Computers and Structures 122 (2013) 88–100

Contents lists available at SciVerse ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

Anisotropic adaptive meshing and monolithic Variational Multiscale method for fluid–structure interaction

E. Hachem [a,*], S. Feghali [a], R. Codina [b], T. Coupez [a]

[a] Center for Material Forming (CEMEF), MINES-ParisTech, UMR CNRS 7635, Sophia-Antipolis, France
[b] Universitat Politecnica de Catalunya (UPC), Jordi Girona 1-3, 08034 Barcelona, Spain

ARTICLE INFO

Article history:
Received 7 September 2012
Accepted 3 December 2012
Available online 5 January 2013

Keywords:
Stabilized finite elements
Monolithic approach
Rigid body motions
Incompressible flows
Fluid–structure interactions
Anisotropic meshing

ABSTRACT

This paper presents a monolithic formulation framework combined with an anisotropic mesh adaptation for fluid–structure interaction (FSI) applications with complex geometry. The fluid–solid interfaces are captured using a level-set method. A new *a posteriori* error estimate, based on the length distribution tensor approach and the associated edge based error analysis, is then used to ensure an accurate capturing of the discontinuities at the fluid–solid interface. It enables to calculate a stretching factor providing a new edge length distribution, its associated tensor and the corresponding metric. The optimal stretching factor field is obtained by solving an optimization problem under the constraint of a fixed number of edges in the mesh. The presence of the structure will be taken into account by means of an extra stress tensor in the Navier–Stokes equations. The system is solved using a stabilized three-field, stress, velocity and pressure finite element (FE) formulation. It consists in the decomposition for both the velocity and the pressure fields into coarse/resolved scales and fine/unresolved scales and also in the efficient enrichment of the extra constraint. We assess the accuracy of the proposed formulation by simulating 2D and 3D time-dependent numerical examples such as: falling disk in a channel, turbulent flows behind an airfoil profile and flow behind an immersed vehicle.

1. Introduction

Fluid–structure Interaction (FSI) describes a wide variety of industrial problems arising in engineering, technology and biomechanics. Due to the high complexity of these problems, FSI simulations are nowadays the focus of numerous investigations and various approaches are proposed to treat them.

Two main approaches for the simulation of FSI problems are still gaining attention lately: partitioned and monolithic approaches. The partitioned approaches allow the use of a specific solver for each domain. The fluid and the structure equations are alternatively integrated in time and the interface conditions are enforced asynchronously. The difficulty remains in transferring the informations between the codes. The coupling between the two phases can be enforced using different schemes: weakly or strongly coupled versions. The former approach manages with just one solution of either field per time step but consequently lack accurate fulfillment of the coupling conditions. The latter requires sub-iterations [1–6]. It is accurate and quite efficient but presents an inherent instability depending on the ratio of the densities and the geometry of the domain [7]. For 3D problems, the numerical cost can increase drastically. Alternatively, authors in [8] propose an immersed particle method able to handle complicated FSI problems including cracking and perforation.

Monolithic methods are still of interest due to their capability to treat the interaction of the fluid and the structure at the interface synchronously [9–11]. The continuity at the interface is obtained naturally and there is no need to enforce it. They impose the use of an appropriate unique constitutive equation describing both the fluid and the solid domains. Interface tracking between the two different domains can be completed by Immersed Boundary (IB) methods [12] where the interface is convected in a Lagrangian way. Other methods such as the fictitious domain method [12,13] treat the coupling between the domains by applying a constraint across the body using a Lagrange multiplier. These constraints may lead to uncoupled physics in the different subdomains of the problem (in the fluid and the solid, for example), yielding inconsistencies when the subdomains evolve in time. This problem may be solved using the so called Fixed-mesh ALE formulation introduced in [14] (see also [15,16] for applications to fluid–structure interaction problems and rigid bodies floating in fluids). Likewise and for more complex problems using the ALE formulation, the authors in [17] proposed a mesh adaptivity procedure for fluid–structure interactions capable of handling high gradients in the solution, boundary layer effects and large structural deformations.

* Corresponding author. Tel.: +33 (0)4 93 95 74 58; fax: +33 4 92 38 97 52.
E-mail addresses: Elie.hachem@mines-paris tech.fr (E. Hachem), Stephanie.El_Feghali@mines-paristech.fr (S. Feghali), ramon. codina@upc.edu (R. Codina), Thierry.Coupez@mines-paristech.fr (T. Coupez).

0045-7949/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compstruc.2012.12.004

In this paper, we focus on a monolithic formulation where the complete problem is written in a fully Eulerian framework and the fluid and solid phases are separated by a level-set function. This was started in [18,19] for simulating the interactions between a fluid and fixed solids. The solid was simply treated as a region with high viscosity and the mesh near the fixed interfaces was refined a priori and before the simulation. The criterion was obviously the levelset function.

In this work, we develop a new monolithic approach that differs from the previous developments in two main aspects. The first one is related to the way we adapt the mesh in particular for moving rigid bodies and the second focuses more on how we treat the solid regions in the Navier–Stokes equations. Therefore, we introduce a new dynamic anisotropic mesh adaptation method to deal properly with moving interfaces which still is a key challenge in most of the monolithic approaches. Inspired from the work in [20], we propose an extension of the edge based error estimation to combine the simultaneous adaptivity to the interface and to the velocity field using one simple global vector field. With such an advantage, it becomes a very useful and practical tool for a wide range of FSI problems. The second point concerns more the development of a three-field stabilized finite element method and its implementation aspects for modeling the interaction between the fluid (laminar or turbulent) and the rigid bodies (fixed or moving). The presence of the structure will be then taken into account by means of an extra stress tensor in the Navier–Stokes equations.

In Sections 2–4, we first consider the level-set function, commonly employed in the simulation of multiphase flows [21], used to distinguish the phases. It allows to easily deal with very complex geometries, large structural deformations and free movements of the structure within a flow domain. However, the level-set intersects the elements arbitrarily and lacks the ability to reproduce the interfaces of complex geometries (i.e. sharp corners). Therefore, we combine it with anisotropic mesh adaptation. An *a posteriori* edge based spatial error indicator relying on the length distribution tensor approach is presented in section 3. The anisotropic adaptation involves building a mesh based on a metric map. It provides both the size and the stretching of elements in a very condensed information data. Working on a nodal based metric, an anisotropic mesh adaptation procedure is obtained under the constraint of a fixed number of nodes. With such an advantage, it becomes a very useful and practical numerical tool. Such an algorithm allows the creation of extremely stretched elements along the interface, which is an important requirement for FSI problems with high density ratios.

In Section 4, we then present the development of the FE solver. The rigid immersed body is treated using the Navier–Stokes solver under constraints of imposing the nullity of the deformations by means of a Lagrange multiplier. The system is solved using a new Variational Multiscale FE method. Thus we propose to extend the decomposition for both the velocity and the pressure fields into coarse/resolved scales and fine/unresolved scales, needed to deal with convection dominated problems and pressure instabilities, with an efficient enrichment of the extra constraint. This choice of decomposition is shown to be favorable for simulating flows at high Reynolds number and to remove spurious oscillations at the interface due to the high discontinuity in the material properties. We retain in this work the advantages of using the P1 finite elements approximation regarding the accuracy and the computational cost, especially for 3D real applications.

The capability of the developed finite element method in handling extremely stretched elements and in producing very satisfactory results is highlighted in Section 5 through different numerical tests. We show that the proposed anisotropic meshing technique is well suited for these fluid–structure interaction problems and could be embedded into different FSI techniques such as [22–24]. This is due to the fact that the latter method takes into account multicomponent fields simultaneously (tensors, vectors, scalars) characterizing the structure and the physics of the problem. For instance it includes the velocity norm, the velocity components and the Level-Set function combined into one single metric field. In Section 6, we give our concluding remarks.

2. Construction of an anisotropic mesh

In this section, we retrace the main steps of the adaptive procedure used to immerse and to represent different complex geometries inside a unique mesh. First we compute the signed distance function (level-set) of a given geometry to each node of the mesh, then we refine anisotropically the mesh at the interface and finally we mix and attribute the physical properties of each domain using appropriate laws. This procedure is repeated iteratively for moving solids.

2.1. Level-set function

A signed distance function of an interface Γ_{im} is used to localize the interface of the immersed body and initialize the desirable properties on both sides of the latter. At any point $\mathbf{x}$ of the computational domain Ω, the level-set function α_{im} corresponds to the signed distance from Γ_{im}. In turn, the interface Γ_{im} is given by the iso-zero of the function α_{im}:

$$\begin{cases} \alpha_{\text{im}}(\mathbf{x}) = \pm d(\mathbf{x}, \Gamma_{\text{im}}), & \mathbf{x} \in \Omega, \\ \Gamma_{\text{im}} = \{\mathbf{x}, \alpha_{\text{im}}(\mathbf{x}) = 0\}. \end{cases} \tag{1}$$

In this paper, a sign convention is used: $\alpha_{\text{im}} \geqslant 0$ inside the solid domain defined by the interface Γ_{im} and $\alpha_{\text{im}} < 0$ outside this domain. Further details about the algorithm used to compute the distance are available in [25]. It is also possible to use functions smoother than $d(\mathbf{x}, \Gamma_{\text{im}})$ far from Γ_{im} (see for example [26]).

2.2. Edge based error estimation

An *a posteriori* error estimate based on the length distribution tensor approach and the associated edge based error analysis [20] is presented. It enables to calculate a stretching factor providing a new edge length distribution, its associated tensor and the corresponding metric. The optimal stretching factor field is obtained by solving an optimization problem under the constraint of a fixed number of edges in the mesh. In this work, we emphasis the application of this new technique to multi-domain problems. Therefore, for addressing a high contrast in the physical parameters, we propose an extension of the *a posteriori* estimation. It combines the simultaneous adaptivity to the level-set scalar field and to the velocity field without increasing the complexity of the computation or intersecting different metrics. Using this approach, the adaptivity will also focus on the change of direction rather than the intensity of the velocity. This is clearly shown behind the obstacle in Fig. 1, whereas the adaptation on the level-set function renders extremely stretched elements along the fluid–solid interface. With such a method, we can provide a very useful and practical tool for the simulation of complex FSI problems. In the following subsections, details of the adaptivity approach will be discussed.

We consider a variable $u \in \mathcal{C}^2(\Omega) = \mathcal{V}$ and $\mathcal{V}_h$ a simple P^1 finite element approximation space: $\mathcal{V}_h = \{w_h \in \mathcal{C}^0(\Omega),\ w_h|_K \in P^1(K),\ K \in \mathcal{K}\}$ where $\Omega = \bigcup_{K\in\mathcal{K}} K$ and K is a simplex (segment, triangle, tetrahedron, ...).

We define $\mathbf{X} = \{\mathbf{X}^i \in \mathbb{R}^d,\ i = 1, \ldots, N\}$ as the set of nodes of the mesh and we denote by U^i the nodal value of u at $\mathbf{X}^i$ and we let Π_h be the Lagrange interpolation operator from $\mathcal{V}$ to $\mathcal{V}_h$ such that: $\Pi_h u(\mathbf{X}^i) = u(\mathbf{X}^i) = U^i,\ \forall i = 1, \ldots, N$. As shown in Fig. 2, we denote

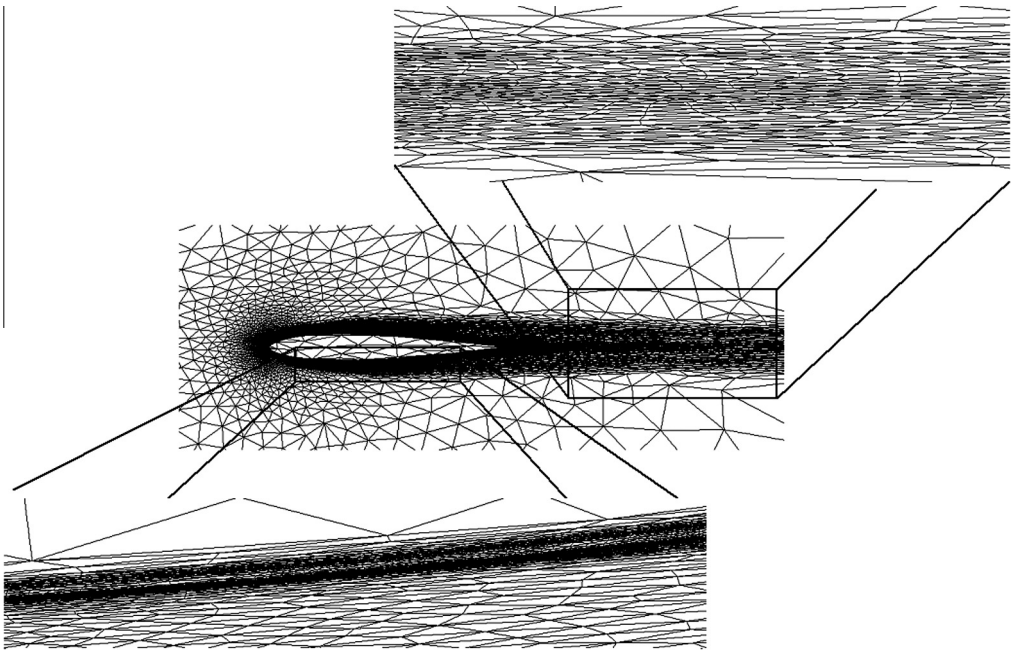

Fig. 1. Anisotropic refined fluid–solid interface of an immersed NACA0012.

the set of nodes connected to node i by $\Gamma(i) = \{j,\ \exists K \in \mathcal{K},\ \mathbf{X}^i, \mathbf{X}^j \text{ are nodes of} K\ \}$.

By introducing the notation: $\mathbf{X}^{ij} = \mathbf{X}^j - \mathbf{X}^i$ and using the analysis carried in [20], we can set:

$$\nabla u_h \cdot \mathbf{X}^{ij} = U^{ij}, \tag{2}$$

$$\|\underbrace{\nabla u_h \cdot \mathbf{X}^{ij}}_{U^{ij}} - \nabla u(\mathbf{X}^i) \cdot \mathbf{X}^{ij}\| \leqslant \max_{\mathbf{Y} \in [\mathbf{X}^i, \mathbf{X}^j]} |\mathbb{H}(u)(Y)\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|, \tag{3}$$

where $\mathbb{H}(u) = \nabla^{(2)} u$ is the associated Hessian of u. Recall that taking $u \in \mathcal{C}^2(\Omega)$ we obtain $\nabla u \in \mathcal{C}^1(\Omega)$.

Applying the interpolation operator on ∇u together with (2) we obtain a definition of the projected second derivative of u in terms of only the values of the gradient at the extremities of the edge:

$$\nabla \mathbf{g}_h \mathbf{X}^{ij} \cdot \mathbf{X}^{ij} = \mathbf{g}^{ij} \cdot \mathbf{X}^{ij}, \tag{4}$$

where $\mathbf{g}_h = \Pi_h \nabla u$, $\mathbf{g}^i = \nabla u(\mathbf{X}^i)$ and $\mathbf{g}^{ij} = \mathbf{g}^j - \mathbf{g}^i$.

Using a mean value argument, we set that: $\exists \mathbf{Y} \in [\mathbf{X}^i, \mathbf{X}^j] | \mathbf{g}^{ij} \cdot \mathbf{X}^{ij} = \mathbb{H}(u)(Y)\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}$.

We use this projection as an approximation of the error along the edge:

$$e_{ij} = \mathbf{g}^{ij} \cdot \mathbf{X}^{ij}. \tag{5}$$

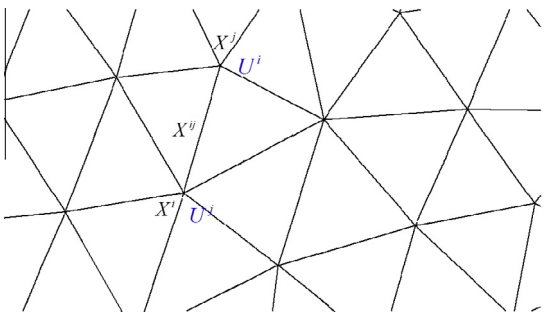

Fig. 2. Length $\mathbf{X}^{ij}$ of the edge joining nodes i and j.

However this equation cannot be evaluated exactly as it requires knowing the gradient of u and also its continuity at the nodes of the mesh. For that reason, we resort to a gradient recovery procedure.

2.3. Gradient recovery

Based on an optimization analysis, the author in [20] proposes a recovery gradient operator defined by:

$$\mathbf{G}^i = (\mathbb{X}^i)^{-1} \sum_{j \in \Gamma(i)} U^{ij} \mathbf{X}^{ij}, \tag{6}$$

where $\mathbb{X}^i = \frac{d}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij}$ is what we call the length distribution tensor at node $\mathbf{X}^i$. Note that this construction preserves the second order: $|(\mathbf{G}^i - \mathbf{g}^i) \cdot \mathbf{X}^{ij}| \sim (\mathbb{H}(u)\mathbf{X}^{ij} \cdot \mathbf{X}^{ij})$ where $\mathbf{G}^i$ is the recovery gradient at node i (given by (6)) and $\mathbf{g}^i$ being the exact value of the gradient at node i.

The approximated error is evaluated by substituting $\mathbf{G}$ by $\mathbf{g}$ in (5):

$$e_{ij} = \mathbf{G}^{ij} \cdot \mathbf{X}^{ij}.$$

2.4. Metric construction from the edge distribution tensor

Taking into account this error analysis, we construct the metric for the unit mesh as follows:

$$\mathbb{M}^i = \left(\frac{d}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} \right)^{-1}.$$

For a complete justification of this result, the reader is referred to [20].

2.5. Error behavior due to varying the edge length

We examine now how the error behaves when the length of the edges changes by stretching coefficients $s_{ij} \in \mathcal{S}$ defined by:

$$\mathcal{S} = \{s_{ij} \in \mathbb{R}^+,\ i = 1,\dots,N,\ j = 1,\dots,N,\ \Gamma(i) \cap \Gamma(j) \neq \emptyset\}.$$

To obtain a new metric depending on the error analysis, a new length for each edge has to be calculated and then used for rebuilding the length distribution tensor. An interesting way of linking the error variations to the changes in edge lengths is by introducing a stretching factor $s \in \mathbb{R}^+$ such that

$$\begin{cases} \widetilde{\mathbf{X}^{ij}} = s\mathbf{X}^{ij} \\ \|\widetilde{e_{ij}}\| = s^2\|e_{ij}\| = s^2\|\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}\| \end{cases} \tag{7}$$

where $\widetilde{e_{ij}}$ and $\widetilde{\mathbf{X}^{ij}}$ are the target error at edge ij and its associated edge length, respectively (see Fig. 3).

Following the lines of [20] we can simply define the metric associated with $\mathcal{S}$ by:

$$\widetilde{\mathbb{M}^i} = \frac{|\Gamma(i)|}{d}(\widetilde{\mathbb{X}^i})^{-1}, \tag{8}$$

where $\widetilde{\mathbb{X}^i} = \frac{d}{|\Gamma(i)|}\sum_{j\in\Gamma(i)} s_{ij}^2 \mathbf{X}^{ij} \otimes \mathbf{X}^{ij}$ is the length distribution tensor and $|\Gamma(i)|$ is the cardinal of $\Gamma(i)$. Let n_{ij} be the number of created nodes in relation with the stretching factor s_{ij} and along the edge ij. When scaling the edges by a factor s_{ij}, the error changes quadratically so that the number of created nodes along the edge ij is given by:

$$n_{ij} = \left(\frac{\widetilde{e_{ij}}}{e_{ij}}\right)^{-\frac{1}{2}} = s_{ij}^{-1}.$$

As per node i, the created nodes along the different edge directions is given by the following tensor:

$$N^i = \left(\frac{d}{|\Gamma(i)|}\sum_{j\in\Gamma(i)} n_{ij}\frac{\mathbf{X}^{ij}}{|\mathbf{X}^{ij}|} \otimes \frac{\mathbf{X}^{ij}}{|\mathbf{X}^{ij}|}\right).$$

So that the total number of created nodes per node i is:

$$n^i = \det\left(\frac{d}{|\Gamma(i)|}\sum_{j\in\Gamma(i)} n_{ij}\frac{\mathbf{X}^{ij}}{|\mathbf{X}^{ij}|} \otimes \frac{\mathbf{X}^{ij}}{|\mathbf{X}^{ij}|}\right).$$

By considering the averaging process of the number of nodes distribution function, the total number of nodes in the adapted mesh is given by

$$N = \sum_i n^i.$$

A direct relation between N and e, assuming a uniform totally balanced error along the edge $\widetilde{e_{ij}} = e =$ constant, is given by:

$$n^{ij}(e) = s_{ij}^{-1}(e) = \left(\frac{e}{e_{ij}}\right)^{-\frac{1}{2}}.$$

Hence, for a node i we have

$$n^i(e) = e^{-\frac{d}{2}}\det\left(\frac{d}{|\Gamma(i)|}\sum_{j\in\Gamma(i)}\left(\frac{1}{e_{ij}}\right)^{-\frac{1}{2}}\frac{\mathbf{X}^{ij}}{|\mathbf{X}^{ij}|} \otimes \frac{\mathbf{X}^{ij}}{|\mathbf{X}^{ij}|}\right) \Longleftrightarrow n^i(e) = e^{-\frac{d}{2}}n^i(1),$$

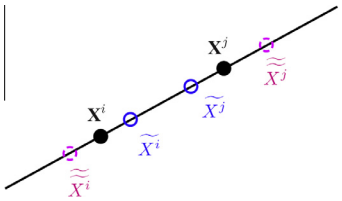

Fig. 3. Varying the edge in its own direction.

so that

$$N = e^{-\frac{d}{2}}\sum_i n^i(1).$$

Therefore, the global induced error for a given total number of nodes N can be determined by:

$$e(N) = \left(\frac{N}{\sum_i n^i(1)}\right)^{-\frac{2}{d}}.$$

Thus, the corresponding stretching factors under the constraint of a fixed number of nodes N are given by:

$$s_{ij} = \left(\frac{e}{e(N)}\right)^{-\frac{1}{2}} = \left(\frac{\sum_i n^i(1)}{N}\right)^{\frac{2}{d}} e_{ij}^{-1/2}.$$

2.6. Extension to multi-component field

Here we propose to construct a unique metric directly from a multi-component vectors field containing, for instance, all the components of the velocity field and/or different level-set functions. Consequently, we do not need to intersect several metrics as in [27] but construct it using the following error vector $\vec{e_{ij}} = \left\{e_{ij}^1, e_{ij}^2, \dots, e_{ij}^n\right\}$ where n is the number of components. Let $u = \{u_1, u_2, \dots, u_n\}, \mathcal{Z} = \mathcal{V}^n$ and $\mathcal{Z}_h = \mathcal{V}_h^n$. In the view of constructing a unique metric, the above theory is applied for each component of u. It comes out immediately that the error is now a vector given by: $\overrightarrow{e_{ij}} = \left\{e_{ij}^1, e_{ij}^2, \dots, e_{ij}^n\right\}$ and then

$$s_{ij} = \left(\frac{\|e(N)\|}{\|\overrightarrow{e_{ij}}\|}\right)^{\frac{1}{2}}.$$

Here, the norm can be the discrete L_2,L_1 or L_∞ norms.

Rather than considering several metric intersections and thus having much computations to perform, we propose herein an easy way to account for different fields in an *a posteriori* analysis while producing a single metric field. We propose then to combine, into one global vector field, both the level-set function and all components of the velocity field in only one metric tensor.

Denote by v_h the finite element solution of the Navier Stokes equations and $\Pi_h v$ its interpolant. In general, we have:

$$\exists c > 0,\ \|v_h - v\| \leqslant c\|\Pi_h v - v\|.$$

Let $v_h(\mathbf{X}^i) = V^i \in \mathbb{R}^d,\ d = 2, 3$ and $\mathcal{Y} = \left(\frac{v}{|v|}, |v|, \alpha\right)$ be the vector field made of $d + 1$ components vector fields, with α the level-set function used to localize an immersed body. We obtain for every node i,

$$\Pi_h\mathcal{Y}(\mathbf{X}^i) = \left(\frac{V^i}{|V^i|}, |V^i|, \alpha\right) = \mathcal{Y}^i.$$

2.7. Mixing laws

The geometry and mechanical properties of each subdomain are characterized by one signed distance function. Once all the subdomains are defined, the mechanical properties can be determined on the whole domain in terms of the level-set function. For the elements crossed by the level-set functions and the their neighbors, fluid–solid mixtures are used to determine the element effective properties. A Heaviside function $H(\alpha)$ for each level-set function is defined by:

$$H(\alpha) = \begin{cases} 1 & \text{if } \alpha > 0, \\ 0 & \text{if } \alpha < 0. \end{cases} \tag{9}$$

In order to achieve a better continuity at the interface [28], the Heaviside function can be smoothed using:

$$H_\varepsilon(\alpha) = \begin{cases} 1 & \text{if } \alpha > \varepsilon, \\ \frac{1}{2}\left(1 + \frac{\alpha}{\varepsilon} + \frac{1}{\pi}\sin\left(\frac{\pi\alpha}{\varepsilon}\right)\right) & \text{if } |\alpha| \leqslant \varepsilon, \\ 0 & \text{if } \alpha < -\varepsilon, \end{cases} \tag{10}$$

where ε is a small parameter such that $\varepsilon = O(h_{\text{im}})$, known as the interface thickness, and h_{im} 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:

$$h_{\text{im}} = \max_{j,l\in K} \nabla\alpha \cdot \mathbf{x}^{jl}, \tag{11}$$

where $\mathbf{x}^{jl} = \mathbf{x}^l - \mathbf{x}^j$ and K is the mesh element which fully or partially belongs to the range of $|\alpha| \leqslant \varepsilon$. 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 $P0$.

3. Governing equations

This section is devoted to the mathematical formulation for a rigid body immersed in an incompressible fluid. The governing equations are considered to be three-dimensional and time-dependent. As the proposed approach is monolithic, a unique constitutive equation will be solved on the whole domain with variable physical properties separated by a prescribed level-set function.

3.1. The Navier–Stokes equations with a rigid body

Let $\Omega \subset \mathbb{R}^d$ be the spatial domain at time $t \in [0,T]$, where d is the space dimension. Let $\partial\Omega$ denote the boundary of Ω. The fluid domain, the solid domain and the interface will be Ω_{f},Ω_{s} and Γ_{im}, respectively. They verify:

$$\overline{\Omega}_{\text{f}} \cup \overline{\Omega}_{\text{s}} = \overline{\Omega} \quad \text{and} \quad \overline{\Omega}_{\text{f}} \cap \overline{\Omega}_{\text{s}} = \Gamma_{\text{im}}.$$

The dynamics of the flow is given by the classical incompressible Navier–Stokes equations, which may be written as

$$\rho_{\text{f}}\ (\partial_t \mathbf{v} + \mathbf{v}\cdot\nabla\mathbf{v}) - \nabla\cdot\boldsymbol{\sigma} = \mathbf{f} \quad \text{in } \Omega_{\text{f}},\ t > 0, \tag{12}$$

$$\nabla\cdot\mathbf{v} = 0 \quad \text{in } \Omega_{\text{f}},\ t > 0, \tag{13}$$

where $\mathbf{v}(\mathbf{x},t)$ is the velocity, $p(\mathbf{x},t)$ is the pressure, ρ_{f} is the fluid density and the Cauchy stress tensor for a Newtonian fluid is given by:

$$\boldsymbol{\sigma} = 2\eta_{\text{f}}\ \boldsymbol{\varepsilon}(\mathbf{v}) - p\ \mathbf{I}_d, \tag{14}$$

where $\mathbf{I}_d$ is the d-dimensional identity tensor and η_{f} is the fluid viscosity. Eqs. (12) and (13) are subject to the boundary and initial conditions

$$\mathbf{v} = \mathbf{v}_{\Gamma,\text{f}} \quad \text{on } \partial\Omega_{\text{f}} \setminus \Gamma_{\text{im}},\ t > 0, \tag{15}$$
$$\mathbf{v} = \mathbf{v}_{\text{im}} \quad \text{on } \Gamma_{\text{im}},\ t > 0, \tag{16}$$
$$\boldsymbol{\sigma}\cdot\mathbf{n} = \mathbf{t}_{\text{im}} \quad \text{on } \Gamma_{\text{im}},\ t > 0, \tag{17}$$
$$\mathbf{v}(\mathbf{x},0) = \mathbf{v}_0(\mathbf{x}) \quad \text{in } \Omega_{\text{f}}, \tag{18}$$

where $\mathbf{v}_{\Gamma,\text{f}}$ is a given velocity boundary condition, $\mathbf{v}_{\text{im}}$ is the velocity at the fluid–solid interface Γ_{im} (the boundary of the immersed body), $\mathbf{n}$ is the outward normal on the solid surface, $\mathbf{t}_{\text{im}}$ the normal stress on this boundary and $\mathbf{v}_0(\mathbf{x})$ is a given initial condition. For simplicity, only Dirichlet-type boundary conditions will be considered on the exterior boundary.

In the present formulation we treat the rigid body as a continuous domain subjected to an additional rigidity constraint. As shown in [19], we may write the equations of motion as the Navier–Stokes equations with this constraint as:

$$\rho_{\text{s}}\ (\partial_t \mathbf{v} + \mathbf{v}\cdot\nabla\mathbf{v}) - \nabla\cdot\boldsymbol{\sigma} = \mathbf{f} \quad \text{in} \quad \Omega_{\text{s}},\ t > 0, \tag{19}$$

$$\nabla\cdot\mathbf{v} = 0 \quad \text{in } \Omega_{\text{s}}, t > 0, \tag{20}$$
$$\boldsymbol{\varepsilon}(\mathbf{v}) = \mathbf{0} \quad \text{in } \Omega_{\text{s}}, t > 0, \tag{21}$$

where ρ_{s} the solid density. In a rigid body there is no deformation, that is to say, $\boldsymbol{\varepsilon}(\mathbf{u}) = \mathbf{0}$ ($\mathbf{u}$ is the displacement field) and $\partial_t\mathbf{u} = \mathbf{v}$. These two equations imply a null value of the deformation-rate tensor (21). The stress tensor is then given by:

$$\boldsymbol{\sigma} = \boldsymbol{\tau}_{\text{s}} - p\,\mathbf{I}_d. \tag{22}$$

Eqs. (19)–(21) need to be supplied with the boundary and initial conditions

$$\mathbf{v} = \mathbf{v}_{\Gamma,\text{s}} \quad \text{on } \partial\Omega_{\text{s}} \setminus \Gamma_{\text{im}},\ t > 0, \tag{23}$$
$$\mathbf{v} = \mathbf{v}_{\text{im}} \quad \text{on } \Gamma_{\text{im}},\ t > 0, \tag{24}$$
$$\boldsymbol{\sigma}\cdot\mathbf{n} = -\mathbf{t}_{\text{im}} \quad \text{on } \Gamma_{\text{im}},\ t > 0, \tag{25}$$
$$\mathbf{v}(\mathbf{x},0) = \mathbf{v}_0(\mathbf{x}) \quad \text{in } \Omega_{\text{s}}, \tag{26}$$

where $\mathbf{v}_{\Gamma,\text{f}}$ is a given velocity boundary condition that needs to be compatible with a rigid body motion if $\partial\Omega_{\text{s}}\setminus\Gamma_{\text{im}} \neq \emptyset$, and the initial condition $\mathbf{v}_0(\mathbf{x})$ must be also compatible with a rigid body motion.

3.2. Full Euleurian formulation

Making use of the notation introduced in Section 1, we may write problem (12)–(18) and problem (19)–(26) in a unified way in the whole computational domain Ω as

$$\begin{aligned} &\rho\ (\partial_t\mathbf{v} + \mathbf{v}\cdot\nabla\mathbf{v}) - \nabla\cdot(2\eta\ \boldsymbol{\varepsilon}(\mathbf{v}) + \boldsymbol{\tau} - p\ \mathbf{I}_d) = \mathbf{f} \quad \text{in } \Omega,\ t > 0 \\ &\nabla\cdot\mathbf{v} = 0 \quad \text{in } \Omega,\ t > 0 \\ &\boldsymbol{\varepsilon}_{\text{s}}(\mathbf{v}) = \mathbf{0} \quad \text{in } \Omega,\ t > 0 \end{aligned} \tag{27}$$

where $\mathbf{v}_\Gamma = \mathbf{v}_{\Gamma,\text{s}}$ on $\partial\Omega_{\text{s}} \cap \partial\Omega$ and $\mathbf{v}_\Gamma = \mathbf{v}_{\Gamma,\text{f}}$ on $\partial\Omega_{\text{f}} \cap \partial\Omega$, $\boldsymbol{\varepsilon}_{\text{s}}(\mathbf{v}) = H(\alpha)\boldsymbol{\varepsilon}(\mathbf{v})$, $\eta = (1 - H(\alpha))\eta_{\text{f}}$, $\rho = \rho_{\text{s}}H(\alpha) + \rho_{\text{f}}(1 - H(\alpha))$ and $\boldsymbol{\tau} = H(\alpha)\boldsymbol{\tau}_{\text{s}}$. The boundary conditions (16,17,24 and 25) are no longer needed.

Let $V \times P \times \mathcal{T}$ be the space where the unknown $(\mathbf{v}, p, \boldsymbol{\tau})$ is sought. The first space, V, is made of vector fields which are square integrable in time with values in $H^1(\Omega)^d$ and satisfying the Dirichlet conditions, where the last two, P and $\mathcal{T}$, are made of distributions in time with values in $P_0 = L^2(\Omega)/\mathbb{R}$ and $\mathcal{T}_0 = L^2(\Omega)^{d\times d}$, respectively (in fact, a subspace of $L^2(\Omega)^{d\times d}$ would be enough, see below). The corresponding test functions will be denoted $\mathbf{w} \in V_0 = H_0^1(\Omega)^d$, $q \in P_0$ and $\boldsymbol{\xi} \in \mathcal{T}_0$. Multiplying by the test functions and integrating by parts, the associated standard weak form of the system (27), can be stated as: Find $\mathbf{v} \in V$, $p \in P$ and $\boldsymbol{\tau} \in \mathcal{T}$ such that

$$\begin{aligned} &\rho(\partial_t\mathbf{v}, \mathbf{w}) + \rho(\mathbf{v}\cdot\nabla\mathbf{v}, \mathbf{w}) - (p, \nabla\cdot\mathbf{w}) + (2\eta\boldsymbol{\varepsilon}(\mathbf{v}), \boldsymbol{\varepsilon}(\mathbf{w})) + (\boldsymbol{\tau}, \boldsymbol{\varepsilon}_{\text{s}}(\mathbf{w},))) \\ &\quad = \langle \mathbf{f}, \mathbf{w}\rangle, \\ &(q, \nabla\cdot\mathbf{v}) = 0, \\ &-(\boldsymbol{\xi}, \boldsymbol{\varepsilon}_{\text{s}}(\mathbf{v})) = 0 \end{aligned} \tag{28}$$

for all $(\mathbf{w}, q, \boldsymbol{\xi}) \in V_0 \times P_0 \times \mathcal{T}_0$.

A possible way to choose $\boldsymbol{\tau}$ is to take it as a symmetric gradient of a vector field. Moreover, this field needs not to be computed if an augmented Lagrangian scheme together with an Uzawa iterative scheme are employed to relax iteratively $(\boldsymbol{\xi}, \boldsymbol{\varepsilon}_{\text{s}}(\mathbf{v})) = 0$. This is what we describe next.

Suppose that we discretize in time problem (28) using a finite difference scheme, and still denote by $\mathbf{v}$, p and $\boldsymbol{\tau}$ the fields to be computed at a given time step. Let $\delta_t\mathbf{v}$ the discrete time derivative and r the penalty parameter in the Uzawa scheme. Treating implicitly the velocity in the calculation of the stress in the solid, the iterative scheme to be performed *within each time step* reads:

1. Set $k = 0$
2. Initialize $\mathbf{v}^0$, p^0 and $\boldsymbol{\tau}^0$ (for example to values in the last time step)
3. $k \leftarrow k + 1$
4. Solve for $\mathbf{v}^k$ and p^k:

$$\rho(\delta_t \mathbf{v}^k, \mathbf{w}) + \rho(\mathbf{v}^k \cdot \nabla \mathbf{v}^k, \mathbf{w}) - (p^k, \nabla \cdot \mathbf{w}) + 2(\eta \boldsymbol{\varepsilon}(\mathbf{v}^k) + H(\alpha)\eta_s \; r \; P_\tau(\boldsymbol{\varepsilon}(\mathbf{v}^k)), \boldsymbol{\varepsilon}(\mathbf{w})) + (\boldsymbol{\tau}^{k-1}, \boldsymbol{\varepsilon}_s(\mathbf{w})) = \langle \mathbf{f}, \mathbf{w} \rangle,$$

$$(q, \nabla \cdot \mathbf{v}^k) = 0.$$

5. Update $\boldsymbol{\tau}^k = \boldsymbol{\tau}^{k-1} + 2\eta_s \; r \; P_\tau(\boldsymbol{\varepsilon}(\mathbf{v}^k))$ in Ω_s.
6. Check convergence: if $\|\mathbf{v}^k - \mathbf{v}^{k-1}\| > \texttt{tol}$ (given tolerance in a given norm), go to 3. Otherwise, proceed to the next time step.

3.3. Rigid body kinetics

For the rigid body motion, when the geometry is simple and the distance function can be computed analytically, it can be sufficient to calculate the optimal angular velocity $\boldsymbol{\omega}$ and the translational velocity $\mathbf{V}$. In fact, once the Navier–Stokes is solved, the velocity $\mathbf{v}_h$ is computed at each a point $\mathbf{x}$ of the domain. $\boldsymbol{\omega}$ and $\mathbf{V}$ are computed by minimizing $\varphi(\mathbf{V},\boldsymbol{\omega})$ defined by:

$$\varphi(\mathbf{V}, \boldsymbol{\omega}) = \int_{\Omega_s} |\mathbf{V} + \boldsymbol{\omega} \wedge \mathbf{x} - \mathbf{v}_h|^2. \tag{29}$$

Note that if $\mathbf{v}_h$ were the true rigid body velocity (as the third equation in (28) imposes), the minimum of φ would be 0. The rigid body position will then be updated as follows:

$$\mathbf{X}^n = \mathbf{X}^o + \Delta t(\mathbf{V} + \boldsymbol{\omega} \wedge \mathbf{X}^o), \tag{30}$$

where $\mathbf{X}^n$ and $\mathbf{X}^o$ are the new and old coordinates of any point belonging to the rigid body respectively.

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 Eq. (28). Based on a mesh $\mathcal{K}_h$ of Ω made of N_{el} elements K, the functional spaces for the velocity, the pressure and the stress are approximated by the finite dimensional spaces V_h, P_h and $\mathcal{T}_h$, respectively.

The stability of the discrete formulation depends on appropriate compatibility restrictions on the choice of the FE spaces. We propose here a Variational Multiscale method (see [29]), which allows the use of equal order continuous interpolations for the three fields, apart from preventing oscillations due to convection dominated flows [30].

Let us split the velocity, pressure and stress solution spaces as $V_h \oplus V'$,$P_h \oplus P'$ and $\mathcal{T}_h \oplus \mathcal{T}'$, respectively. Subscript h is used here and in the following to denote the FE (coarse) component, whereas the prime is used for the so called subgrid scale (fine) component of the unknowns. According to this, we have

$$\mathbf{v} = \mathbf{v}_h + \mathbf{v}' \in V_h \oplus V',$$

$$p = p_h + p' \in P_h \oplus P',$$

$$\boldsymbol{\tau} = \boldsymbol{\tau}_h + \boldsymbol{\tau}' \in \mathcal{T}_h \oplus \mathcal{T}'.$$

If the spaces for the test functions are split likewise, with a subscript 0 to identify them, problem (28) becomes: find $(\mathbf{v}_h + \mathbf{v}', p_h + p', \boldsymbol{\tau}_h + \boldsymbol{\tau}') \in V_h \oplus V' \times P_h \oplus P' \times \mathcal{T}_h \oplus \mathcal{T}'$ such that

$$\rho(\delta_t(\mathbf{v}_h + \mathbf{v}'), \mathbf{w}_h + \mathbf{w}') + \rho((\mathbf{v}_h + \mathbf{v}') \cdot \nabla(\mathbf{v}_h + \mathbf{v}'), \mathbf{w}_h + \mathbf{w}') - (p_h + p', \nabla \cdot (\mathbf{w}_h + \mathbf{w}')) + 2(\eta \boldsymbol{\varepsilon}(\mathbf{v}_h + \mathbf{v}'), \boldsymbol{\varepsilon}(\mathbf{w}_h + \mathbf{w}')) + (\boldsymbol{\tau}_h + \boldsymbol{\tau}', \boldsymbol{\varepsilon}_s(\mathbf{w}_h + \mathbf{w}')) = \langle \mathbf{f}, \mathbf{w}_h + \mathbf{w}' \rangle, \tag{31}$$

$$(q_h + q', \nabla \cdot (\mathbf{v}_h + \mathbf{v}')) = 0, \tag{32}$$

$$-(\boldsymbol{\xi}_h + \boldsymbol{\xi}', \boldsymbol{\varepsilon}_s(\mathbf{v}_h + \mathbf{v}')) = 0, \tag{33}$$

for all $(\mathbf{w}_h + \mathbf{w}', q_h + q', \boldsymbol{\xi}_h + \boldsymbol{\xi}') \in V_{h,0} \oplus V'_0 \times P_{h,0} \oplus P'_0 \times \mathcal{T}_{h,0} \oplus \mathcal{T}'_0$. Recall that δ_t stands for an approximation to the time derivative ∂_t. Even if time has been discretized, we have kept the notation for the functional spaces for simplicity.

Even though the subgrid scales (or subscales) could be approximated without further assumptions and inserted into the previous equations (see [31]), we will make use of some common approximations that are explained in [19] and lead to the discrete variational problem:

$$\rho(\delta_t \mathbf{v}_h, \mathbf{w}_h) + \rho(\mathbf{v}_h \cdot \nabla \mathbf{v}_h, \mathbf{w}_h) - (p_h + p', \nabla \cdot \mathbf{w}_h) + 2(\eta \boldsymbol{\varepsilon}(\mathbf{v}_h), \boldsymbol{\varepsilon}(\mathbf{w}_h)) + (\boldsymbol{\tau}_h + \boldsymbol{\tau}', \boldsymbol{\varepsilon}_s(\mathbf{w}_h)) + \sum_K (\mathbf{v}', -\rho \mathbf{v}_h \cdot \nabla \mathbf{w}_h - \nabla \cdot (2\eta \boldsymbol{\varepsilon}(\mathbf{w}_h)))_K = \langle \mathbf{f}, \mathbf{w}_h \rangle, \tag{34}$$

$$(q_h, \nabla \cdot \mathbf{v}_h) - \sum_K (\mathbf{v}', \nabla q_h)_K = 0, \tag{35}$$

$$-(\boldsymbol{\varepsilon}_s(\mathbf{v}_h), \boldsymbol{\xi}_h) + \sum_K (\mathbf{v}', \chi_s \nabla \cdot \boldsymbol{\xi}_h)_K = 0, \tag{36}$$

for all $(\mathbf{w}_h, q_h, \boldsymbol{\xi}_h) \in V_{h,0} \times P_{h,0} \times \mathcal{T}_{h,0}$, where $\sum_K$ stands for the summation over all the elements of the finite element partition $\mathcal{K}_h$ and $(\cdot,\cdot)_K$ denotes the L^2 product in each K.

The problem for the fine scales is obtained taking $(\mathbf{w}_h, q_h, \boldsymbol{\xi}_h) = (\mathbf{0}, 0, \mathbf{0})$ in (31)–(33). Introducing the finite element residuals

$$\mathcal{R}_v = \mathbf{f} - \rho \delta_t \mathbf{v}_h - \rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h - \nabla p_h + \chi_s \nabla \cdot \boldsymbol{\tau}_h + \nabla \cdot (2\eta \boldsymbol{\varepsilon}(\mathbf{v}_h)),$$

$$\mathcal{R}_p = -\nabla \cdot \mathbf{v}_h,$$

$$\mathcal{R}_\tau = \boldsymbol{\varepsilon}_s(\mathbf{v}_h)$$

and using the same ideas as in [32,33], it turns out that the subscales may be approximated within each element $K \in \mathcal{K}_h$ by

$$\mathbf{v}' = \alpha_v(\mathcal{R}_v), \quad p' = \alpha_p(\mathcal{R}_p), \quad \boldsymbol{\tau}' = \alpha_\tau(\mathcal{R}_\tau),$$

where α_v, α_p and α_τ are the so called stabilization parameters that we compute within each element as

$$\alpha_v = \left[\left(\frac{c_1 \eta}{\rho h^2} \right)^2 + \left(\frac{c_2 \|\mathbf{v}_h\|_K}{h} \right)^2 \right]^{-1/2}, \tag{37}$$

$$\alpha_p = \left[\left(\frac{\eta}{\rho} \right)^2 + \left(\frac{c_2 \|\mathbf{v}_h\|_K h}{c_1} \right)^2 \right]^{1/2}, \tag{38}$$

$$\alpha_\tau = c_3 \frac{h}{L} 2\eta_s, \tag{39}$$

where h is the element size, L a characteristic length of the computational domain, $\|\mathbf{v}\|_K$ a characteristic norm of $\mathbf{v}_h$ (with the same units as $\mathbf{v}_h$) in element K and c_1, c_2 and c_3 are algorithmic constants. We take them as $c_1 = 4$, $c_2 = 2$ and $c_3 = 1$ for linear elements. For the linear elements used in the numerical examples, terms of the form $\nabla \cdot (2\eta \boldsymbol{\varepsilon}(\mathbf{w}_h))$ involving second derivatives within each element can be neglected.

Note that the calculation of h is one of the main subjects of this paper. Indeed, the stability coefficients depend on the local mesh size h and weight the extra terms added to the weak formulation. In the case of strongly anisotropic meshes with highly stretched elements, the definition of h is still an open problem and plays a critical role in the design of the stabilizing coefficients. We followed the lines in [34] to compute h as the diameter of K in the direction of the velocity $\mathbf{v}_h$ as follows (see Fig. 4):

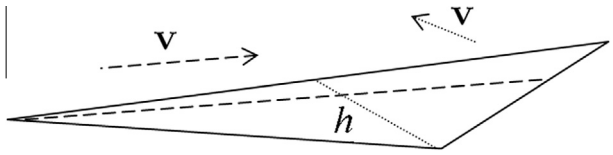

Fig. 4. Longest triangle length in the streamline direction.

$$h = \frac{2|\mathbf{v}_h|}{\sum_{i=1}^{N_K} |\mathbf{v}_h \cdot \nabla \varphi_i|}, \tag{40}$$

where N_K is the number of vertices of K and $\varphi_1, \ldots, \varphi_{N_K}$ are the usual basis functions of $P_1(K)$ mapped onto K.

Inserting the expression for the subscales obtained in (34)–(36) we finally obtain the needed stabilized finite element problem. It consists of finding $(\mathbf{v}_h, p_h, \boldsymbol{\tau}_h) \in V_h \times P_h \times \mathcal{T}_h$ such that

$$\begin{aligned}
&\rho(\delta_t \mathbf{v}_h, \mathbf{w}_h) + \rho(\mathbf{v}_h \cdot \nabla \mathbf{v}_h, \mathbf{w}_h) - (p_h, \nabla \cdot \mathbf{w}_h) + 2(\eta \boldsymbol{\varepsilon}(\mathbf{v}_h), \boldsymbol{\varepsilon}(\mathbf{w}_h)) \\
&\quad + (\boldsymbol{\tau}_h, \boldsymbol{\varepsilon}_s(\mathbf{w}_h)) + \sum_K \alpha_v (\rho \delta_t \mathbf{v}_h + \rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h - \chi_s \nabla \cdot \boldsymbol{\tau}_h \\
&\quad - \nabla \cdot (2\eta \boldsymbol{\varepsilon}(\mathbf{v}_h)), \rho \mathbf{v}_h \cdot \nabla \mathbf{w}_h + \nabla \cdot (2\eta \boldsymbol{\varepsilon}(\mathbf{w}_h)))_K \\
&\quad + \sum_K \alpha_p (\nabla \cdot \mathbf{v}_h, \nabla \cdot \mathbf{w}_h) + \sum_K \alpha_\tau (\boldsymbol{\varepsilon}_s(\mathbf{v}_h), \boldsymbol{\varepsilon}_s(\mathbf{w}_h)) = \langle \mathbf{f}, \mathbf{w}_h \rangle \\
&\quad + \sum_K \alpha_v (\mathbf{f}, \rho \mathbf{v}_h \cdot \nabla \mathbf{w}_h + 2\eta \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{w}_h))_K,
\end{aligned} \tag{41}$$

$$\begin{aligned}
&(q_h, \nabla \cdot \mathbf{v}_h) + \sum_K \alpha_v (\rho \delta_t \mathbf{v}_h + \rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h - \chi_s \nabla \cdot \boldsymbol{\tau}_h - \nabla \\
&\quad \cdot (2\eta \boldsymbol{\varepsilon}(\mathbf{v}_h)), \nabla q_h)_K = \sum_K \alpha_v (\mathbf{f}, \nabla q_h)_K,
\end{aligned} \tag{42}$$

$$\begin{aligned}
&-(\boldsymbol{\varepsilon}_s(\mathbf{v}_h), \xi_h) + \sum_K \alpha_v (\rho \delta_t \mathbf{v}_h + \rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h - \chi_s \nabla \cdot \boldsymbol{\tau}_h - \nabla \\
&\quad \cdot (2\eta \boldsymbol{\varepsilon}(\mathbf{v}_h)), -\chi_s \nabla \cdot \xi_h)_K = \sum_K \alpha_v (\mathbf{f}, -\chi_s \nabla \cdot \xi_h)_K
\end{aligned} \tag{43}$$

for all $(\mathbf{w}_h, q_h, \xi_h) \in V_{h,0} \times P_{h,0} \times \mathcal{T}_{h,0}$. We have assumed $\mathbf{f} \in L^2(K)^n$ for simplicity.

At this point the problem suffers from the lack of an appropriate choice for $\mathcal{T}_h$ to make system (41)–(43) uniquely solvable. This can be circumvented by using a sort of augmented Lagrangian scheme coupled with an Uzawa iterative scheme, as explained in the algorithm of Section 2. The final fully discrete iterative scheme is described in [19] and will be not repeated here.

5. Numerical experiments

In this section, we present five numerical examples to illustrate the flexibility of the approach dealing with complex geometry and to asses its accuracy. The numerical simulations were carried out using the C++ CimLib finite element library (see [35,36]). The first obtained results with the proposed approach are compared to solutions obtained by standard solutions (classical boundary conditions).

Table 1
Parameter used in the computation of a falling disk in a channel.

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

5.1. Falling disk in a channel

We consider first a classical benchmark: a rigid disk with radius $R = 0.125$ cm falling under the action of gravitational force inside a 2-D channel of dimension $[0,2] \times [0,6]$. Parameters used in this example are tabulated in Table 1.

In [37] the velocity of particulate flow with rigid circular disks using the fictitious domain method is calculated and reported. 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 Fig. 5 are found for both the velocity and the position of the center of the disk with respect to the time. Figs. 6 and 7 illustrate respectively the velocity contour plot for v_x and v_y at selected times surrounding the zero iso-zero of the disk level-set function. The agreement between the two numerical solutions shows that the present approach is able to predict well the behavior of the fluid in the presence of rigid body.

5.2. Falling cylinder in an incompressible fluid

We follow the lines in [38] to analyze extensively the terminal velocity v_T of a falling cylinder in an incompressible fluid. The same parameters as [38] are used to assess the solution using for instance different viscosity values and different meshes. The rigid cylinder falls under the gravitational force. We prescribe then a zero pressure on the top of the fluid channel and no slip walls on the sides and bottom. The dimension of the fluid domain is $2L \times 8L$ with $L = 0.02$m and the cylinder radius is $r = 5$mm. The fluid is considered incompressible with density 1000 kg/m^3 and the solid density is 2000 kg/m^3. The graviational force is 9.8m s^{-1}. Note that for comparisons, we chose from the reference the most precise results computed using a 160×640 grid. The values of the viscosity are: $\eta_f = 0.1$ kg s^{-1}, 0.2 kg s^{-1}, 0.5 kg s^{-1}, 1 kg s^{-1}, 2 kg s^{-1}, 5kg s^{-1} and 10 kg s^{-1}.

All the computed data are tabulated in Table 2. Note that we are solving the Navier–Stokes equations while the terminal velocity is derived under a Stokes flow assumption. As a result, the terminal

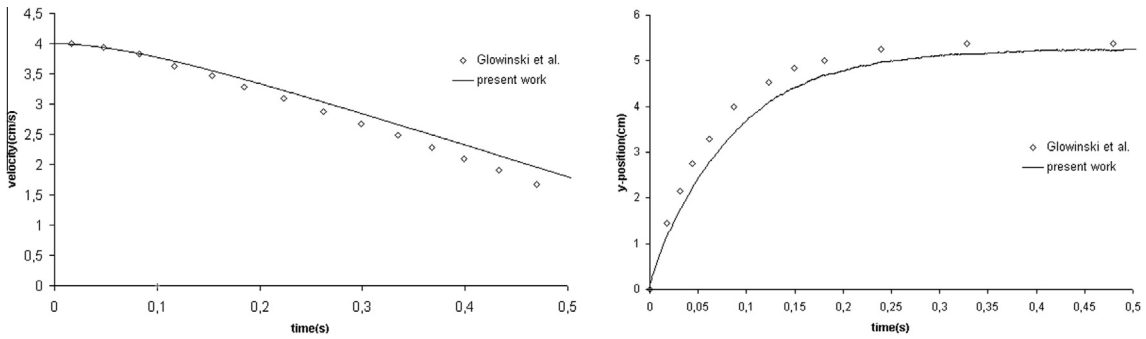

Fig. 5. Evolution of the vertical velocity and position of the falling disk.

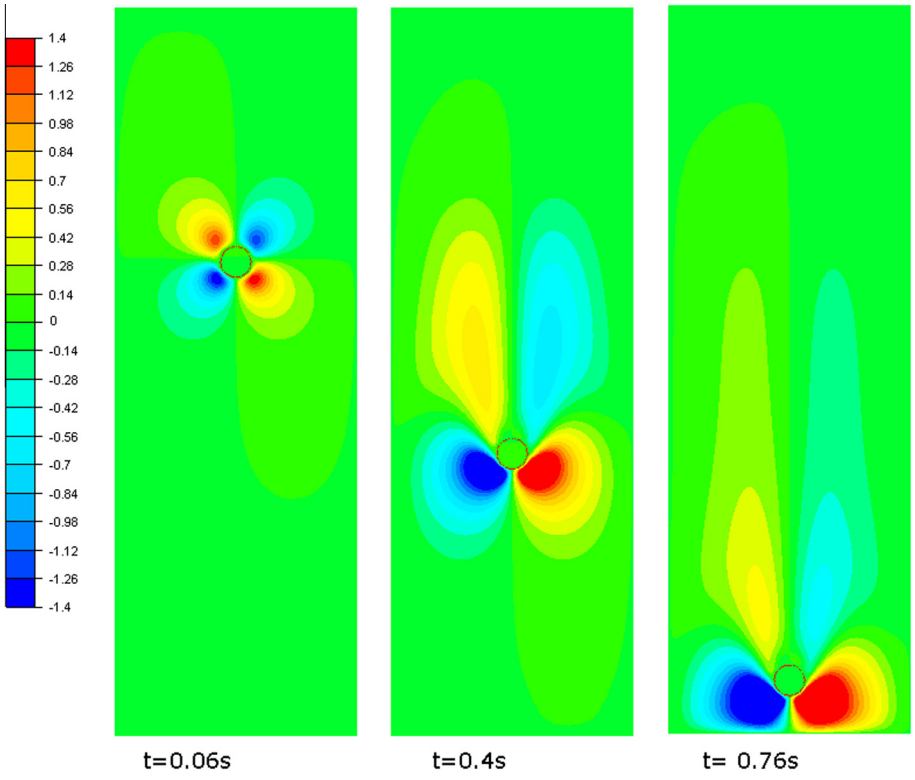

Fig. 6. Profile of the velocity v_x surrounding the iso-zero of the disk level-set function.

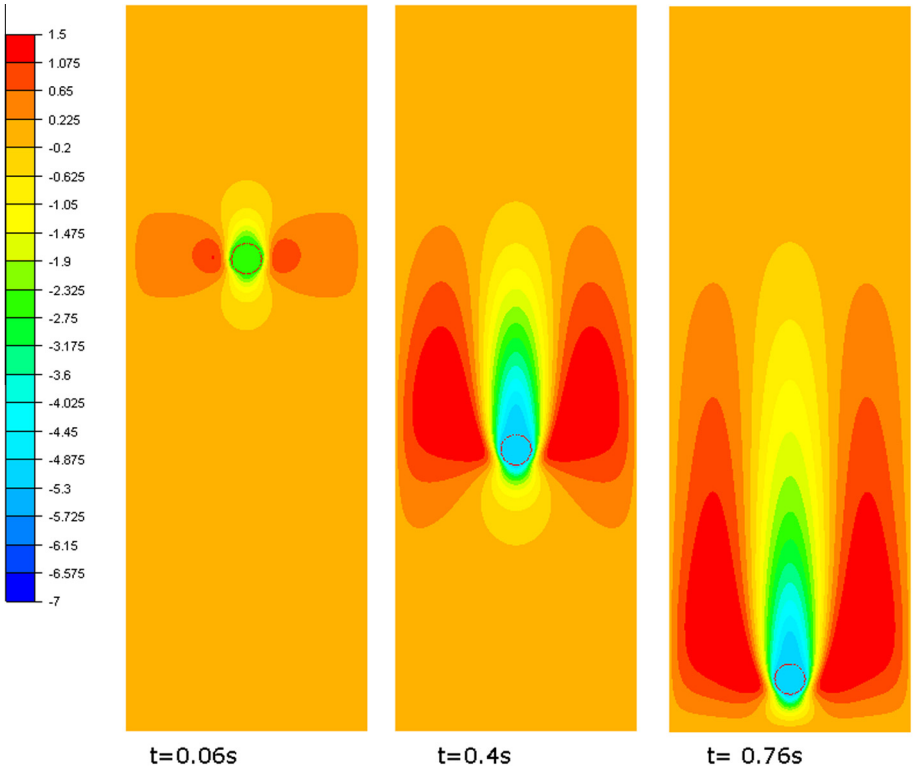

Fig. 7. Profile of the velocity v_y surrounding the iso-zero of the disk level-set function.

velocity calculated will be usually lower than the analytic solution. As observed in Fig. 8 and in Table 2, we obtain a closer agreement on the finest mesh. Robinson et al. [38] solve the FSI problem in a monolithic framework using a MAC grid discretization of the fluid and a fully Lagrangian discretization of the structure. Comparing our computational results to those reported in [38], we can find

Table 2
Terminal velocity for the falling cylinder problem.

η_s	Ref. [38] 160×640	$h = 0.001$	$h = 0.0005$	$h = 0.00025$
0.1	−0.1966	−0.167770	−0.19230704	−0.196834
0.2	−0.1417	−0.12367	−0.13596088	−0.14087306
0.5	−0.06721	−0.059924	−0.06253623	−0.066432
1	−0.03399	−0.030207	−0.03266043	−0.0335803
2	−0.01702	−0.01523	−0.0160292	−0.0165884
5	−0.006828	−0.00617	−0.0066192	−0.00674716
10	−0.003417	−0.003104	−0.0032519	−0.00338712

good agreements for the terminal velocities and a good potential for the developed formulation.

5.3. Tetris benchmark

Four rigid bodies with different densities are falling under the gravitational force in a $[0,2] \times [0,6]$ channel. When several rigid bodies fall in an incompressible fluid channel, the interactive motions of these bodies show an interesting phenomena. At the beginning, each body has the same acceleration due to the gravitational force. As time passes, the velocity of the upper bodies becomes faster than the lower ones since they undergo more resistance against the fluid comparing to the upper ones.

The coordinates of the lower left corner of each solid, the corresponding geometries as well as the densities are given in Table 3 and presented in Figs. 10 and 11. The arrangement of the four rigid bodies at $t = 0$ and the used refined FE mesh are shown in Fig. 9. If we consider that we are at position A and the mesh is well refined around the objects, then this position and this mesh can be considered as a distorted mesh in respect to the new position B. So, basically, the mesh is coarsened around the last position (A) and refined around the new position (B) while keeping the same number of elements. Indeed, to track the bodies all along the simulation and to render a well respected geometry in terms of angles and accurate interfaces, the proposed mesh adaptation algorithm is applied iteratively. Consequently, information is transferred from one mesh to another constructed mesh. Even though there are different methods to improve mesh-to-mesh data transfer, the induced error is in principle of the adequate order and we have not noticed any accuracy misbehavior. The number of elements is fixed to 35,000. The density and viscosity of the fluid used in this example are $\rho_f = 1$ kg/m^3 and $\eta_f = 0.005$ Pa s.

The objective of this test is to show the capability of the method to handle high density ratios. Fig. 12 illustrates the respected geometry due to the use of the anisotropic mesh

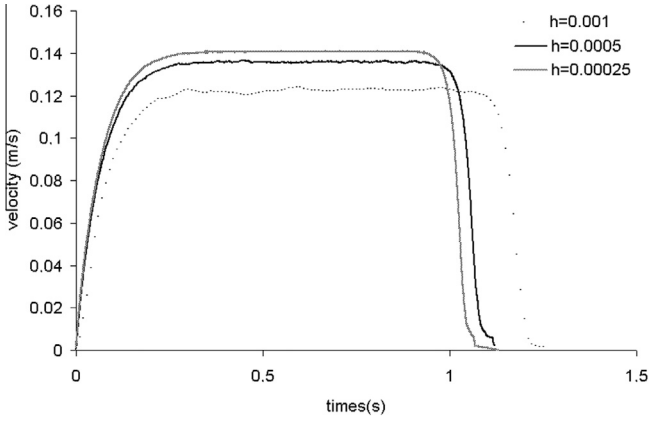

Fig. 8. The evolution of the terminal velocity of the cylinder for $\eta_s = 0.2$ kg/s and $h = 0.001$, $h = 0.0005$ and $h = 0.00025$.

Table 3
The properties of the four rigid bodies.

Solid	Initial position	ρ_s (kg/m^3)
a	(0.35; 4)	4000
b	(0.32; 5)	8000
c	(1; 5.6)	3000
d	(1.25; 4.8)	8000

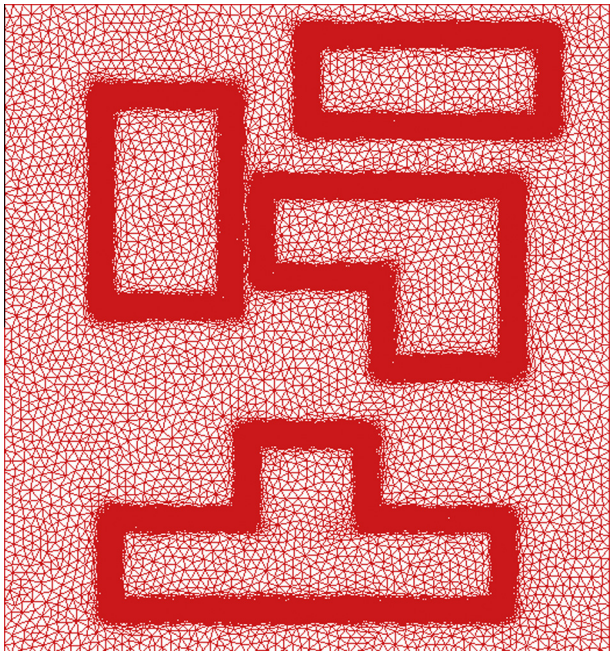

Fig. 9. Zoom on the upper part of the domain showing the initial used finite element mesh.

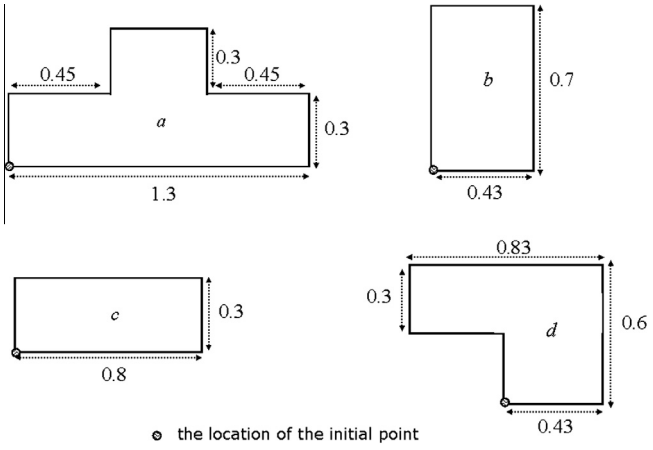

Fig. 10. The proposed geometry for the four rigid bodies.

adaptation and the velocity vectors at different time instants. All the vortices behind the solid objects confirm the capability of the proposed stabilized monolithic formulation to solve at the same time a convection dominated flow in the fluid and a rigid body velocity in the solid. The evolution of the velocity and the corresponding position taken at the center of the body *a* are plotted in Fig. 13 for comparison. We acknowledge that the solution of this test case is direct and similar to the previous numerical examples without assessing the interaction between the bodies. The purpose

is merely to demonstrate the capability of a high density difference and the use of low fluid viscosity that the proposed monolithic formulation can handle.

5.4. 2D immersion of a NACA0012 airfoil in an incompressible fluid at Reynolds 5000

We continue by investigating the flow around a NACA0012 airfoil in a channel [39]. This study is considered as an important step to investigate the feasibility of the proposed monolithic fluid–structure formulation. The purpose is to show the flexibility of the method to deal with a large variety of geometries. Rather than spending the effort on the mesh construction around the airfoil, we can bypass this step and we consider the simplest rectangular domain. As mentioned before, the NACA profile will be represented then using a simple distance function.

Therefore, we consider two cases. In the first one, we use the classical approach and we impose zero boundary conditions on the profile. So the effort will be concentrated on the geometry and on building the fluid mesh while well respecting the curvatures of the airfoil profile. In the second case, (i) we consider a large simple channel domain, (ii) we compute analytically the distance function of the NACA profile located at the center, (iii) we apply the anisotropic mesh adpatation using the variation of the gradients of the level-set function, and finally, (iv) we mix and assign the physical properties. The obtained finite element meshes that will be used in the two cases are depicted in Fig. 14.

The proposed *a posteriori* estimation combines the simultaneous adaptivity on the level-set scalar field and the velocity field. Since these fields are highly directional (in particular at the interface) the mesh technique shows that it is capable of changing completely the mesh (coarsening in some regions and refining in other

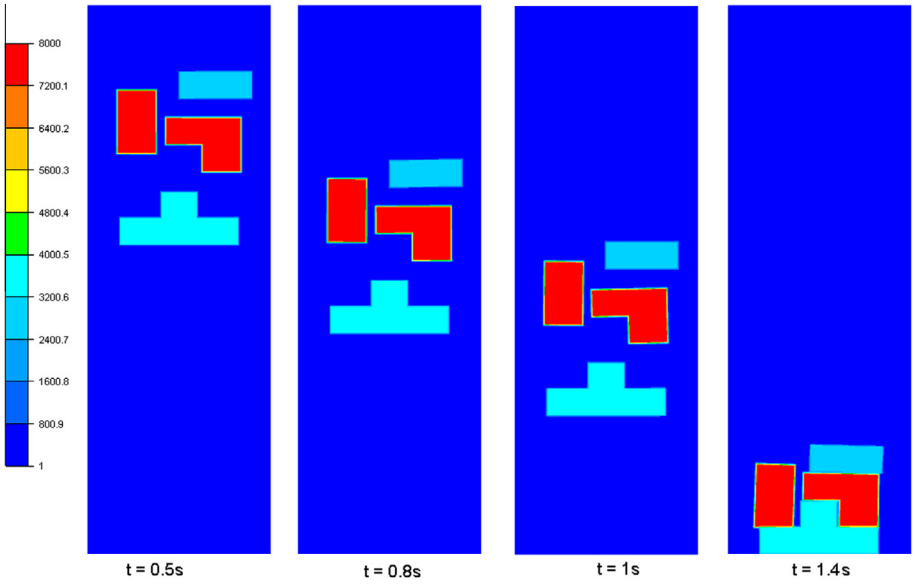

Fig. 11. Density distribution at different time instants with anisotropic adapted interfaces.

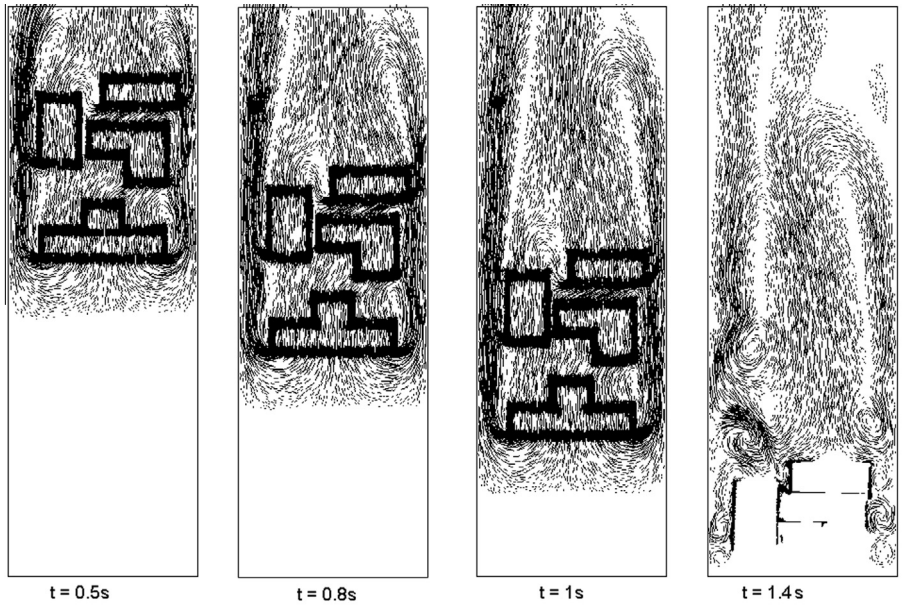

Fig. 12. Velocity vectors at different time.

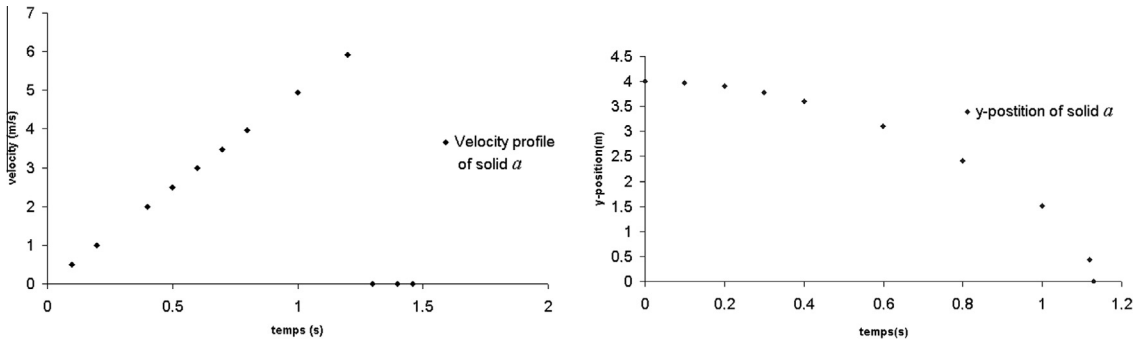

Fig. 13. Evolution of the velocity and the position taken at the center of the body *a*.

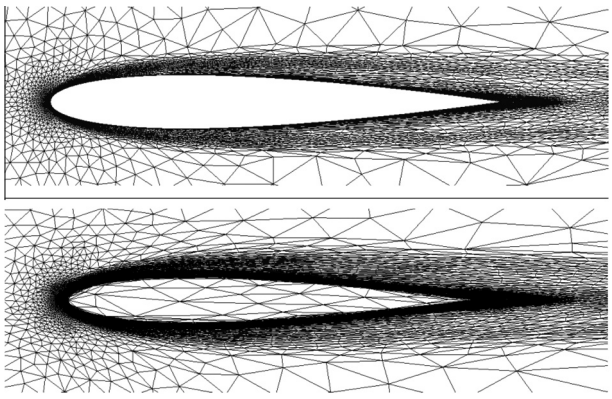

Fig. 14. Finite element meshes: only the fluid domain (up) – the air-NACA domain (bottom).

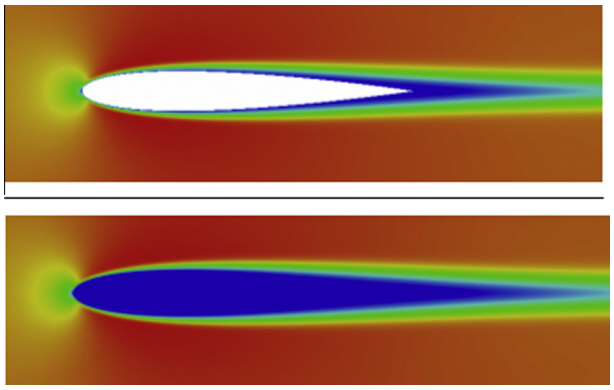

Fig. 16. Velocity Norm calculated over the fluid domain (up) and the entire domain (bottom).

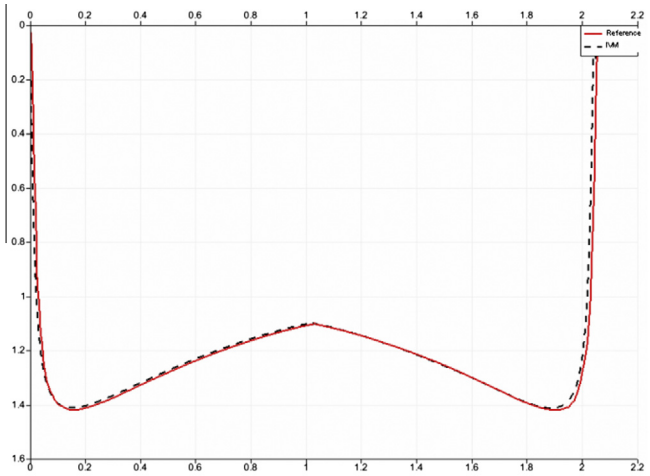

Fig. 15. Comparisons of drag coefficients. (–) reference, (···) present work.

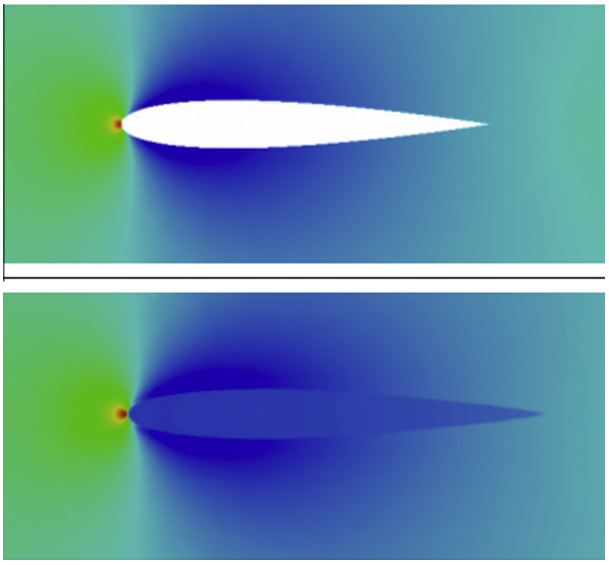

Fig. 17. Pressure distribution calculated over the fluid domain (up) and the entire domain (bottom).

regions) in an anisotropic way and at a low number of elements (i.e. the elements are stretched in a certain direction). Consequently, boundary layers and inner layers are automatically captured due to the anisotropically adapted mesh exhibiting highly stretched elements.

We apply the same conditions on both test cases and we compare the solutions. The Reynolds number Re_∞ based on the cord c is equal to 5000 and the angle of incidence to 0. The number of elements is fixed to 45,000.

A comparison of the drag coefficients using the classical and the new approach is presented in Fig. 15. Snapshots of the norm of the velocity, the pressure fields and the streamlines computed on the entire domain (fluid and structure) and on only the fluid domain are shown in Figs. 16–18 respectively. The very good agreement between the two numerical solutions shows that the present solver is able to predict accurately the behavior of the fluid and the presence of the solid.

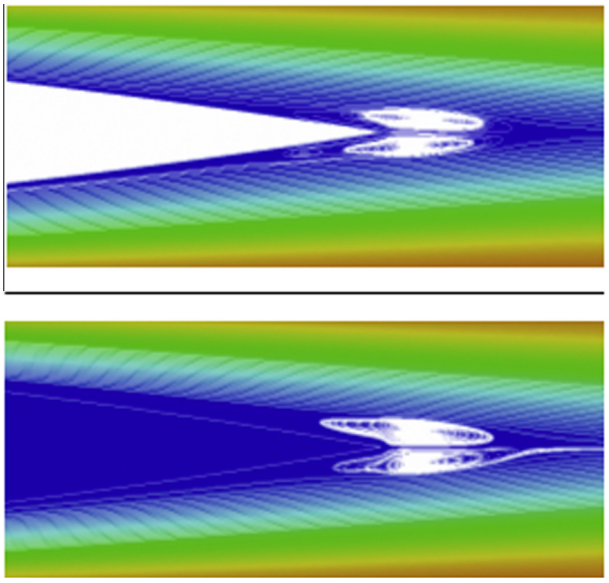

Fig. 18. Streamlines distribution calculated over the fluid domain (up) and the entire domain (bottom).

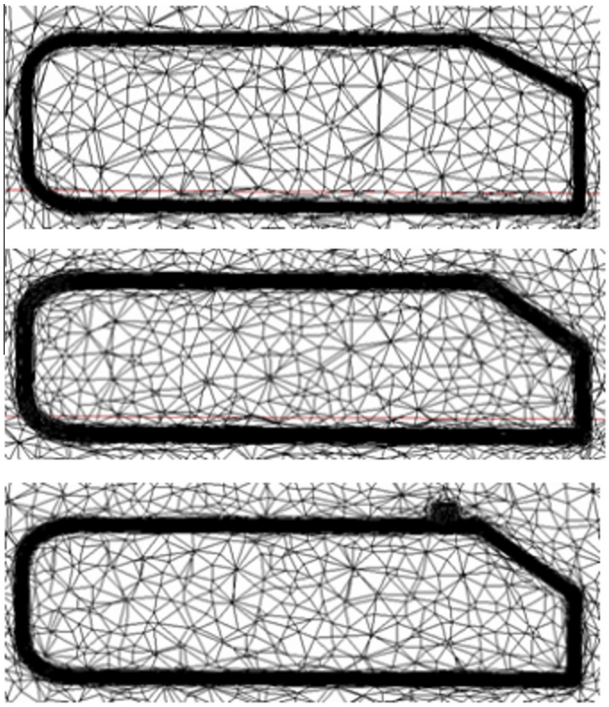

Fig. 19. Remeshing the fluid–solid interfaces and modifying the rear angle from 25° (up) to 35° (center) and adding a small cylinder on the top (bottom).

5.5. Unsteady flow past a 3D immersed simplified vehicle model

A similar test case is aimed at exploring the capabilities of the model when used in a situation involving more complex geometries in 3D. Here, we simulate the flow past an immersed simplified vehicle model. The Ahmed body is one of the first benchmark proposed to investigate the stability and fuel consumption of an automobile at high cruising speeds. It represents a simplified car geometry that can be used to study the automotive aerodynamics and isolate relevant flow phenomena. A critical slant angle of 30° was found to lead to a dramatical change in the flow pattern.

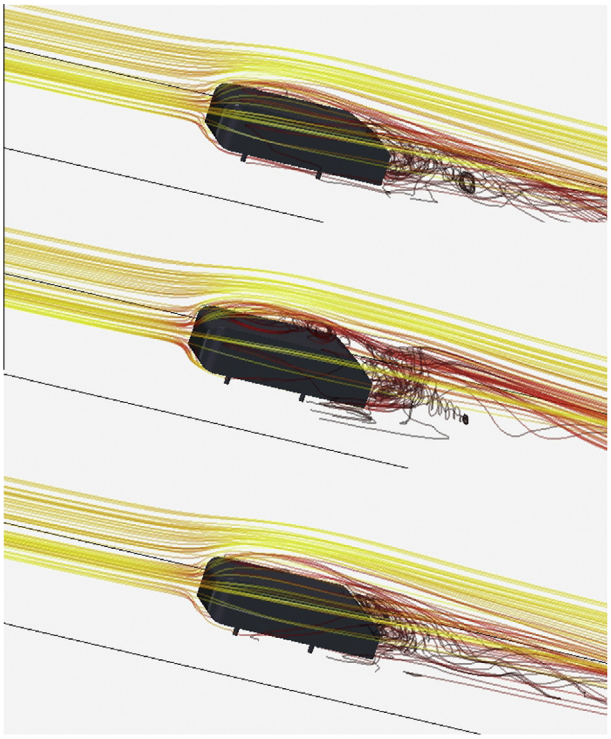

Fig. 20. Streamlines behind the vehicle (rendered by the zero-isovalues of the distance function) at different slant angle (a) 25° (b) 35° and (c) with the small cylinder.

The flexibility of the proposed monolithic approach resides in the possibility to vary easily the rear slant angle of the immersed vehicle from 25° to 35°. It also allows to add/remove small appendices to the geometry in order to study their effects and to reduce eventually the turbulent behavior behind the immersed body. Fig. 19 presents a plane cut of the adapted meshes showing different slant angles.

The simulations for the two different slant angles are given in Fig. 20. The 35° angle case shows more complex flow behavior. For this reason, we have added an appendix, a small cylinder on the top of the vehicle to study and reduce the turbulent behavior behind the immersed body.

6. Conclusion

In this paper a stabilized three-field velocity–pressure-stress finite element model, designed for the computation of rigid bodies in an incompressible Navier–Stokes flow, has been described. The method is based on treating a single set of equations for the whole domain. The presence of the solid, rendered by a distance function, is taken into account as an extra stress tensor in the Navier–Stokes equations. The considered formulation allows equal-order interpolation for the three-fields. An *a posteriori* edge based spatial error estimator, relying on the length distribution tensor approach, is developed. It allows the creation of extremely stretched elements along the interface, which is an important requirement for problems with high density ratios, for large structural deformations and for free movements of the structure within a flow domain. 2D and 3D numerical experiments were presented and results were compared against reference or other approaches. The capability of the method to simulate the fluid-rigid body interaction with complex geometries was demonstrated. The results presented here show that this method can be used in a wide range of application for multi body fluid–solid problems. Further research will focus on tackling deformable solid interaction.

Acknowledgment

The authors gratefully acknowledge support from the Carnot M.I.N.E.S.

References

[1] Mok DP, Wall WA. Partitioned analysis schemes for the transient interactions in incompressible flows and non linear flexible structures. In: Trends in computational structural mechanics. Barcelona: CIMNE; 2001.
[2] Le Tallec P, Mouro J. Fluid structure interaction with large structural displacements. Comput Methods Appl Mech Eng 2001;190(24–25):3039–67.
[3] Wall WA, Mok DP, Ramm E. Partitioned analysis approach for the transient coupled response of viscous fluids and flexible structures. In: ECCM'99, European conference on computational mechanics, 1999.
[4] Fernández MA, Moubachir M. A Newton method using exact Jacobians for solving fluid–structure coupling. Comput Struct 2005;83(2-3):127–42.
[5] Gerbeau J-F, Vidrascu M. A quasi-Newton algorithm based on a reduced model for fluid structure interaction problems in blood flow. Math Model Numer Anal 2003;37:631–47.
[6] Gerbeau J-F, Vidrascu M, Frey P. Fluid structure interaction in blood flows on geometries coming from medical imaging. Comput Struct 2005;83(2–3):155–65.
[7] Caussin P, Gerbeau J-F, Nobile F. Added-mass effect in the design of partitioned algorithms for fluid–structure problems. Comput Methods Appl Mech Eng 2005;194:4506–27.
[8] Rabczuk T, Gracie R, Song J-H, Belytschko T. Immersed particle method for fluid–structure interaction. Int J Numer Methods Fluids 2010;81:48–71.
[9] Bathe KJ, Zhang H, Wang MH. Finite element analysis of incompressible and compressible fluid flows with free surfaces and structural interactions. Comput Struct 1995;56:193–213.
[10] Bathe KJ, Zhang H. A flow-condition-based interpolation finite element procedure for incompressible fluid flows. Comput Struct 2002;80:1267–77.
[11] Michler C, van Brummelen E, Hulshoff S, de Borst R. A monolithic approach to fluid–structure interaction. Comput Fluids 2004;33:839–48.
[12] Peskin CS. The immersed boundary method. Acta Numer 2002;11:479–517.
[13] Glowinski R, Pan TW, Helsa T-I, Joseph DD. A distributed Lagrange multiplier/fictious domain methods for particulate flows. Int J Multiph Flow 1999;25(5):755–94.
[14] Codina R, Houzeaux J, Coppola-Owen H, Baiges J. The Fixed-Mesh ALE approach for the numerical approximation of flows in moving domains. J Comput Phys 2009;228:1591–611.
[15] Baiges J, Codina R. The Fixed-Mesh ALE approach applied to solid mechanics and fluid–structure interaction problems. Int J Numer Methods Eng 2010;81:1529–57.
[16] Baiges J, Codina R, Coppola-Owen H. The Fixed-Mesh ALE approach for the numerical simulation of floating solids. Int J Numer Methods Fluids 2011;67:1004–23.
[17] Kohno H, Bathe KJ. A flow-condition-based interpolation finite element procedure for triangular grids. Int J Numer Methods Fluids 2006;51:673–99.
[18] Feghali S, Hachem E, Coupez T. Monolithic stabilized finite element method for rigid body motions in the incompressible Navier–Stokes flow. Eur J Comput Mech 2010;19(5–7):547–73.
[19] Hachem E, Feghali S, Codina R, Coupez T. Immersed stress method for fluid structure interaction using anisotropic mesh adaptation. Int J Numer Methods Eng. Submitted for publication.
[20] Coupez T. Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing. J Comput Phys 2011;230:2391–405.
[21] Ville L, Silva L, Coupez T. Convected level set method for the numerical simulation of fluid buckling. Int J Numer Methods Fluids 2011;66:324–44.
[22] Bathe KJ, Zhang H. Finite element developments for general fluid flows with structural interactions. Int J Numer Methods Eng 2004;60:213–32.
[23] Bathe KJ, Zhang H. A mesh adaptivity procedure for CFD and fluid–structure interactions. Comput Struct 2009;87:604–17.
[24] Wang K, Rallu A, Gerbeau J-F, Farhat C. Algorithms for interface treatment and load computation in embedded boundary methods for fluid and fluid–structure interaction problems. Int J Numer Methods Fluids 2011;67:1175–206.
[25] 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. Int J Numer Methods Eng 2009;78(8):980–1008.
[26] Codina R, Soto O. A numerical model to track two-fluid interfaces based on a stabilized finite element method and the level set technique. Int J Numer Methods Fluids 2002;40:293–301.
[27] Alauzet F, Frey P, George P-L, Mohammadi B. 3D transient fixed point mesh adaptation for time-dependent problems: application to CFD simulations. J Comput Phys 2007;222:592–623.
[28] van der Pijl S, Segal A, Vuik C, Wesseling P. A mass-conserving level-set method for modelling of multi-phase flows. Int J Numer Methods Fluids 2005;47:339–61.
[29] Hughes TRJ. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origin of stabilized methods. Comput Methods Appl Mech Eng 1995;127:387–401.
[30] Hachem E, Rivaux B, Kloczko T, Digonnet H, Coupez T. Stabilized finite element method for incompressible flows with high Reynolds number. J Comput Phys 2010;229(23):8643–65.
[31] Codina R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. Comput Methods Appl Mech Eng 2002;191:4295–321.
[32] Badia S, Codina R. Stabilized continuous and discontinuous Galerkin techniques for Darcy flow. Comput Methods Appl Mech Eng 2010;199:1654–67.
[33] Codina R. Finite element approximation of the three field formulation of the Stokes problem using arbitrary interpolations. SIAM J Numer Anal 2009;47:699–718.
[34] Tezduyar TE, Osawa Y. Finite element stabilization parameters computed from element matrices and vectors. Comput Methods Appl Mech Eng 2000;190(3-4):411–30.
[35] Digonnet H, Coupez T. Object-oriented programming for fast and easy development of parallel applications in forming processes simulation. In: Bathe KJ, editor. Second MIT conference on computational fluid and solid mechanics, 2003; pp. 1922–1924.
[36] Mesri Y, Digonnet H, Coupez T. Advanced parallel computing in material forming with CIMLIB. Eur J Comput Mech 2009;18:669–94.
[37] Glowinski R, Pan T-W, Helsa TI, Joseph DD, Periaux J. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow. J Comput Phys 2001;169:363–426.
[38] Robinson A, Schroeder C, Fedkiw R. A symmetric positive definite formulation for monolithic fluid–structure interaction. J Comput Phys 2010;230(4):1547–66.
[39] Pagnutti D, Olivier-Gooch C. Two-dimensional Delaunay-based anisotropic mesh adaptation Doug Pagnutti. Eng Comput 2010;4:407–18.