Monolithic stabilized finite element method for rigid body motions in the incompressible Navier-Stokes flow

Monolothic SFEM for FSI

Stephanie Feghali — Elie Hachem — Thierry Coupez

Center for Material Forming (CEMEF)
MINES-ParisTech (UMR - CNRS)
1 rue Claude Daunesse
BP 207, F-06904 Sophia-Antipolis cedex
{stephanie.el_feghali, elie.hachem, thierry.coupez}@mines-paristech.fr

ABSTRACT. *We propose a new immersed volume method for solving rigid body motions in the incompressible Navier-Stokes flow. The used monolithic formulation gives rise to an extra stress tensor in the Navier-Stokes equations coming from the presence of the structure in the fluid. The system is solved using a finite element variational multiscale (VMS) method, which consists in here of a decomposition for both the velocity and the pressure fields into coarse/resolved scales and fine/unresolved scales. The distinctive feature of the proposed approach resides in the efficient enrichment of the extra constraint. We assess the behaviour and accuracy of the proposed formulation in the simulation of 2D and 3D examples.*

RÉSUMÉ. *On propose une méthode d'immersion de volume pour résoudre le mouvement de corps rigides dans un écoulement fluide. On utilise une formulation monolithique basée sur l'utilisation d'un tenseur de contrainte supplémentaire dans les équations de Navier-Stokes provenant de la présence de la structure dans le fluide. Le système est résolu en utilisant une formulation variationnelle multi-échelle (VMS), qui consiste en une décomposition de vitesse et pression en grandes échelles (résolues) et petites échelles (non résolues). La particularité de l'approche proposée réside dans l'enrichissement de l'extra-contrainte. Nous évaluons le comportement et la précision de la formulation proposée par des exemples en 2D et 3D.*

KEYWORDS: *monolothic stabilized finite elements, rigid body motions, incompressible flows, fluid Structure interactions, unstructured mesh.*

MOTS-CLÉS : *éléments finis stabilisés, mouvements de corps rigide, écoulement incompressible, interaction fluide structure, maillage non structuré.*

DOI:10.3166/EJCM.19.547-573 © 2010 Lavoisier, Paris

1. Introduction

Numerical simulations of fluid-structure interaction (FSI) are of first interest in numerous industrial problems (aeronautics, heat treatments, aerodynamic, bioengineering...). Because of the high complexity of such problems, analytical study are in general not sufficient to understand and solve them. Additionally, in spite of the available computers performance and the actual maturity of computational fluid dynamics, several key issues in the domain of computational FSI still prevent simulations from really helping in solving academic and industrial problems. FSI simulations are then nowadays the focus of numerous investigations, and various approaches are proposed to treat them.

Most of the commercial software packages solve FSI problems using an Arbitrary Lagrangian Eulerian (ALE) formulation (Hirt *et al.*, 1974; Hughes *et al.*, 1981; Donea *et al.*, 1982). The solid domain is treated with a Lagrangian formulation. The nodes belonging to the interface between the solid and the fluid are moved with the solid. The displacement of the nodes in the fluid domain do not depend on the fluid motion, but only ensures the continuity between the fluid and the solid domain, and a good mesh quality. ALE methods are robust and accurate, and do not need any extra degrees of freedom. However, important problems arise if the deformations, displacements and rotations of the solid becomes very important (Benson, 1989; Souli *et al.*, 2000; van Loon *et al.*, 2007).

A higher popularity has been gained recently by partitioned approaches which allow the use specific solver for each domain. The difficulty remains in transfering the information between the codes. The coupling between the two phases can be enforced using different schemes : weakly or strongly coupled version. The former approach manages with just one solution of either field per time step but consequently lack accurate fulfilment of the coupling conditions. The latter requires sub-iterations. The predominant approach consists in solving the problem iteratively, using fixed-point schemes (Le Tallec *et al.*, 2001) or Newton Krylov methods (Fernández *et al.*, 2005; Gerbeau *et al.*, 2003; Michler *et al.*, 2005; Gerbeau *et al.*, 2005). Actually, the fixed-point methods with dynamic relaxation seem to be the most interesting variant (Küttler *et al.*, 2008). This approach allows the use of fluid and solid solvers for each of the two phases. It is accurate and quite efficient but present an inherent instability depending on the ratio of the densities and the geometry of the domain (Caussin *et al.*, 2005). As a result, the numerical cost increase drastically and coupling algorithms may not converge. For 3D problems, such difficulties become even more severe.

Monolithic approaches have been proposed to overcome these drawbacks. The whole domain (composed by fluid and solid phases) is considered as a single one, meshed by a single grid, and solved with an Eulerian framework. The continuity at the interface is then obtained naturally and there is no need to enforce it, as it was the case in partitioned methods. If the multi-mesh approaches permit the use of classical fluid and solid solvers, monolithic approaches impose the use of an appropriated unique constitutive equation describing both the fluid and the solid domain. Interface tracking, between the two different domains, can be completed by Immersed Boundary (IB) methods (Peskin, 2002) where the interface is convected on a Lagrangian way. Other methods such as the fictitious domain (Peskin, 2002; Glowinski *et al.*, 1999) treat the coupling between the domains by applying a constraints across the rigid body using a Lagrange multiplier.

Here in this work, we use the immersed volume method based on the level-set approach (Hachem, 2009). Mesh adaptation, and particularly anisotropic mesh adaptation is used to track the fluid-solid interface with a good precision and reasonable computational time (Laure *et al.*, 2007; Hachem, 2009).

The rigid solid is treated using the Navier-Stokes solver under constrains to impose the nullity of the deformations. It can be done by simply penalizing the strain rate using a very important viscosity in the solid, which can sometimes be sufficient (Coupez *et al.*, 2010; Ritz *et al.*, 1999; Hachem, 2009). It is also possible to enforce directly the nullity of the strain by using an Augmented Lagrange Mulptipliers method (Glowinski *et al.*, 2001; Coquerelle *et al.*, 2008; Janela *et al.*, 2005), solved by an iterative Uzawa algorithm. The problem is solved by adding an extra-stress tensor comming from the presence of the structure in the fluid. Linear or harmonic mixture laws of the mechanical properties characterizing each domain are then applied at the interface.

However, in the presence of high discontinuity materials properties at the interface between the fluid and solid, numerical oscillations appear at the interface and pollute the global solutions. Therefore, we propose an extension of the multiscale-stabilized Navier-Stokes solver (Hachem, 2009; Hachem *et al.*, 2010) taking into account the solid constraint as an extra tensor. Recall that the classical finite element approximation for the flow problem must verifies two criterias : the compatibility condition known by the inf-sup or 'Brezzi-Babuska' condition which required an appropriate pair of the function spaces for the velocity and the pressure (Franca *et al.*, 1988; Codina *et al.*, 2008; Codina, 2001) and the stablility in the convection dominated flows (Hachem *et al.*, 2010; Codina, 2000). For these reasons, a robust monolithic Navier-Stokes solver is extended with a stabilization procedure for rigid body motion. It uses a multiscale stablilized finite element method based on the enrichment of the functional spaces for the velocity, the pressure and the stress.

An outline of the paper is as follow. The Immersed Volume Method is introduced in Section 2. Section 3 is dedicated to present the strong and weak form of the equations of motion. The stabilizing schemes from a variational multiscale point of view is described in Section 4. Then, in Section 5, the numerical scheme is presented. Section 6 presents some benchmark problems to validate the method, and the numerical simulation of a falling disk in a channel is examined.

2. Immersed volume method

The immersed volume method is based on solving the single set of equations by differentiating the subdomains and refining the mesh at this interface using the level-set method. This section presents the complete description of the method, which is structured into three subsections : immerse the solid using level-set function, mix the physical properties and finally apply the anisotropic mesh adaptation at the vicinity of the interface (Hachem, 2009).

2.1. *Level-set approach, mixing laws*

A signed distance function of an interface Γ_i is used to localize the interface of the immersed body Ω_i and initialize the desirable properties on both sides of this latter. At any point **x** of the computational domain Ω, the level-set function α_i corresponds to the distance from Γ_i. In turn, the interface Γ_i is then given by the iso-zero of the function α_i :

$$\begin{cases} \alpha_i(\mathbf{x}) = \pm d(\mathbf{x}, \Gamma_i), \mathbf{x} \in \Omega, \\ \Gamma_i = \{\mathbf{x}, \alpha_i(\mathbf{x}) = 0\} \end{cases} \tag{1}$$

In this paper, a sign convention is used : $\alpha_i \geq 0$ inside the solid domain defined by the interface Γ_i and $\alpha_i \leq 0$ outside this domain. Further details about the algorithm used to compute the distance are available in (Bruchon *et al.*, 2009).

2.2. *Mixing laws*

The geometry and mechanical properties of each subdomain are characterized by one signed distance function. Once all the sub-domains are defined, the mechanical properties can then be determined on the whole domain in function of the level-set function. For the elements crossed by the level-set functions and the their neighbours, fluid-solid mixtures are used to determine the element effective properties.

A heaviside function $H(\alpha)$ for each level-set function is defined by :

$$H(\alpha) = \begin{cases} 1 & \text{if } \alpha > 0 \\ 0 & \text{if } \alpha < 0 \end{cases} \tag{2}$$

In order to achieve a better continuity at the interface (van der Pijl *et al.*, 2005), the heaviside function can be smoothed using :

$$H_\varepsilon(\alpha) = \begin{cases} 1 & \text{if } \alpha > \varepsilon \\ \dfrac{1}{2}\left(1 + \dfrac{\alpha}{\varepsilon} + \dfrac{1}{\pi}\sin\left(\dfrac{\pi\alpha}{\varepsilon}\right)\right) & \text{if } |\alpha| \leq \varepsilon \\ 0 & \text{if } \alpha < -\varepsilon \end{cases} \qquad [3]$$

where ε is a small parameter such that $\varepsilon = O(h_i)$, known as the interface thickness, and h_i is the mesh size in the normal direction to the interface. In the vicinity of the interface, it can be computed using the following expression :

$$h_i = \max_{j,l \in K} \boldsymbol{\nabla}\alpha \cdot \mathbf{x}^{jl} \qquad [4]$$

where $\mathbf{x}^{jl} = \mathbf{x}^l - \mathbf{x}^j$ and K is the mesh element.

According to the chosen approximations, the heaviside function is then approximated using linear interpolations $P1$ between fluid and solid properties or a piecewise constant interpolation $P0$.

2.3. *Anisotropic mesh adaptation*

Accurate calculation of the velocities, strains and stresses along the fluid-solid interface is critical for a correct modelling of industrial applications. The difficulty arises due to the discontinuity of the properties of the material across the interface. If this latter is not aligned with the element edges, it may intersect the element arbitrarily such that the accuracy of the finite element approach can be compromised. In order to circumvent this issue, the level-set process is thus coupled to an anisotropic mesh adaptation as described in (Gruau *et al.*, 2005). The idea of this method is to pre-adapt the mesh at the interface. The mesh becomes locally refined, elements are streched, which enables to sharply define the interface and to save a great number of elements compared to classical isotropic refinement. This anisotropic adaptation is performed by constructing a metric map that allows the mesh size to be imposed in the direction of the distance function gradient. Let us briefly described the main principles of this technique. First of all, one has to resort to a so-called metric which is a symmetric positive defined tensor representing a local base that modify the distance computation, such that :

$$||\mathbf{x}||_{\mathrm{M}} = \sqrt{^T\mathbf{x} \cdot \mathbb{M} \cdot \mathbf{x}}\,, \quad <\mathbf{x}, \mathbf{y}>_{\mathrm{M}} =^T \mathbf{x} \cdot \mathbb{M} \cdot \mathbf{y}\,. \qquad [5]$$

The metric $\mathbb{M}$ can be regarded as a tensor whose eigenvalues are related to the mesh sizes, and whose eigenvectors define the directions for which these sizes are applied. For instance, using the identity tensor, one recovers the usual distances and directions of the Euclidean space. In our case the direction of mesh refinement is given by the unit normal to the interface which corresponds to the gradient of the level-set function : $\mathbf{x} = \boldsymbol{\nabla}\alpha/||\boldsymbol{\nabla}\alpha||$. A default mesh size, or background mesh size, h_d is imposed far from the interface and it is reduced as the interface comes closer. A likely choice for the mesh size evolution is the following :

$$h = \begin{cases} h_d & \text{if } |\alpha(\mathbf{x})| > e/2 \\ \dfrac{2h_d(m-1)}{m\,e}|\alpha(\mathbf{x})| + \dfrac{h_d}{m} & \text{if } |\alpha(\mathbf{x})| \leq e/2 \end{cases} \qquad [6]$$

Eventually, at the interface, the mesh size is reduced by a factor m with respect to the default value h_d. Then this size increases until equalling h_d for a distance that corresponds to the half of a given thickness e. The unit normal to the interface $\mathbf{x}$ and the mesh size h defined above, lead to the following metric :

$$\mathbb{M} = C\,(\mathbf{x} \otimes \mathbf{x}) + \frac{1}{h_d}\,\mathbb{I} \quad \text{with} \quad C = \begin{cases} 0 & \text{if } |\alpha(\mathbf{x})| \geq e/2 \\ \dfrac{1}{h^2} - \dfrac{1}{h_d^2} & \text{if } |\alpha(\mathbf{x})| < e/2 \end{cases} \qquad [7]$$

where $\mathbb{I}$ is the identity tensor. This metric returns to isotropic far from the interface (with a mesh size equal to h_d for all directions) and to anisotropic near the interface (with a mesh size equal to h_i in the direction $\mathbf{x}$ and equal to h_d in the others). This method can be assisted by a posteriori anisotropic error estimator, the search of the optimal mesh (metric) that minimizes the error estimator. As a result, an optimal metric as a minimum of an error indicator function and for a given number of elements is obtained.

In practice, the mesh is generated in several steps using the MTC mesher and remesher developed by (Coupez, 2000). Further details on the anisotropic mesh generation can be found in (Gruau *et al.*, 2005). The proposed mesh generation algorithm works well for 2D or 3D complex shapes. It allows the creation of meshes with extremely anisotropic elements stretched along the interface. The mesh size is then only refined in the direction of the high physical and mechanical properties gradients. This allow both conserving a high precision in the calculus and in the geometry description, in spite of an important decrease of the total number of degrees of freedom. The grid is furthermore only modified in the vicinity of the interface which keeps the computational work devoted to the grid generation low. Note also that the proposed method can easily handle arbitrary complex geometries. As shown in Figure 1 which presents a close-up on the interface zone at the end of the anisotropic adaptation process, the mesh has been gradually refined when approaching the interface. Consequently, only additional nodes are locally added in this region, whereas the rest of domain keeps the same background size.

3. Governing equations

This section is devoted to the mathematical formulation for rigid body immersed in an incompressible fluid. The governing equations are considered to be three-

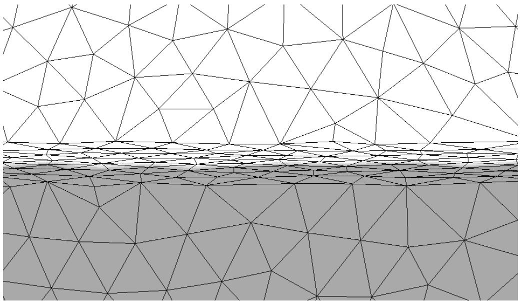


Figure 1. *Zoom on the fluid-solid interface after anisotropic adaptation*

dimensional, unsteady. As the proposed approach is monolithic, a unique constitutive equation will be solved on the whole domain, with a variation of the parameters depending on the phase that should be modelled.

First, each system of equation is presented seperately. Let $\Omega \subset \mathbb{R}^n$ be the spatial domain at time $t \in [0, T]$, where n is the space dimension. Let $\partial\Omega$ denote the boundary of Ω. Ω_f, Ω_s and Ω_{fsi} be respectively the fluid domain, the solid domain and the interface. They verify :

$$\Omega_f \cup \Omega_s = \Omega \quad \text{and} \quad \Omega_f \cap \Omega_s = \Omega_{fsi} \qquad [8]$$

The dynamic of the flow is given by the classical incompressible Navier-Stokes equations :

$$\rho_f \left(\partial_t \mathbf{v} + \mathbf{v} \cdot \boldsymbol{\nabla} \mathbf{v}\right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = \mathbf{f} \quad \text{in } \Omega_f \qquad [9]$$

$$\boldsymbol{\nabla} \cdot \mathbf{v} = 0 \quad \text{in } \Omega_f \qquad [10]$$

$$\mathbf{v} = \mathbf{v_\Gamma} \quad \text{on } \Gamma \qquad [11]$$

$$\mathbf{v} = \mathbf{v_i} \quad \text{on } \Omega_{fsi} \qquad [12]$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{d} \quad \text{on } \Omega_{fsi} \qquad [13]$$

$$\mathbf{v}(\mathbf{x}, \mathbf{0}) = \mathbf{v_0}(\mathbf{x}) \quad \text{in } \Omega_f \qquad [14]$$

where $\mathbf{v}$ is the velocity vector, ρ_f is the fluid density, , $\mathbf{n}$ is the outward normal on the solid surface, $\mathbf{v_i}$ is the velocity at fluid-solid interface Ω_{fsi} and $\mathbf{f}$ the applied force vector. For an incompressible fluid the divergence-free constraint [10] gives rise to the pressure in the fluid. The stress tensor for a newton fluid is then given by :

$$\boldsymbol{\sigma} = 2\eta_f\, \boldsymbol{\varepsilon}(\mathbf{v}) - p\, \mathbf{I_d} \qquad [15]$$

where p is the pressure, $\mathbf{I_d}$ is the identity tensor, $\boldsymbol{\varepsilon}(\mathbf{v}) = (\boldsymbol{\nabla}\mathbf{v} +^T \boldsymbol{\nabla}\mathbf{v})/2$ the deformation-rate tensor and η_f is the dynamic viscosity.

In the present formulation we treat the rigid body as a continuous domain subjected to an addictionnal rigidity constraint. The governing equations for particle motion are then given by :

$$\rho_s\, (\partial_t \mathbf{v} + \mathbf{v} \cdot \boldsymbol{\nabla}\mathbf{v}) - \boldsymbol{\nabla} \cdot \sigma = \mathbf{f} \quad \text{in } \Omega_s \qquad [16]$$

$$\boldsymbol{\nabla} \cdot \mathbf{v} = 0 \quad \text{in } \Omega_s \qquad [17]$$

$$\boldsymbol{\varepsilon}(\mathbf{v}) = 0 \quad \text{in } \Omega_s \qquad [18]$$

$$\mathbf{v} = \mathbf{v_i} \quad \text{on } \Omega_{fsi} \qquad [19]$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = -\mathbf{d} \quad \text{on } \Omega_{fsi} \qquad [20]$$

$$\mathbf{v}(\mathbf{x}, \mathbf{0}) = \mathbf{v_0}(\mathbf{x}) \quad \text{in } \Omega_s \qquad [21]$$

where ρ_s the solid density. In a rigid body there is no deformation $\boldsymbol{\varepsilon}(\mathbf{u}) = 0$ ($\mathbf{u}$ is the displacement field) and $\partial_t \mathbf{u} = \mathbf{v}$. These two equations implies a null value of the deformation-rate tensor [18]. From another part the rigidity constraint [18] ensure that the velocity field is a divergence-free. Hence Equation [17] is a redundant equation. Nevertheless we choose to keep this constraint to account the pressure term. As noted earlier Equation [17] gives rise to a pressure fluid. Similarly, Equation [18] gives rise to a stress field $\boldsymbol{\tau}$. The stress tensor is then given by :

$$\boldsymbol{\sigma} = -p\, \mathbf{I_d} + \boldsymbol{\tau}_s \qquad [22]$$

Adding both systems, the strong form for the whole domain reads :

$$\begin{cases} \rho\, (\partial_t \mathbf{v} + \mathbf{v} \cdot \boldsymbol{\nabla}\mathbf{v}) - \boldsymbol{\nabla} \cdot (2\eta\, \boldsymbol{\varepsilon}(\mathbf{v}) + \boldsymbol{\tau} - p\, \mathbf{I_d}) = \mathbf{f} & \\ \boldsymbol{\nabla} \cdot \mathbf{v} = 0 & \\ \boldsymbol{\varepsilon}_s(\mathbf{v}) = 0 & \\ \mathbf{v} = \mathbf{v_b} & \text{on } \partial\Omega \\ \mathbf{v}(\mathbf{x}, \mathbf{0}) = \mathbf{v_0}(\mathbf{x}) & \end{cases} \qquad [23]$$

where $\boldsymbol{\varepsilon}_s(\mathbf{v}) = H(\alpha)\boldsymbol{\varepsilon}(\mathbf{v})$, $\eta = (1 - H(\alpha))\eta_f$, $\rho = \rho_s H(\alpha) + \rho_f(1 - H(\alpha))$ and $\boldsymbol{\tau} = H(\alpha)\boldsymbol{\tau}_s$. Once the object is immersed inside the computational domain using the proposed technique, the equations at the fluid-solid interface are naturally satisfied. As a consequence, the boundary conditions [12]-[13]-[19]-[20] are no longer needed. The boundary velocity $\mathbf{v_b}$ results from Equation [11]. If the solid boundary intersect with the domain boundary, $\mathbf{v_b}$ can be easily adapted.

Weak form

Multiplying by the test functions and integrating by parts, the associated standard weak form of the system (23), can be stated as : Find $\mathbf{v} \in V = \left(H_0^1(\Omega)\right)^n$, $p \in Q = L^2(\Omega)$ and $\tau \in \mathcal{T} = L^2(\Omega)^{n\times n}$ such that :

$$\begin{cases} \rho\left(\partial_t \mathbf{v}, \mathbf{w}\right) + \rho\mathbf{v}\cdot\boldsymbol{\nabla}\mathbf{v}, \mathbf{w}) - (p\,\boldsymbol{\nabla}\cdot\mathbf{w}) + (2\eta\boldsymbol{\varepsilon}(\mathbf{v}) : \boldsymbol{\varepsilon}(\mathbf{w})) + (\boldsymbol{\tau} : \boldsymbol{\varepsilon}(\mathbf{w})) \\ \qquad\qquad = (\mathbf{f}, \mathbf{w}) \\ (\boldsymbol{\nabla}\mathbf{v}, q) = 0 \\ (\boldsymbol{\varepsilon}_s(\mathbf{v}) : \boldsymbol{\tau}^*) = 0 \end{cases} \qquad [24]$$

A penalization factor η_s will be added to this formulation, so that $\eta = H(\alpha)\eta_s + (1 - H(\alpha))\eta_f$. As a consequence an extra term is added $(2\eta_s\boldsymbol{\varepsilon}(\mathbf{v}) : \boldsymbol{\varepsilon}(\mathbf{w}))$. This can be done as long as $\boldsymbol{\varepsilon}(\mathbf{w})$ belongs to the functional space of $\boldsymbol{\tau}$.

4. Stabilized finite-element method (SFEM)

In this section, we describe briefly the Galerkin finite-element approximation and the corresponding stabilization method for the resulting discrete system of Equations [24]. Based on a mesh $\mathcal{K}_h$ of Ω into set of N_{el} elements K, the functional spaces for the velocity, the pressure and the stress are approached by the finite dimensional spaces spanned by V_h, P_h and $\mathcal{T}_h$. As it is well known, the stability of the discrete formulation depends on appropriate compatibility restrictions on the choice of the finite element spaces, as stated by the inf-sup condition. According to this, standard Galerkin mixed elements with continuous equal order linear/linear interpolation for the three fields are not stable. Lack of stability shows as uncontrollable oscillations that pollute the solution. Fortunately, the strictness of the inf-sup condition can be avoided by modifying the discrete variational form, for instance, by means of introducing appropriate numerical techniques that can provide the necessary stability to the desired choice of interpolation spaces. The objective of this work is precisely to present stabilization methods which allow the use of equal order continuous interpolations for velocites and pressures and piecewise constant interpolation for stressess. The basic idea of the sub-grid scale approach is to consider that the unknowns can be split in two components, *e.g.* coarse and fine, corresponding to different scales or levels of resolution (Hachem *et al.*, 2010). First, we solve the fine scale and then we replace their effect into the large scale. This means approximating the velocity, pressure, stress solution space as $V_h \oplus V'$, $P_h \oplus P'$ and $\mathcal{T}_h \oplus \mathcal{T}'$. To this end, $\mathbf{v}, p, \boldsymbol{\tau}$ will be approximated as :

$$\left\{\begin{array}{lcll} \mathbf{v} & = & \mathbf{v}_h + \mathbf{v}' & \in V_h \oplus V' \\ p & = & p_h + p' & \in Q_h \oplus Q' \\ \boldsymbol{\tau} & = & \boldsymbol{\tau_h} + \boldsymbol{\tau'} & \in \mathcal{T}_h \oplus \mathcal{T}' \end{array}\right. \quad [25]$$

Introducing the splitting, the system of equations (24) is : Find $(\mathbf{v}, p, \boldsymbol{\tau})$ such that $\forall\, \mathbf{w} \in V_h^0 \oplus V',\ \ q \in Q_h \oplus Q'$ et $\boldsymbol{\tau}^* \in \mathcal{T}_h \oplus \mathcal{T}'$

$$\left\{\begin{array}{l} \rho\,(\partial_t(\mathbf{v}_h + \mathbf{v}'), \mathbf{w}_h + \mathbf{w}') + \rho((\mathbf{v}_h + \mathbf{v}') \cdot \nabla(\mathbf{v}_h + \mathbf{v}'), \mathbf{w}_h + \mathbf{w}') \\ \quad -(p_h + p', \nabla \cdot (\mathbf{w}_h + \mathbf{w}')) + (2\eta\boldsymbol{\varepsilon}(\mathbf{v}_h + \mathbf{v}') : \boldsymbol{\varepsilon}(\mathbf{w}_h + \mathbf{w}')) \\ \quad +((\boldsymbol{\tau_h} + \boldsymbol{\tau'}) : \boldsymbol{\varepsilon}(\mathbf{w}_h + \mathbf{w}')) = (\mathbf{f}, \mathbf{w}_h + \mathbf{w}') \\ (\nabla \cdot (\mathbf{v}_h + \mathbf{v}'), q_h + q') = 0 \\ (\boldsymbol{\varepsilon}_s(\mathbf{v}_h + \mathbf{v}') : (\boldsymbol{\tau_h^*} + \boldsymbol{\tau_h^{*\prime}}))_{\Omega_s} = 0 \end{array}\right. \quad [26]$$

At this stage, three important remarks have to be made :

i) when using linear interpolation functions, the second derivatives vanish.

ii) the subscales are not tracked in time, therefore, quasi-static subscales are considered here ; however, the subscale equation remains quasi time-dependent.

iii) the convective velocity of the non-linear term may be approximated using only large-scale part so that $(\mathbf{v_h} + \mathbf{v}')\nabla \cdot (\mathbf{v_h} + \mathbf{v}') \simeq \mathbf{v_h} \cdot \nabla(\mathbf{v_h} + \mathbf{v}')$.

Consequently, the coarse scale problem reduces to the following :

$$\left\{\begin{array}{l} \rho(\partial_t \mathbf{v}_h, \mathbf{w}_h) + \rho(\mathbf{v}_h \cdot \nabla(\mathbf{v}_h + \mathbf{v}'), \mathbf{w}_h) - (p_h + p', \nabla \cdot \mathbf{w}_h) \\ \quad +(2\eta\boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\mathbf{w}_h)) + ((\boldsymbol{\tau_h} + \boldsymbol{\tau'}) : \boldsymbol{\varepsilon}(\mathbf{w}_h)) = (\mathbf{f}, \mathbf{w}_h) \\ (\nabla \cdot (\mathbf{v}_h + \mathbf{v}'), q_h) = 0 \\ (\boldsymbol{\varepsilon}_s(\mathbf{v}_h + \mathbf{v}') : \boldsymbol{\tau_h^*})_{\Omega_s} = 0 \end{array}\right. \quad [27]$$

and the fine scale problem :

$$\left\{\begin{array}{l} \mathbf{w}' + \rho(\mathbf{v}_h \cdot \nabla(\mathbf{v}_h + \mathbf{v}'), \mathbf{w}') - (p_h + p', \nabla \cdot \mathbf{w}') + (2\eta\boldsymbol{\varepsilon}(\mathbf{v}') : \varepsilon(\mathbf{w}')) \\ \quad +((\boldsymbol{\tau_h} + \boldsymbol{\tau'}) : \boldsymbol{\varepsilon}(\mathbf{w}')) = (\mathbf{f}, \mathbf{w}') \\ (\nabla \cdot (\mathbf{v}_h + \mathbf{v}'), q') = 0 \\ (\boldsymbol{\varepsilon}_s(\mathbf{v}_h + \mathbf{v}') : \boldsymbol{\tau^{*\prime}}) = 0 \end{array}\right. \tag{28}$$

Rearranging the terms of Equation [28] the fine scale problem reads to :

$$\left\{\begin{array}{l} \rho(\mathbf{v}_h \cdot \nabla\mathbf{v}', \mathbf{w}') + ((2\eta\boldsymbol{\varepsilon}(\mathbf{v}') : \boldsymbol{\varepsilon}(\mathbf{w}')) + (\boldsymbol{\tau'} : \boldsymbol{\varepsilon}(\mathbf{w}')) + (\nabla p', \mathbf{w}') = (\mathcal{R}, \mathbf{w}') \\ (\nabla \cdot \mathbf{v}', q') = -(\nabla \cdot \mathbf{v}_h, q') \\ (\boldsymbol{\varepsilon}_s(\mathbf{v}') : \tau^{*\prime}) = -(\boldsymbol{\varepsilon}_s(\mathbf{v}_h) : \boldsymbol{\tau^{*\prime}}) \end{array}\right. \tag{29}$$

with $\mathcal{R}$ the momentum residuals :

$$\mathcal{R} = \mathbf{f} - \rho\partial_t\mathbf{v}_h - \rho\mathbf{v}_h \cdot \nabla\mathbf{v}_h - \nabla p_h + \nabla \cdot \boldsymbol{\tau_h} \tag{30}$$

Following the lines in (Badia *et al.*, 2010) and using exactly the same procedure , it can be shown that $\mathbf{v}'$, p' and $\boldsymbol{\tau'}$ may be approximated within each element by :

$$\left\{\begin{array}{l} \mathbf{v}' = \sum_{K\in\mathcal{K}_h} \tau_k \tilde{P}_{\mathbf{v}}(\mathcal{R}) \\ p' = -\sum_{K\in\mathcal{K}_h} \tau_c \tilde{P}_p(\nabla \cdot \mathbf{v}) \\ \boldsymbol{\tau'} = -\sum_{K\in\mathcal{K}_h} \tau_t \tilde{P}_{\boldsymbol{\tau}}(\boldsymbol{\varepsilon}(\mathbf{v}_h)) \end{array}\right. \tag{31}$$

where the so called stabilization parameters τ_k, τ_c and τ_t can be computed as (Cereva *et al.*, 2010) :

$$\begin{aligned} \tau_k &= \left(\left(\frac{2}{\Delta t}\right)^2 + \left(\frac{4\eta}{\rho h^2}\right)^2 + \left(\frac{2 \parallel v \parallel_k}{h}\right)^2\right)^{-1/2}, \\ \tau_c &= \left(\left(\frac{\eta}{\rho}\right)^2 + \left(\frac{c_2 \parallel \mathbf{v_k} \parallel_\mathbf{k}}{c_1 h}\right)^2\right)^{1/2}, \\ \tau_t &= c_3\frac{h}{L}. \end{aligned} \tag{32}$$

The constants c_1, c_2 and c_3 are independant from h, h being the characteristic length of the element, L the characteristic lenght of the domain and $\parallel \mathbf{v} \parallel_{\mathbf{k}}$ the norm of the velocity in the center of the element. The simplest choice is to take $\tilde{P}_{\mathbf{v}}$, $\tilde{P}_p$ and $\tilde{P}_{\boldsymbol{\tau}}$ as the identity when applied to the residuals. As a hindrance of choosing a piecewise constant interpolation for $\boldsymbol{\tau_h}$, the last term in the momentum residuals $\mathcal{R}$ is not adequately represented. To this end, one can employ either of the two strategies : a simplified recovery using the least squares approach or the DEVSS approach. A reconstruction of $\boldsymbol{\tau_h}$ in the sabilization term would be made as the difference of the projected value on the continuous piecewise space $\boldsymbol{\tilde{\tau}_h}$ and $\boldsymbol{\tau_h}$. Considering the coarse-scale sub problem [27] and substituting $\mathbf{v}'$, p' and $\boldsymbol{\tau}'$ in terms that contain the fine scale velocity we get :

$$\left\{\begin{array}{l}
\rho(\partial_t \mathbf{v}_h, \mathbf{w}_h) + \rho(\mathbf{v}_h \cdot \nabla(\mathbf{v}_h), \mathbf{w}_h) - (p_h, \nabla \cdot \mathbf{w}_h) + (2\eta\boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\mathbf{w}_h)) \\
\quad +(\boldsymbol{\tau_h} : \boldsymbol{\varepsilon}(\mathbf{w}_h)) - (\mathbf{f}, \mathbf{w}_h) \\
\quad +\sum_{K \in \mathcal{K}_h} \tau_k \left(\rho(\partial_t \mathbf{v}_h + \mathbf{v}_h \cdot \nabla \mathbf{v}_h)_k + \nabla p_h - \nabla \cdot \boldsymbol{\tilde{\tau}_h} - \mathbf{f}, \rho \mathbf{v}_h \nabla \mathbf{w}_h\right)_k \\
\quad +\sum_{K \in \mathcal{K}_h} \tau_c (\nabla \cdot \mathbf{v}_h, \nabla \cdot \mathbf{w}_h)_k \\
\quad -\sum_{K \in \mathcal{K}_h} \tau_t (\boldsymbol{\varepsilon}(\mathbf{v}_h), \boldsymbol{\varepsilon}(\mathbf{w}_h))_k = 0 \\
(\nabla \mathbf{v}_h, q_h) \\
\quad +\sum_{K \in \mathcal{K}_h} \tau_k \left(\rho(\partial_t \mathbf{v}_h + \mathbf{v}_h \cdot \nabla \mathbf{v}_h) + \nabla p_h - \nabla \cdot \boldsymbol{\tilde{\tau}_h} - \mathbf{f}, \nabla q_h\right)_k = 0 \\
(\boldsymbol{\varepsilon_s}(\mathbf{v}_h) : \boldsymbol{\tau_h^*}) \\
\quad +\sum_{K \in \mathcal{K}_h} \tau_k \left(\rho(\partial_t \mathbf{v}_h + \mathbf{v}_h \cdot \nabla \mathbf{v}_h) + \nabla p_h - \nabla \cdot \boldsymbol{\tilde{\tau}_h} - \mathbf{f}, \nabla \cdot \boldsymbol{\tau_h^*}\right)_k = 0
\end{array}\right. \quad [33]$$

5. Numerical scheme

Three equation with three primary variables requires larger computational coast. To circumvent this issue an agmentated Lagragian method and Uzawa's algorithm would be used to solve the system without increasing the size of linear system. In the same iteration, the problem of non-linearity, time integration and computation of the Lagrange multiplier would be solved. An implicit time scheme with a Newton method for the non linear term is used. At each time step t_n the procedure is the following :

1. Initialization with values obtained at the previous time step :

$$\mathbf{v}_h = \mathbf{v}_h(t_{n-1}),\ p_h = p_h(t_{n-1}),\ \boldsymbol{\tau_h^0} = 0, \boldsymbol{\tilde{\tau}_h} = 0 \quad [34]$$

2. At step k find $\mathbf{v}_{\mathbf{h}}^{\mathbf{k}}$ and p_h^k with system :

$$\left\{\begin{array}{l}
\rho\left(\frac{\mathbf{v}_h^k}{\Delta t}+\mathbf{v}_h\cdot\nabla\mathbf{v}_h^k,\mathbf{w}_h\right)-(p_h^k,\nabla\cdot\mathbf{w}_h)+(2\eta\boldsymbol{\varepsilon}(\mathbf{v}_h^k),\boldsymbol{\varepsilon}(\mathbf{w}_h))\\
\quad+\sum_{K\in\mathcal{K}_h}\tau_k\left(\rho\left(\frac{\mathbf{v}_h^k}{\Delta t}+\mathbf{v}_h\cdot\nabla\mathbf{v}_h^k\right)+\nabla p_h^k,\rho\mathbf{v}_h\nabla\mathbf{w}_h\right)_k\\
\quad+\sum_{K\in\mathcal{K}_h}\tau_c(\nabla\cdot\mathbf{v}_h^k,\nabla\cdot\mathbf{w}_h)_k\\
\quad-\sum_{K\in\mathcal{K}_h}\tau_t(\boldsymbol{\varepsilon}(\mathbf{v}_h),\boldsymbol{\varepsilon}(\mathbf{w}_h))_k\\
\qquad\qquad\qquad=-(\boldsymbol{\tau}_{\boldsymbol{h}}^{\boldsymbol{k}}:\boldsymbol{\varepsilon}(\mathbf{w}_h))+\left(\mathbf{f}+\frac{\mathbf{v}_h}{\Delta t},\mathbf{w}_h\right)\\
\qquad\qquad\qquad\quad+\sum_{K\in\mathcal{K}_h}\tau_k\left(\rho\frac{\mathbf{v}_h}{\Delta t}+\nabla\cdot\tilde{\boldsymbol{\tau}}_{\boldsymbol{h}}^{\boldsymbol{k}}+\mathbf{f},\rho\mathbf{v}_h\nabla\mathbf{w}_h\right)_k\\
(\nabla\cdot\mathbf{v}_h^k,q_h)+\sum_{K\in\mathcal{K}_h}\tau_k\left(\rho\left(\frac{\mathbf{v}_h^k}{\Delta t}+\mathbf{v}_h\cdot\nabla\mathbf{v}_h^k\right)+\nabla p_h^k,\nabla q_h\right)_k\\
\qquad\qquad\qquad=\sum_{K\in\mathcal{K}_h}\tau_k\left(\rho\frac{\mathbf{v}_h}{\Delta t}+\nabla\cdot\tilde{\boldsymbol{\tau}}_{\boldsymbol{h}}^{\boldsymbol{k}}+\mathbf{f},\nabla q_h\right)_k
\end{array}\right. \qquad [35]$$

3. update $\tilde{\boldsymbol{\tau}}_{\boldsymbol{h}}$ and $\boldsymbol{\tau}_{\boldsymbol{h}}$:

$$\left\{\begin{array}{l}
\boldsymbol{\tau}_{\boldsymbol{h}}^{\boldsymbol{k+1}}=\boldsymbol{\tau}_{\boldsymbol{h}}^{\boldsymbol{k}}+\eta_u\boldsymbol{\varepsilon}(\mathbf{v}_h^k)\qquad\text{where}\qquad\eta_u=H(\alpha)\eta_s.\\
\tilde{\boldsymbol{\tau}}_{\boldsymbol{h}}^{\boldsymbol{k+1}}|_i=\frac{1}{\sum_{k\in\mathcal{K}(i)}\mid K\mid}\sum_{\sum_{k\in\mathcal{K}(i)}|K|}\boldsymbol{\tau}_{\boldsymbol{h}}^{\boldsymbol{k+1}}{}_{|_i}\mid K\mid
\end{array}\right. \qquad [36]$$

where $\mathcal{K}(i)$ set of nodes connected to node i.

4. Check $\parallel\ \mathcal{R}e(\mathbf{v}_h^k)\ \parallel<\ e_1$ to stop the loop on k, where $\mathcal{R}e(\mathbf{v}_h^k)$ is the residual of the inner uzawa's iterrations compared to the initial residual of the global iterations.

5. Calculate the rigid body motion : Once $\mathbf{v}_h(t_n)$ is computed of a point with $\boldsymbol{x}$ as coordinates, the optimal angular velocity $\boldsymbol{\omega}^n$ and tranlational velocity $\mathbf{V}^n$ are calculated by minimizing $\varphi(\mathbf{V}^n,\boldsymbol{\omega}^n)$ defined by :

$$\varphi(\mathbf{V}^n,\boldsymbol{\omega}^n)=\int_{\Omega_s}\mid\mathbf{V}^n+\boldsymbol{\omega}^n\wedge\boldsymbol{x}-\mathbf{v}_h(t_n)\mid^2 dx \qquad [37]$$

the minimum is atteint once

$$\left\{\begin{array}{l}
\frac{\partial\varphi(\mathbf{V}^n,\boldsymbol{\omega}^n)}{\partial\mathbf{V}^n}=0\\
\frac{\partial\varphi(\mathbf{V}^n,\boldsymbol{\omega}^n)}{\partial\boldsymbol{\omega}^n}=0
\end{array}\right. \qquad [38]$$

6. Update the rigid body position, compute $\mathbf{X}(t^{n+1})$:

$$\mathbf{X}(t^{n+1})=\mathbf{X}(t^n)+\Delta t\left(\mathbf{V}^n+\boldsymbol{\omega}^n\wedge\mathbf{X}(t^n)\right) \qquad [39]$$

A more accurate scheme for updating the rigid body position would be the Adams-Bashfort scheme (Hwang *et al.*, 2004). This scheme will be investigated in a furture work.

6. Validation for 2D cases

In this section, we present relatively simple 2D and 3D test cases in order to validate the proposed formulation and to check the accuracy and the efficiency of the immersed volume method. All the numerical simulations were carried out by using the C++ CimLib finite element library (see (Digonnet *et al.*, 2003; Mesri *et al.*, 2009)). The results obtained with the implemented code, referred as IVM, are then compared with those obtained either by standard solution or by other approaches.

6.1. *Immersion of solid bodies in fluid*

In the first model problem, we consider a square fluid domain with two immersed rigid bodies, as shown in Figure 2, induced by an imposed gradient pressure at the intlet and outlet of the fluid domain, with the following boundary conditions :

$$\begin{aligned} &\mathbf{v} = 0 && \text{at} \quad \partial\Omega_2 \cap \Omega_4 \\ &p = 100 && \text{at} \quad \partial\Omega_1 \\ &p = 0 && \text{at} \quad \partial\Omega_3 \end{aligned} \qquad [40]$$

Table 1. *Parameter of two rigid bodies in a viscous fluid*

Parameter	ρ_f	ρ_s	η_f
Unit	kg/m^3	kg/m^3	$Pa.s$
Value	1000	1000	1000

The objective of this test is to illustrate the formulation and to check implementation of the proposed method by comparing results to the solution representing only the fluid domain. The used parameters are tabulated in Table 1. The velocity and pressure fields computed with the proposed method are shown in Figures 3 and 4 and compared to results calculated only on the fluid domain. The agreement between the two calculations shows that the present solver is able to predict accuractely the behaviour of the fluid and the presence of the solid. The pressure distribution caused by the interaction is more interesting, which is depicted in Figure 4 on both domains. The effect of the stabilization is depicted in the zoomed-in view of the pressure near the interface as shown in Figure 5. The variational multiscale method eliminates the pollution in the solution especially on the interface.

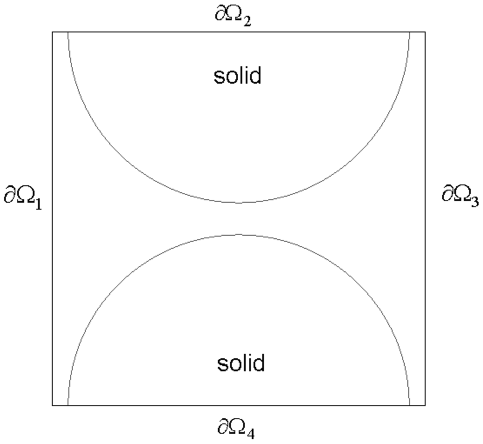
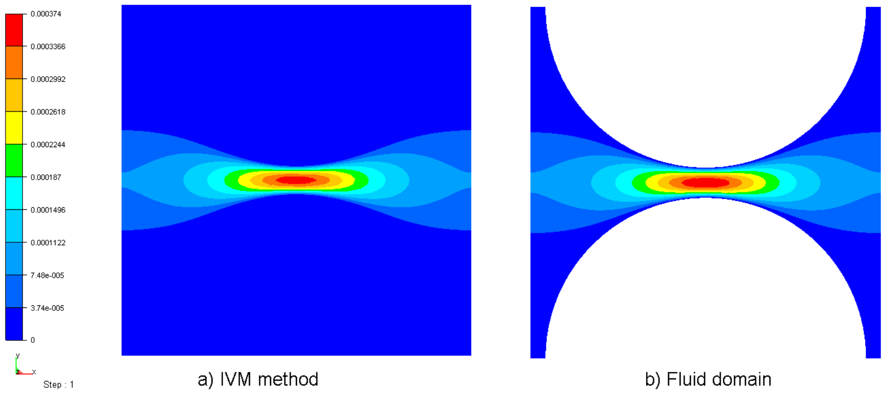


Figure 2. *Shematic representation of two rigid bodies in an incompressible flow*



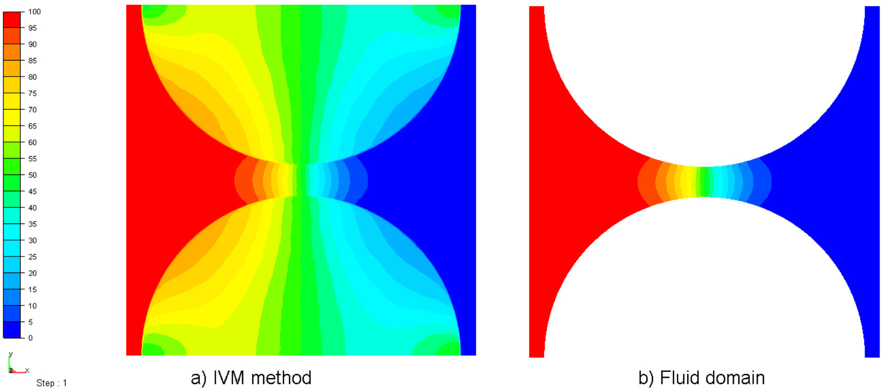
a) IVM method

b) Fluid domain

Figure 3. *Norme of the velocity with and without IVM*

6.2. *Several disks in an imposed flow*

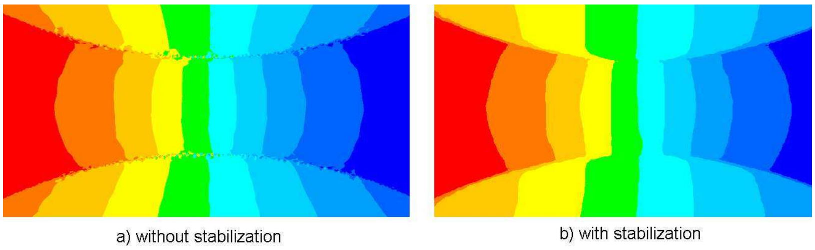
For further validation with several small particles immersed in a fluid, fifteen rigid circular disks and ten rigid semi-circular disks with radius $R = 5mm$ are immersed in an imposed fluid flow, with a velocity at the intlet $u = 0.015m.s^{-1}$ (see Figure 6). The no-slip boundary conditions are imposed on the tube wall and bottom of the domain. The length of the tube is $L = 19R$ and its width is $D = 12R$. The density and viscosity of the fluid used in this example are $\rho_f = 1370kg/m^3$ and $\eta_f = 0.001Pa.s$,



a) IVM method

b) Fluid domain

Figure 4. *Pressure distribution with and without IVM*



a) without stabilization

b) with stabilization

Figure 5. *Close up view for the pressure at the interface with and without stabilization*

and the density of the particule is $\rho_s = 2000 kg/m^3$. The mesh size is $h = 0.0005m$. Figure 7 shows the velocity vectors of the flow. The restriction of the velocity profile is well highlighted in the wake region of the disks. Figures 8 and 9 illustrate the contour plots for u and v velocity respectively. These figures also show the behaviour of the fluid flow commonly observed in several obstacles. It can be noticed, the maximum velocity is attained in the region where the separation distance between the two disks is minimum. The pressure is presented in Figure 10. As mentioned earlier, only the real pressure of fluid is highlighted. The same example is repeated in $3D$. As shown in Figure 11, the fluid-solid interface is well captured. The behaviour of the fluid flow is similar to the $2D$ case. Once again, The proposed $3D$ FSI solver exhibits good stability properties on anisotropic unstructured meshes. The extension of this problem is to take one of the disks as an elastic body.

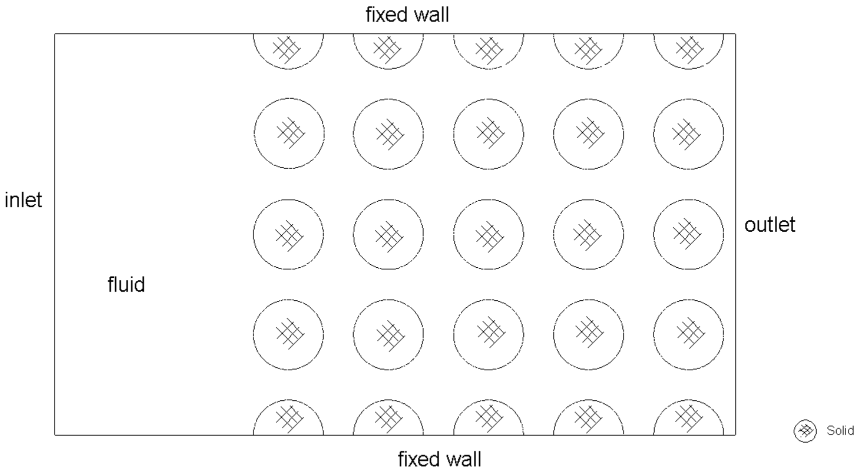


Figure 6. *Modelling several rigid disks in an imposed flow*

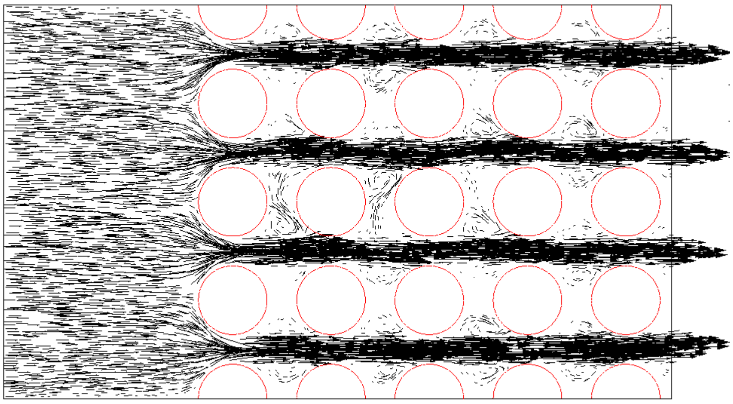


Figure 7. *Velocity distribution between the rigid disks (the circles present the zero isovalue of the level-set)*

6.3. *Falling disk in a channel*

We consider a disk with radius $R = 0.125cm$ falling under the action of gravitational force inside a channel of dimension $[0, 2] \times [0, 6]$. Parameters used in this example are tabulated in Table 2. Among these parameters, h denotes the mesh size.

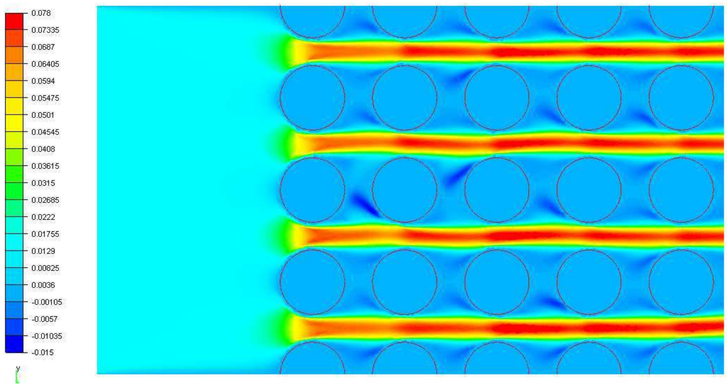


Figure 8. *u-velocity in an imposed fluid flow (the circles present the zero isovalue of the level-set)*

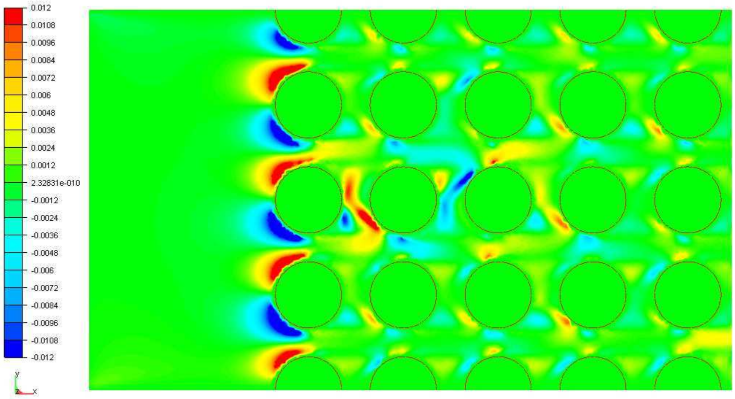


Figure 9. *v-velocity in an imposed fluid flow (the circles present the zero isovalue of the level-set)*

Table 2. *Parameter used in the computation of a falling disk in a channel*

Parameter	ρ_f	ρ_s	η_f	$\triangle t$	g	h
Unit	g/cm^3	g/cm^3	g/cms	s	cm/s^2	cm
Value	1	1.25	0.1	0.005	980	0.04

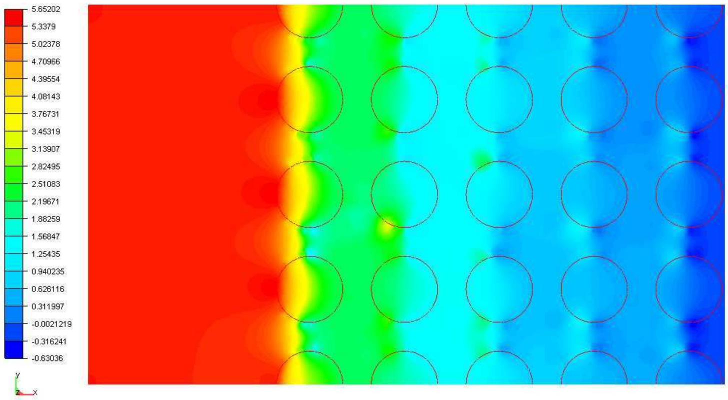


Figure 10. *Pression distribution in an imposed fluid flow (the circles present the zero isovalue of the level-set)*

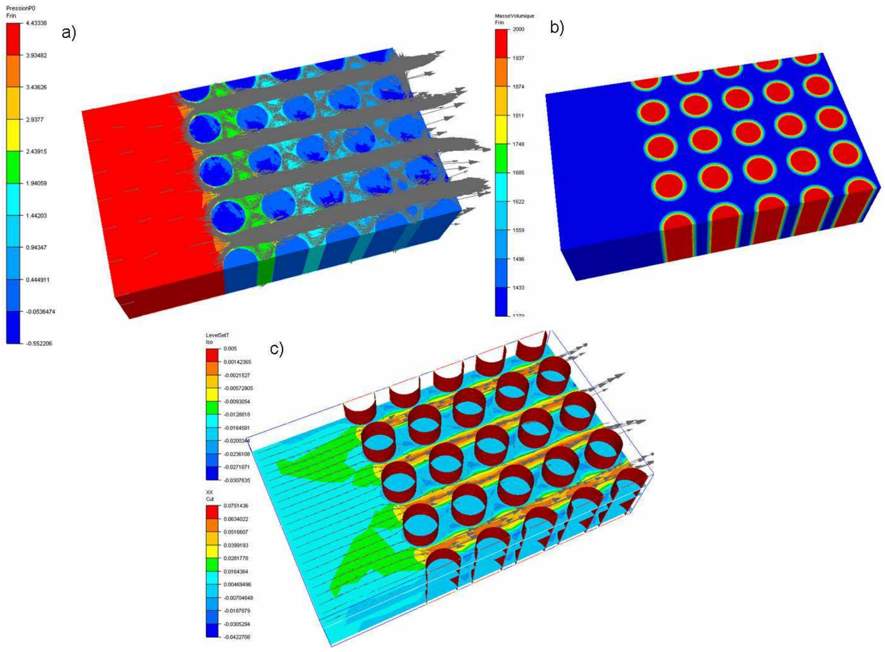


Figure 11. $3D$ *modelling of several rigid disks in an imposed flow a) velocity b) density c) Level-set*

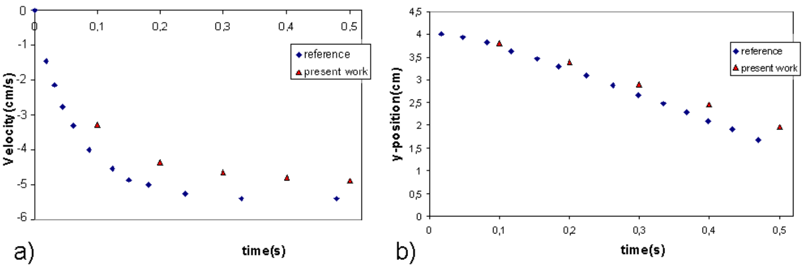
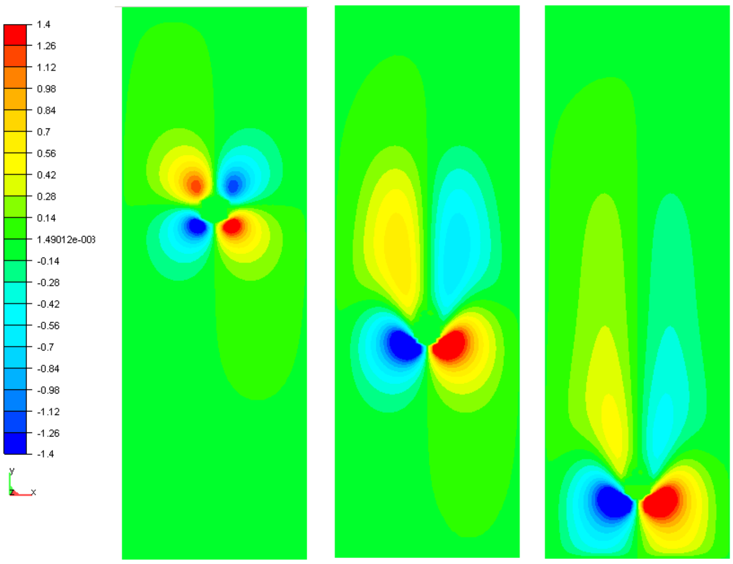


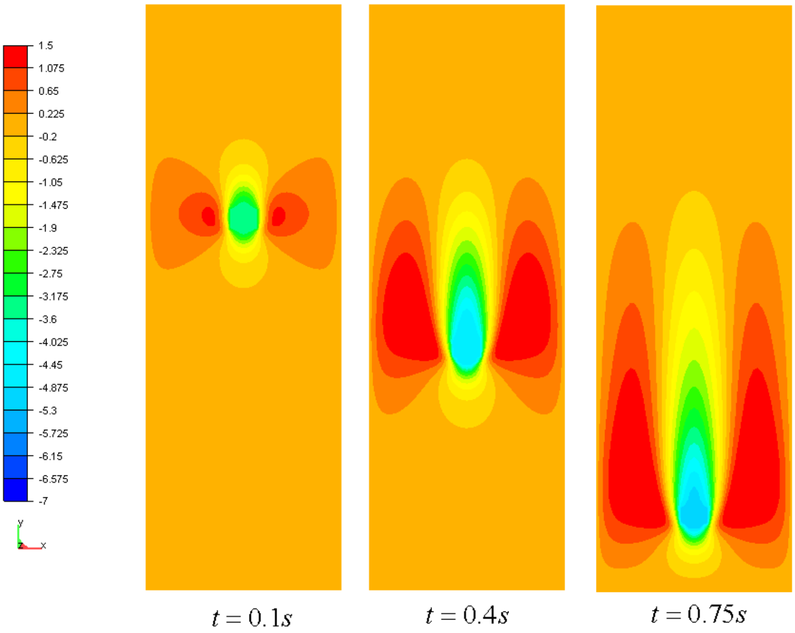
Figure 12. *The vertical velocity a) and the vertical postition b) of the center of a disk in a channel*



$t = 0.1s$ $t = 0.4s$ $t = 0.75s$

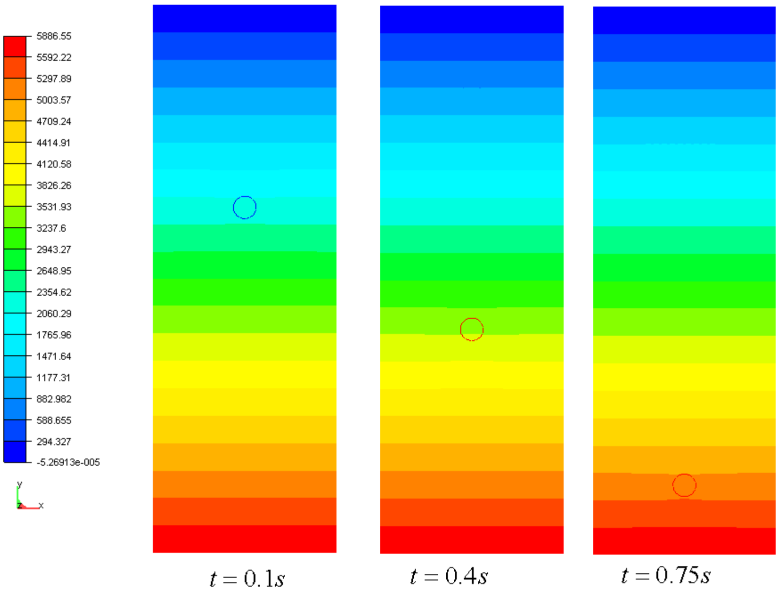
Figure 13. *u-velocity on the entire domain due to the falling disk in a channel*

In (Glowinski *et al.*, 2001) the velocity of particulate flow with rigid circular disks using fictitious domain method is calculated. We compare our computational results to this reference. This test case is well documented in the literature and considered as a challenging benchmark. Close agreements in Figure 12 are found for the velocity and position of the center of the disk as t goes on. The slight differences noticed in here are mostly related to the use of a first order time interpolation scheme for the solid motion. This matter can be adjusted using the Adams-Bashfort scheme. Figures



$t = 0.1s$ $t = 0.4s$ $t = 0.75s$

Figure 14. *v-velocity on the entire domain due to the falling disk in a channel*



$t = 0.1s$ $t = 0.4s$ $t = 0.75s$

Figure 15. *Excess pressure on the entire domain due to the falling disk in a channel*

13 and 14 illustrate, respectively, the velocity contour plot for u and v at selected instances. It can be observed from Figure 15 that the pressure computed at different time is not affected by the presence of the solid and free of any spurious oscillations. This result endorse the computation of the fluid pressure only.

6.4. *Tetris benchmark*

Four rigid bodies with different density $\rho_{s_1} = 3000kg/m^3$, $\rho_{s_2} = \rho_{s_3} = 8000kg/m^3$, $\rho_{s_4} = 8000kg/m^3$ are falling under the gravitationnal force. When several rigid bodies in the incompressible fluid channel at the initial time are falling, the interactive motions of these bodies inside the fluid have an interesting phenomena. At the beginning, each body has the same acceleration by gravitational force. As time passes more, the velocity of the upper bodies becomes faster than that of the lowers since the lowers undergoes more resistance against the fluid comparing to the upper ones. The rigid bodies are arranged, initially, as shown in Figure 16, together with the finite element mesh at $t = 0$. Mesh adaptation is needed to better capture all the interfaces. The error estimator method is used in this example, with a fixed number of 26423 elements. The density and viscosity of the fluid used in this example are $\rho_f = 1kg/m^3$ and $\eta_f = 0.0005Pa.s$.

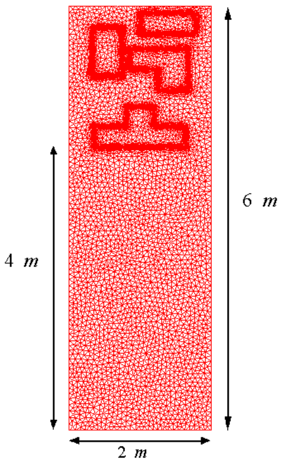


Figure 16. *Immersion of four rigid bodies : geometry and finite element mesh at* $t = 0$

The objective of this test is to show the capability of the method to handle high discontinuities of the solids and fluid physical properties. The Figures 17 and 18 show, respectively, the finite element mesh and the effect of the anisotropic mesh adaptation on respecting the geometry of the rigid bodies at different time instants. While the velocity vectors at different time instants is depicted in Figure 19. All the

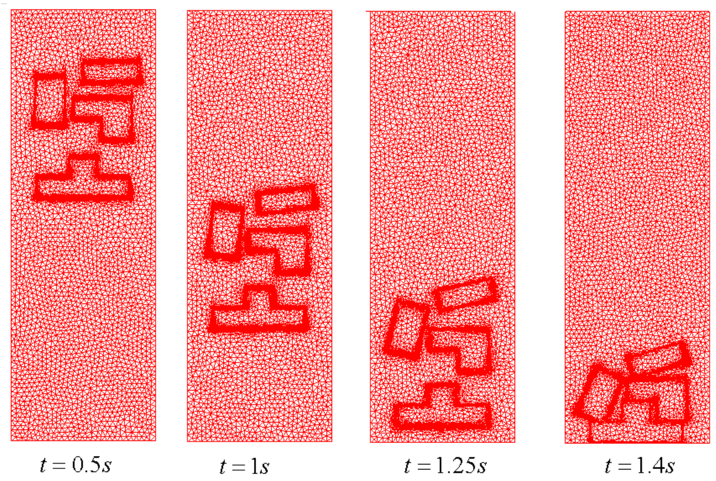


Figure 17. *Immersion of four rigid bodies : finite element mesh of four rigid bodies at different time instants*

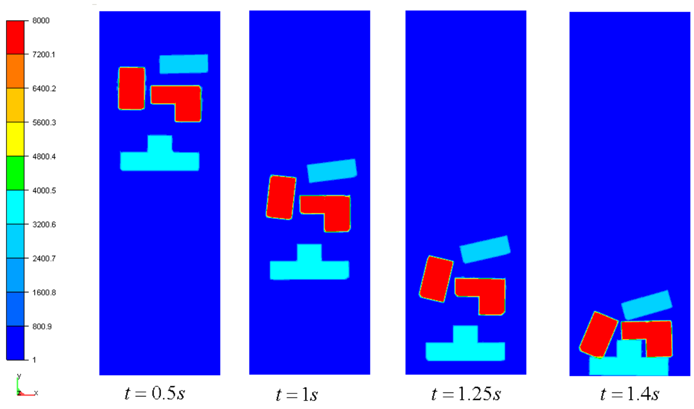


Figure 18. *Density distribution for different time step with anisotropic adapted interfaces*

vortices behind the solid objects are well computed. The developed solver is able, at the same time, to take into account different solid bodies in an incompressible fluid flow with very low viscosity. Interaction between the solids is highlited.

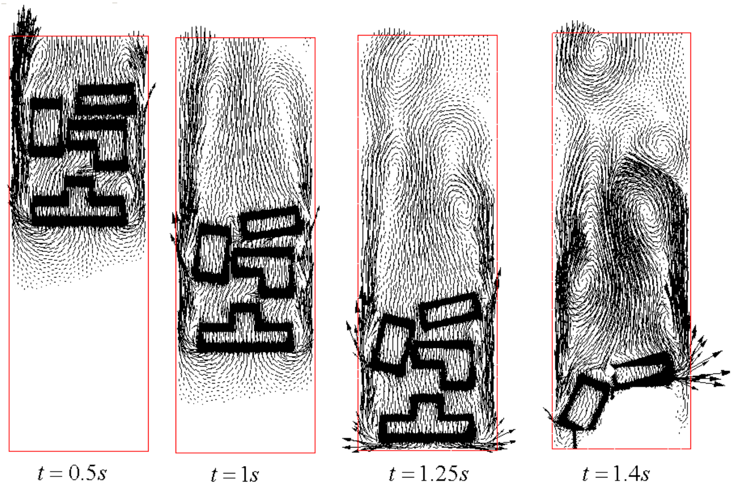


Figure 19. *Immersion of four rigid bodies : velocity vectors at different time instants*

7. Conclusion

In this paper we have described an evalutaion of the stabilized three-field velocity-pressure-stress, designed for the computation of rigid bodies in an incompressible Navier-Stokes flow. The proposed approach solves one set of equation in both domains with different materials properties. The presence of the solid is taken into account as an extra stress in the Navier-Stokes equation. The formulation considered allows equal-order interpolation for the velocity and pressure. The use of uzawa's algorithm to solve the system requires a lower-order interpolation for the stress field, since if not, the left side of the system will totally changed and another system would be reached. A new developped monolithic multiscale-stabilized finite element method is presented. The approach is applied to the numerical simulation of 2D and 3D test cases. The capability of the model to simulate the fluid-rigid body interaction was demonstrated. Results are assessed by comparing the predictions with reference or other approaches. Furhter research will focus on the use of a continous piecewise interpolation for the stress field. The improvement of this methodology could be also taking a better scheme for particule discplacement with a higher order time integration schemes. Enlarge the field of application and tackling deformable solid interaction and more 3D numerical simulation are considered as perspective work.

Acknowledgement

The authors gratefully acknowledge support from the Transvalor and CIM Team.

8. References

Badia S., Codina R., « Stabilized continuous and discontinuous galerkin techniques for Darcy flow », *Computer Methods in Applied Mechanics and Engineering*, vol. 199, p. 1654-1667, 2010.

Benson D., « An efficient, accurate, simple ALE method for nonlinear finite element programs », *Computer Methods in Applied Mechanics and Engineering*, vol. 72, n° 3, p. 305-350, 1989.

Bruchon J., Digonnet H., Coupez T., « Using a signed distance function for the simulation of metal forming processes : Formulation of the contact condition and mesh adaptation », *International Journal for Numerical Methods in Engineering*, vol. 78, n° 8, p. 980-1008, 2009.

Caussin P., Gerbeau J.-F., F. N., « Added-mass effect in the design of partitioned algorithms for fluid-structure problems », *Computer Methods in Applied Mechanics and Engineering*, vol. 194, n° 42-44, p. 4506-4527, 2005.

Cereva M., Chinumenti M., Codina R., « Mixed stabilized finite element methods in nonlinear solid mechanics : PartI : Formulation », *Computer Methods in Applied Mechanics and Engineering*, vol. 199, n° 37-40, p. 2559-2570, 2010.

Codina R., « Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods », *Computer Methods in Applied Mechanics and Engineering*, vol. 190, n° 13-14, p. 1579-1599, 2000.

Codina R., « Pressure stability in fractional step finite element methods for incompressible flows », , vol. 170, p. 112-140, 2001.

Codina R., González-Ondina J. M., Díaz-Hernández G., Principe J., « Finite element approximation of the modified Boussinesq equations using a stabilized formulation », *International Journal for Numerical Methods in Fluids*, vol. 57, n° 9, p. 1249-1268, 2008.

Coquerelle M., Cottet G.-H., « A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies », *Journal of Computational Physics*, vol. 227, n° 21, p. 9121-9137, 2008.

Coupez T., « Génération de maillage et adaptation de maillage par optimisation locale », *Revue européenne des éléments finis*, vol. 9, p. 403-423, 2000.

Coupez T., Hugues D., Hachem E., Laure P., Silva L., Valette R., – *Multi domain Finite Element Computations : Application to Multiphasic Problems– Arbitrary Lagrangian-Eulerian and Fluid-Structure Interaction*, ISTE Ltd and John Wiley Sons Inc, 2010.

Digonnet H., Coupez T., « Object-oriented programming for fast and easy development of parallel applications in forming processes simulation », *K.J. Bathe, (ed)., Second MIT Conference on Computational Fluid and Solid Mechanics*, p. 1922-1924, 2003.

Donea J., Giuliani S., Halleux J. P., « An Arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions », *Computer Methods in Applied Mechanics and Engineering*, vol. 33, n° 1-3, p. 689-723, 1982.

Fernández M. A., Moubachir M., « A Newton method using exact jacobians for solving fluid-structure coupling », *Computers and Structure*, vol. 83, n° 2-3, p. 127-142, 2005.

Franca L. P., Hughes T. J. R., « Two classes of mixed finite element methods », *Computer Methods in Applied Mechanics and Engineering*, vol. 69, n° 1, p. 89-129, 1988.

Gerbeau J.-F., Vidrascu M., « A quasi-Newton algorithm based on a reduced model for fluid structure interaction problems in blood flow », *Mathematical Modelling and Numerical Analysis*, vol. 37, n° 4, p. 631-647, 2003.

Gerbeau J.-F., Vidrascu M., Frey P., « Fluid structure interaction in blood flows on geometries coming from medical imaging », *Computers and Structure*, vol. 83, n° 2-3, p. 155-165, 2005.

Glowinski R., Pan T., Helsa T. I., Joseph D. D., Periaux J., « A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies : application to particulate flow », *Journal of Computational Physics*, vol. 169, n° 2, p. 363-426, 2001.

Glowinski R., Pan T. W., Helsa T.-I., Joseph D. D., « A distributed lagrange multiplier/fictious domain methods for particulate flows », *International Journal of Multiphase flow*, vol. 25, n° 5, p. 755-794, 1999.

Gruau C., Coupez T., « 3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric », *Computer Methods in Applied Mechanics and Engineering*, vol. 194, n° 18-49, p. 4951-4976, 2005.

Hachem E., Stabilized Finite Element Method for Heat Transfer and Turbulent Flows inside Industrial Furnaces, PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 2009.

Hachem E., B. R., Kloczko T., Digonnet H., Coupez T., « Stabilized finite element method for incompressible flows with high Reynolds number », *Journal of Computational Physics*, vol. 229, n° 23, p. 8643-8665, 2010.

Hirt C. W., Amsden A. A., Cook J. L., « An Arbitrary Lagrangian-Eulerian computing method for all speeds », *Journal of Computational Physics*, vol. 14, n° 3, p. 227-253, 1974.

Hughes T. J. R., Liu W. K., Zimmerman T., « Lagrangian-Eulerian finite element formulation for incompressible viscous flow », *Computer Methods in Applied Mechanics and Engineering*, vol. 29, n° 3, p. 329-349, 1981.

Hwang W. R., Hulsen M. A., Meijer H. E., « Direct simulations of particle suspensions in a viscoelastic fluid in sliding bi-periodic frames », *Journal of Non-Newtonian Fluid Mechanics*, vol. 121, n° 1, p. 15-33, 2004.

Janela J., Lefebvre A., Maury B., « A penalty method for the simulation of fluid-rigid body interaction », *ESAIM Proceedings*, vol. 14, p. 115-123, 2005.

Küttler U., Wall A. W., « Fixed-point fluidstructure interaction solvers with dynamic relaxation », *Computational Mechanics*, vol. 43, n° 1, p. 61-72, 2008.

Laure P., Beaume G., Basset O., Silva L., Coupez T., « Numerical methods for solid particles in particulate flow simulations », *European Journal of Computational Mechanics*, vol. 16, p. 365-383, 2007.

Le Tallec P., Mouro J., « Fluid structure interaction with large structural displacements », *Computer Methods in Applied Mechanics and Engineering*, vol. 190, n° 24-25, p. 3039-3067, 2001.

Mesri Y., Digonnet H., Coupez T., « Advanced parallel computing in material forming with CIMLIB », *European Journal of Computational Mechnanics*, vol. 18, n° 7-8, p. 669-694, 2009.

Michler C., E.H. van Brummelen E., de Borst R., « An interface NewtonKrylov solver for fluidstructure interaction », *International Journal for Numerical Methods in Fluids*, vol. 47, n° 10-11, p. 1189-1195, 2005.

Peskin C. S., « The immersed boundary method », *Acta Numerica*, vol. 11, p. 479-517, 2002.

Ritz J. B., Caltagirone J. P., « A Numerical Continuous Model for the Hydrodynamics of Fluid Particle Systems », *International Journal for Numerical Methods in Fluids*, vol. 30, n° 8, p. 1067-1090, 1999.

Souli M., Ouahsine A., Lewin L., « ALE formulation for fluid structure interaction problems », *Computer Methods in Applied Mechanics and Engineering*, vol. 190, n° 5-7, p. 659-675, 2000.

van der Pijl S., Segal A., Vuik C., Wesseling P., « A mass-conserving Level-Set method for modelling of multi-phase flows », *International Journal for Numerical Methods in Fluids*, vol. 47, n° 4, p. 339-361, 2005.

van Loon R., Anderson P. D., van de Vosse F. N., Sherwin S. J., « Comparison of Various fluid-structure interaction methods for deformable bodies », *Computers and Structures*, vol. 85, n° 11-14, p. 833-843, 2007.