0.5$ is released in a truncated conical domain. The equilibrium is reached when the lift and the drag provided by the flow balance the gravity force. Over the inferior flat surface there is an inlet on which we impose a constant Dirichlet boundary condition (parabolic profile of unitary

maximum velocity magnitude, that is $\mathbf{v}_{\max} = 1$). Over the superior flat surface homogeneous Neumann boundary conditions are imposed. The lateral wall is a no-slip surface. The density of the fluid and solid are $\rho_f = 1$ and $\rho_s = 2.5$, respectively, and the viscosity of the fluid is $\mu = 0.05$. The sphere is lifted by the flow against the gravity force, which is $\mathbf{f} = -9.81\mathbf{e}_z$, being $\mathbf{e}_z$ the longitudinal axis of the cone. The number off time steps between two consecutive mesh resets and readjustments iss 20. The tolerance for the convergence of the fluid-structure interaction problem is $\epsilon = 0.01$. The time step here is $\Delta t = 0.01$ and the Reynolds number reaches a maximum value $\mathrm{Re} = 20$. The fluid mesh consists of 170357 nodes and the solid mesh of 2136 nodes.

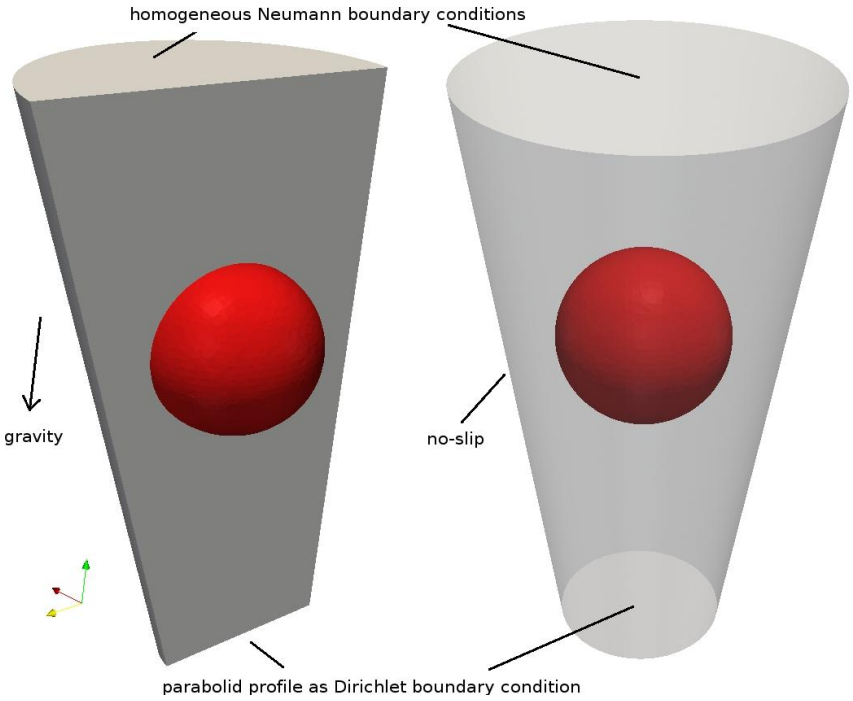


Figure 6: Sphere in a conic fluid domain with inflow boundary conditions.

Figure 7 shows the displacement of the sphere and the number of iterations to converge at each time step. The steady state is almost reached after 500 time steps, for which the lift and the drag provided by the flow balance the gravity force due to the difference between fluid and solid densities.

In Figure 8 the magnitude of the velocity field is displayed for several time instants throughout the simulation. We can observe that the accurate representation of the solid boundaries allows us to get rid of the roughness boundary effect seen in clasical immersed approaches.

In Figure 9 the reset and readjustment procedures are shown. At $t = 0.4$ there is a reset and readjustment step, so from the reference mesh $\mathcal{T}_f^0$ we get a body fitted mesh $\mathcal{T}_f^{t_{40}}$. This mesh will be attached to the solid movement at all times and will move with it until reaching the next time step in which we perform another reset and readjustment step. So, after 20 time steps we get the body fitted mesh $\mathcal{T}_f^{t_{59}}$. Then we go back to the reference mesh $\mathcal{T}_f^0$ (mesh reset) and readjust the mesh fitting the body again at time $t = 0.6$, obtaining the mesh $\mathcal{T}_f^{t_{60}}$.

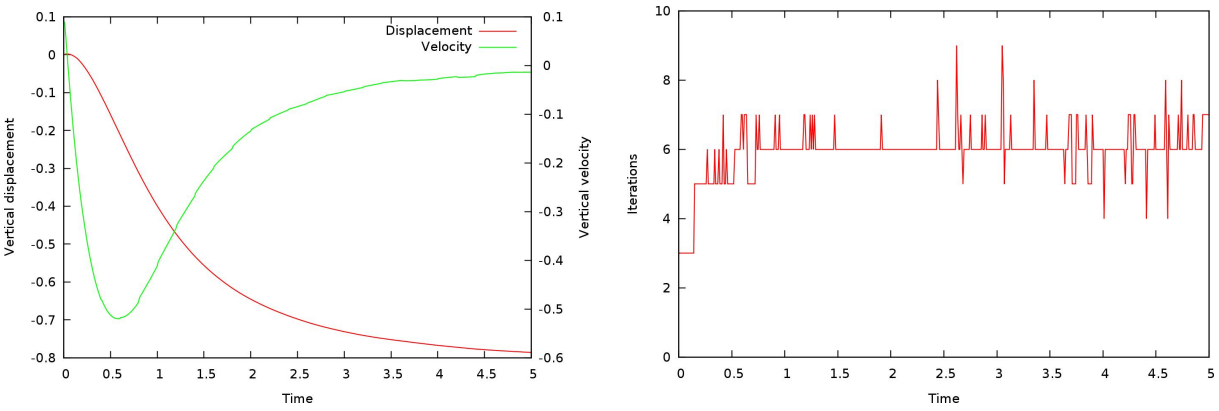
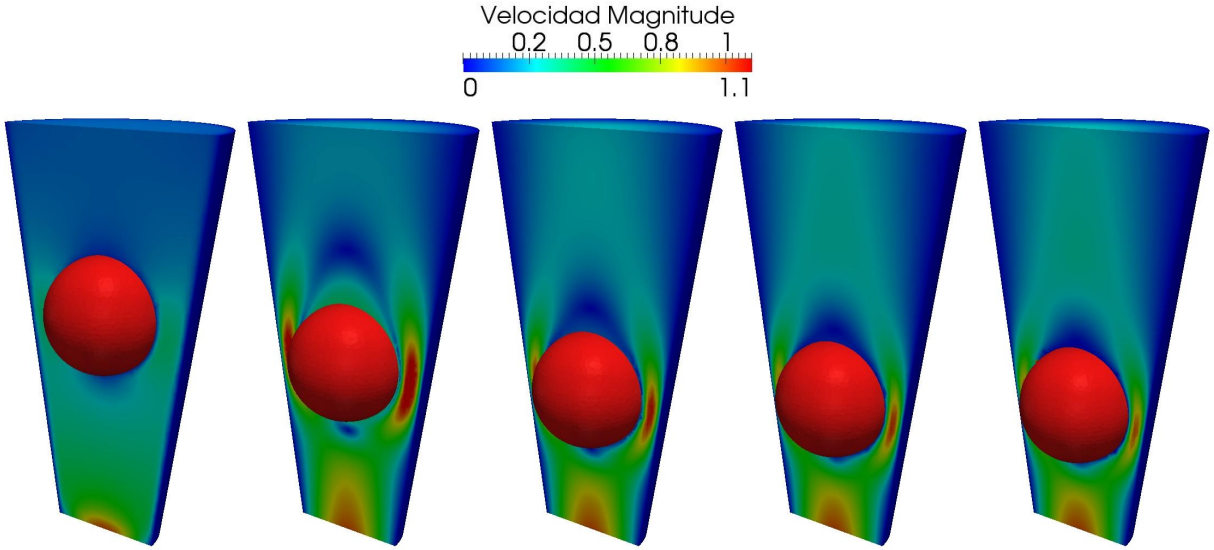


Figure 7: Vertical displacement and velocity in the falling sphere and nuber of iterations to reach convergence.



(a) $t = 0.0$. (b) $t = 1.0$. (c) $t = 2.0$. (d) $t = 3.0$. (e) $t = 5.0$.

Figure 8: Magnitude of the velocity field for several time instants.

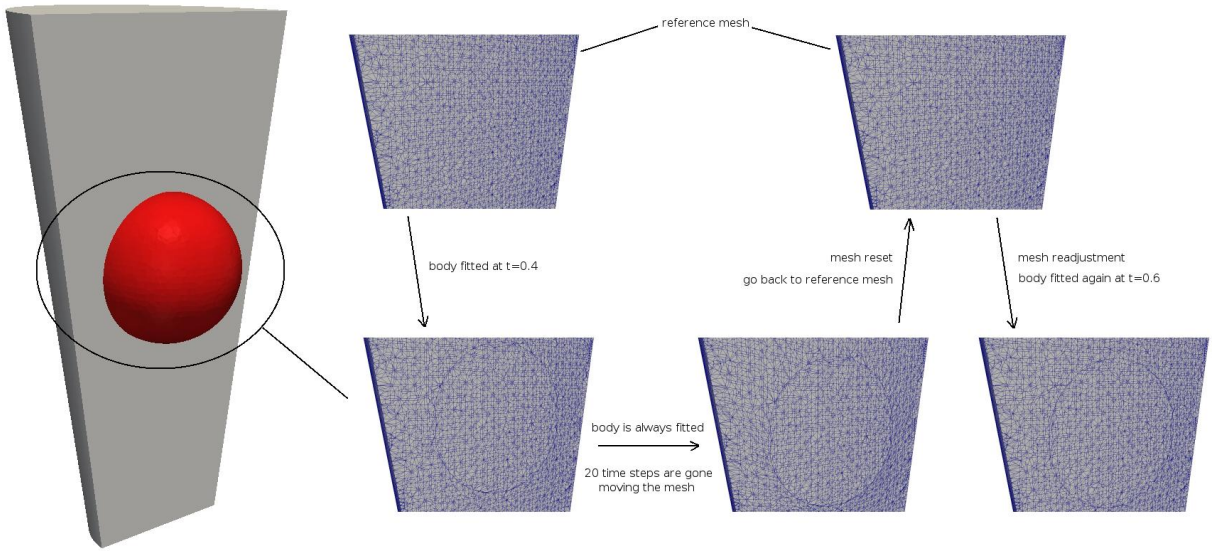


Figure 9: Reset and readjusment procedure to fit solid boundaries.

5 FINAL REMARKS

In this work an hybrid immersed/body-fitted approach for the fluid-structure interaction problem was presented and tested in the case of rigid bodies. Two advantages of this approach must be pointed out in contrast to the algorithms proposed in the literature. Firstly, the solid boudaries are represented accurately at all time because the strategy is, at all times body-fitted. Secondly, the topology of the mesh does not change throughout the simulation. The price to be paid is the set of auxiliary problems that have to be solved and searches that have to be made in order to transfer the information correctly among the different meshes that play a role in the computation. Nevertheless, when efficiently implemented like in the present case the computational cost is dominated by the Navier-Stokes problem.

The computational implementation was carried out for using parallel computing. The performance of the implementation is quite satisfactory in view of the behavior of the speed-up as a function of the number of processors.

Some open issues are to be taken into account in the future work such as the adjustment procedure by which the reference mesh fits the body, the interpolation of the velocity field between different meshes and also the extension to deformable solids.

6 ACKNOWLEDGMENTS

The first author acknowledges the support of the Argentine agency CONICET. The second and third authors acknowledge the financial support of the Brazilian agencies FAPERJ and CNPq.

REFERENCES

Baaijens F. A fictitious domain/mortar element method for fluid-structure interaction. *Int. J. Num. Meth. Fluids*, 35:743–761, 2001.

Blanco P., Feijóo R., and Dari E. A variational framework for fluid–solid interaction problems based on immersed domains. Theoretical bases. *Comp. Meth. Appl. Mech. Engrg.*, 197:2353–2371, 2008.

Borazjani I., Ge L., and Sotiropoulos F. Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies. 227:7587–7620, 2008. J. Comput. Phys.

Buscaglia G. and Dari E. Implementation of the Lagrange-Galerkin method for the incompressible navier-stokes equations. 15:23–36, 1992. Int. J. Num. Meth. Fluids.

Buscaglia G., Dari E., Lew A., and Raschi M. Un programa general de elementos finitos en paralelo. In *VI Congreso Argentino de MecÃąnica Computacional, MECANICA COMPUTACIONAL, Vol. XIX N.15*, pages 845–854. Mendoza, 1999a.

Buscaglia G., Ferrari H., Carrica P., and Dari E. An application of distributed computing to the finite element investigation of lift force on a freely-rotating sphere in simple shear flow. In *Proceedings of the ASME Fluids Engineering Division, The American Society of Mechanical Engineers, volume FED-Vol.250, D. Stock, editor*, pages 265–271. 1999b.

Dari E. and Vénere M. Algoritmos eficientes para la búsqueda del elemento de una red que contiene un punto dado. In *VII Encuentro Nacional de Investigadores y Usuarios del MÃl'todo de Elementos Finitos, MECANICA COMPUTACIONAL, Vol. X N.10*, pages 455–464. Mar del Plata, 1991.

Deparis S., Discacciati M., Fourestey G., and Quarteroni A. Fluid-structure algorithms based on Steklov-Poincaré operators. *Comp. Meth. Appl. Mech. Engrg.*, 195:5797–5812, 2006.

Glowinski R., Pan T.W., Hesla T., and Joseph D. A distributed Lagrange multiplier/fictitious domain method for particulate flows. *Int. J. Multiphase Flow*, 25:755–794, 1999a.

Glowinski R., Pan T.W., Hesla T., Joseph D., and Périaux J. A distributed Lagrange multiplier/fictitious domain method for flows around moving rigid bodies: application to particulate flow. *Int. J. Num. Meth. Fluids*, 30:1043–1066, 1999b.

Glowinski R., Pan T.W., and Périaux J. Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies. *Comp. Meth. Appl. Mech. Engrg.*, 151:181–194, 1998.

Hart J.D., Peters G., Schreurs P., and Baaijens F. A three–dimensional computational analysis of fluid–structure interaction in the aortic valve. *J. Biomech.*, 36:103–112, 2003.

Heil M. An efficient solver for the fully coupled solution of large–displacement fluid-structure interaction problems. *Comp. Meth. Appl. Mech. Engrg.*, 193:1–23, 2004.

Lew A. and Buscaglia G. A discontinuous–Galerkin–based immersed boundary method. 76:427–454, 2008. Int. J. Num. Meth. Engng.

Liu W., Kim D., and Tang S. Mathematical foundations of the immersed finite element method. *Comput. Mech.*, 39:211–222, 2007.

Liu W., Liu Y., Farrell D., Zhang L., Wang X., Fukui Y., Patankar N., Zhang Y., Bajaj C., Lee J., Hong J., Chen X., and Hsu H. Immersed finite element method and its applications to biological systems. *Comp. Meth. Appl. Mech. Engrg.*, 195:1722–1749, 2006.

Löhner R., Baum J., Mestreau E., Sharov D., Charman C., and Pelessone D. Adaptive embedded unstructured grid methods. *Int. J. Num. Meth. Engng.*, 60:641–660, 2004.

Matthies H. and Steindorf J. Partitioned strong coupling algorithms for fluid-structure interaction. *Comput. Struct.*, 81:805–812, 2003.

Patankar N., Singh P., Joseph D., Glowinski R., and Pan T.W. A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows. *Int. J. Multiphase Flow*, 26:1509–1524, 2000.

Peskin C. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002.

Roma A., Peskin C., and Berger M. An adaptive version of the immersed boundary method. *J. Comput. Phys.*, 153:509–534, 1999.

Tallec P.L., Gerbeau J.F., Hauret P., and Vidrascu M. Fluid structure interaction problems in large deformation. *Comptes Rendus Mecanique*, 333:910–922, 2005.

Tallec P.L. and Mouro J. Fluid structure interaction with large structural displacements. *Comp. Meth. Appl. Mech. Engrg.*, 190:3039–3067, 2001.

Wang X. From immersed boundary method to immersed continuum method. *Int. J. Multiscale Comp. Engng.*, 4:127–145, 2006.

Wang X. and Liu W. Extended immersed boundary method using FEM and RKPM. *Comp. Meth. Appl. Mech. Engrg.*, 193:1305–1321, 2004.

Zhang L., Gerstenberger A., Wang X., and Liu W. Immersed finite element method. *Comp. Meth. Appl. Mech. Engrg.*, 193:2051–2067, 2004.