INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2013; **94**:805–825
Published online 2 April 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.4481

Immersed stress method for fluid–structure interaction using anisotropic mesh adaptation

E. Hachem[1,*,†], S. Feghali[1], R. Codina[2] and T. Coupez[1]

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

SUMMARY

This paper presents advancements toward a monolithic solution procedure and anisotropic mesh adaptation for the numerical solution of fluid–structure interaction with complex geometry. First, a new stabilized three-field stress, velocity, and pressure finite element formulation is presented for modeling the interaction between the fluid (laminar or turbulent) and the rigid body. The presence of the structure will be taken into account by means of an extra stress in the Navier–Stokes equations. The system is solved using a finite element variational multiscale method. We combine this method with anisotropic mesh adaptation to ensure an accurate capturing of the discontinuities at the fluid–solid interface. We assess the behavior and accuracy of the proposed formulation in the simulation of 2D and 3D time-dependent numerical examples such as the flow past a circular cylinder and turbulent flows behind an immersed helicopter in a forward flight.

Received 24 April 2012; Revised 14 November 2012; Accepted 14 January 2013

KEY WORDS: monolothic approach; fluid–structure interactions; rigid body; stabilized finite elements; incompressible flows; anisotropic meshing

1. INTRODUCTION

Numerical simulations of fluid–structure interaction (FSI) with complex geometries pose many challenges for computational simulations. In spite of the available computer performance and the actual maturity of computational fluid dynamics and computational structural dynamics, several key issues in the domain of computational FSI still prevent simulations from really helping in solving some 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 ALE formulation [1–3]. 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 does not depend on the fluid motion but only ensures the continuity between the fluid and solid domains and a good mesh quality (see [4–6]). A higher popularity has been gained lately by partitioned approaches that allow the use of specific solver for each domain. The difficulty remains in transferring the information between the codes. The coupling between the two phases can be enforced usually using two schemes: weakly or strongly coupled version. 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. The predominant approach consists in solving the problem iteratively, using fixed point schemes [7–9] or Newton–Krylov methods

*Correspondence to: E. Hachem, Center for Material Forming (CEMEF), MINES-ParisTech, UMR CNRS 7635, Sophia-Antipolis, France.

†E-mail: Elie.Hachem@mines-paristech.fr

[10–12]. Actually, the fixed point methods with dynamic relaxation seem to be the most interesting variant [13]. This approach allows the use of fluid and solid solvers for each of the two phases. Although this has proven to be an accurate and efficient approach, it presents an inherent instability depending on the ratio of the densities and the complex geometry of the domain [14]. It was shown in [15] that this instability is more related to the fact that the fluid is incompressible. Moreover, the loosely coupled partitioned procedure can be stable and accurate for nonlinear FSI, as long as the fluid is modeled as a compressible medium. The coupling across nonmatching fluid and structural mesh interfaces is achieved using a conservative method described in [16].

Alternatively, monolithic approaches consist of considering the whole domain as a single one, meshed by a single grid, and solved with a Eulerian framework. Continuity at the interface is then obtained naturally, and there is no need to enforce it. The monolithic approaches impose the use of an appropriate constitutive equation describing both the fluid and solid domains. Interface tracking between the two different domains can be completed by the level set method. The embedded boundary method [17] easily handles meshing of the fluid domain and large structural motions and deformations. However, it presents some complication, especially when transmitting fluid–structure conditions. This issue can be overcome once the fluid pressure and velocity conditions on the interface are treated simultaneously and the flow-induced loads on this interface are well computed [18]. Other fixed mesh methods such as the immersed boundary methods [19] or the fictitious domain method [19, 20] treat the coupling between the domains by applying constraints across the rigid body using an augmented lagrange multiplier, possibly solved by an iterative Uzawa algorithm. These constraints may lead to uncoupled physics in the different subdomains of the problem (e.g., in the fluid and the solid), yielding inconsistencies when the subdomains evolve in time. This problem may be solved using the so called fixed-mesh ALE formulation introduced in [21] (see also [22, 23] for applications to FSI problems and rigid bodies floating in fluids).

In this work, we retain the advantages of the monolithic approaches and we propose a new three-field stabilized FEM combined with a fast anisotropic mesh adaptation algorithm. It can be seen as an extension of the immersed volume method (IVM) developed in [24–27] to deal with real FSIs problems. The motivation behind pursuing such general approach comes from the desire to easily deal with a large diversity of shapes, dimensions, and physical properties of structures. By computing the signed distance function (level set) of a given geometry to each node of the mesh, we can easily identify the solid part from the surrounding fluid as the zero level of this function. Consequently, different parts (i.e., helicopter) are immersed in a larger domain (i.e., wind channel) of different material properties.

In general, the latter intersects the element arbitrarily, and consequently, the accuracy of the finite element approach can be compromised. To circumvent this issue, the level set solution is coupled to advanced methods in anisotropic mesh adaptation at the interface. The algorithm allows the creation of extremely stretched elements along the interface, which is an important requirement for FSI problems and turbulence modeling having internal/boundary layers.

The novelty of this paper resides more on developing a three-field stabilized FEM for FSI. This allows modeling the interaction between the fluid (laminar or turbulent) and the structure in question. First, we modify the Navier–Stokes equations by adding an extra stress that takes into account the presence of the structure. For illustration, the rigid immersed body is treated using the Navier–Stokes solver under the constraints of imposing its nullity of the deformation rate by means of a Lagrange multiplier [24]. Second, because the added stress tensor will affect the fluid solver, we extend the variational multiscale (VMS) FEM to deal with both the fluid and the solid using an appropriate constitutive model. Thus, the decomposition for both the velocity and pressure fields into coarse/resolved scales and fine/unresolved scales needed to deal with convection-dominated problems and pressure instabilities will be extended 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. Finally, the implementation aspects and the corresponding algorithms to solve this new system will be presented and detailed.

This paper is organized as follows. First, the IVM is introduced in Section 2. Section 3 is dedicated to present the governing equations. The stabilizing schemes from a VMS point of view are described

DOI: 10.1002/nme

in Section 4. Then, in Section 5, the numerical scheme is presented. Section 6 presents some benchmark problems and comparisons with the literature and the numerical simulation of unsteady flow past an immersed helicopter in forward flight. Finally, Section 7 is dedicated to conclusions and future work.

2. IMMERSED VOLUME METHOD

The IVM has shown to be an effective geometric representation tool [27]. It is 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 using the gradient of the distance function, and finally we mix and attribute the physical properties of each domain using appropriate laws. In this section, we revisit briefly these steps.

2.1. Level set approach

A signed distance function of an interface Γ_{im} is used to localize the interface of the immersed body Ω_{im} 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 distance from Γ_{im}. In turn, the interface Γ_{im} is then 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 $\alpha_{\text{im}} \leqslant 0$ outside this domain. Further details about the algorithm used to compute the distance are available in [28]. It is also possible to use functions smoother than $d(\mathbf{x}, \Gamma_{\text{im}})$ away from Γ_{im} (see, e.g., [29]).

2.2. 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 then 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 the following:

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

To achieve a better continuity at the interface [30], the Heaviside function can be smoothed using

$$H_\varepsilon(\alpha) = \begin{cases} 1 & \text{if } \alpha > \varepsilon \\ \frac{1}{2}\left(1 + \frac{\alpha}{\varepsilon} + \frac{1}{\pi}\sin\left(\frac{\pi\alpha}{\varepsilon}\right)\right) & \text{if } |\alpha| \leqslant \varepsilon \\ 0 & \text{if } \alpha < -\varepsilon \end{cases} \tag{3}$$

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:

$$h_{\text{im}} = \max_{j,l \in K} \nabla\alpha \cdot \mathbf{x}^{jl} \tag{4}$$

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

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 $P0$.

DOI: 10.1002/nme

2.3. Anisotropic mesh adaptivity

The anisotropic mesh adaptation is aimed at improving the quality of the mesh. This algorithm relies on an appropriate metric map construction that allows the mesh size to be imposed in the direction of the distance function gradient. The mesh is refined locally, and elements are stretched, which enables to sharply define the interface and to save a great number of elements compared with classical isotropic refinement. The details of this technique can be found in [31, 32]; hence, a brief description is presented here. First, a modified distance computation is achieved via a symmetric positive defined tensor $\mathbb{M}$, which acts as a metric with the inner product and associated norm

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathrm{M}} = {}^{T}\mathbf{x} \cdot \mathbb{M} \cdot \mathbf{y}\,, \quad ||\mathbf{x}||_{\mathrm{M}} = \sqrt{{}^{T}\mathbf{x} \cdot \mathbb{M} \cdot \mathbf{x}} \tag{5}$$

which can be defined at each point of the computational domain. 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 that corresponds to the gradient of the level set function: $\mathbf{x} = \nabla\alpha/||\nabla\alpha||$. 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:

$$h = \begin{cases} h_{\mathrm{d}} & \text{if } |\alpha(\mathbf{x})| > e/2 \\ \frac{2h_{\mathrm{d}}(m-1)}{m\,e}|\alpha(\mathbf{x})| + \frac{h_{\mathrm{d}}}{m} & \text{if } |\alpha(\mathbf{x})| \leqslant e/2 \end{cases} \tag{6}$$

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 equaling 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 previously lead to the following metric:

$$\mathbb{M} = C(\mathbf{x} \otimes \mathbf{x}) + \frac{1}{h_{\mathrm{d}}^2}\,\mathbb{I} \quad \text{with} \quad C = \begin{cases} 0 & \text{if } |\alpha(\mathbf{x})| \geqslant e/2 \\ \frac{1}{h^2} - \frac{1}{h_{\mathrm{d}}^2} & \text{if } |\alpha(\mathbf{x})| < e/2 \end{cases} \tag{7}$$

where $\mathbb{I}$ is the identity tensor. The mesh resolution can be forced to concentrate in particular areas of interest. The metric returns to isotropy far from the interface (with a mesh size equal to h_{d} for all directions) and to anisotropy near the interface (with a mesh size equal to h_{im} in the direction $\mathbf{x}$ and equal to h_{d} in the others). Once the metric is computed, we use a mesh technique based on the local modification and the conformity control through the theorem for minimal volume preserving [32]. This method can be assisted by a posteriori anisotropic error estimator, which is 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 may be obtained.

The proposed mesh generation algorithm provides a high precision in the calculus and in the geometry description, in spite of an important decrease of the total number of DOFs. 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 the domain keeps the same background size.

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 3D 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. We shall start presenting the formulation in the case of the stationary Stokes problem, and then, we shall move to the general transient and nonlinear problem.

DOI: 10.1002/nme

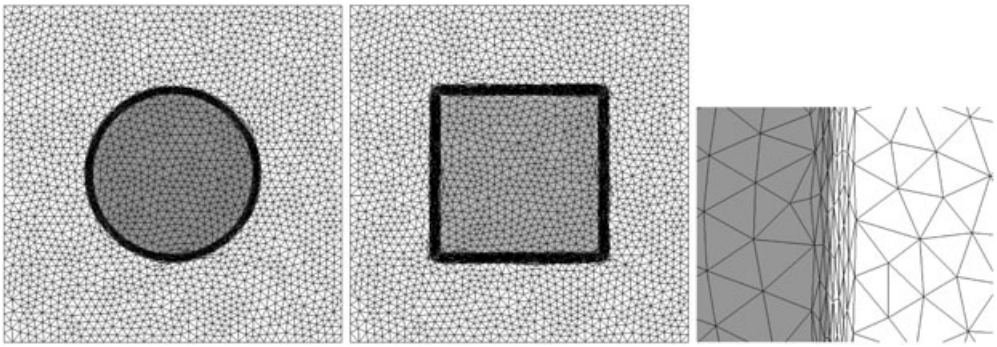


Figure 1. Anisotropic refined fluid–solid interface of (left) an immersed disk, (center) an immersed square, and (right) a zoom on the refined interface.

3.1. Rigid body restriction in a stationary Stokes flow

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. The fluid domain, the solid domain, and the interface will be Ω_f, Ω_s, and Γ_im, respectively. They verify the following:

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

Consider the Stokes problem, first posed on the whole domain Ω: find a velocity $\mathbf{v} : \Omega \longrightarrow \mathbb{R}^n$ and a pressure $p : \Omega \longrightarrow \mathbb{R}$ such that

$$\begin{aligned} -2\eta_\mathrm{f} \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{v}) + \nabla p &= \mathbf{f} \\ \nabla \cdot \mathbf{v} &= 0 \end{aligned}$$

where η_f is the fluid viscosity, $\boldsymbol{\varepsilon}(\mathbf{v})$ the strain rate tensor, and $\mathbf{f}$ the vector of body forces. We may assume for simplicity the homogeneous Dirichlet boundary conditions.

The solution is the optimal point of the functional

$$L_1(\mathbf{v}, p) = \eta_\mathrm{f}(\boldsymbol{\varepsilon}(\mathbf{v}), \boldsymbol{\varepsilon}(\mathbf{v})) - (p, \nabla \cdot \mathbf{v}) - \langle \mathbf{f}, \mathbf{v} \rangle$$

Here and in the following paragraphs, $(\cdot, \cdot)$ stands for the standard L^2 product of two functions and $\langle \cdot, \cdot \rangle$ for the integral of the product of two functions in the general case, including appropriate duality pairings if necessary.

Consider now the presence of the rigid body Ω_s in Ω. *It may be characterized by the fact that* $\boldsymbol{\varepsilon}(\mathbf{v}) = \mathbf{0}$ in Ω_s. Imposing this restriction through an appropriate Lagrange multiplier $\boldsymbol{\tau}$, the problem to be solved consists of finding the optimal point of the functional

$$L_2(\mathbf{v}, p, \boldsymbol{\tau}) = \eta_\mathrm{f}(\boldsymbol{\varepsilon}(\mathbf{v}), \boldsymbol{\varepsilon}(\mathbf{v})) - (p, \nabla \cdot \mathbf{v}) + (\boldsymbol{\tau}, \boldsymbol{\varepsilon}(\mathbf{v}))_\mathrm{s} - \langle \mathbf{f}, \mathbf{v} \rangle$$

where $(\cdot, \cdot)_\mathrm{s}$ denotes the inner product in Ω_s. If $(\mathbf{w}, q, \boldsymbol{\xi})$ are the test functions for $(\mathbf{v}, p, \boldsymbol{\tau})$, the weak form of the problem consists of finding $(\mathbf{v}, p, \boldsymbol{\tau})$ in the adequate functional spaces such that

$$2\eta_\mathrm{f}(\boldsymbol{\varepsilon}(\mathbf{v}), \boldsymbol{\varepsilon}(\mathbf{w})) + (\boldsymbol{\tau}, \boldsymbol{\varepsilon}(\mathbf{w}))_\mathrm{s} - (p, \nabla \cdot \mathbf{w}) = \langle \mathbf{f}, \mathbf{w} \rangle \tag{8}$$

$$(q, \nabla \cdot \mathbf{v}) = 0 \tag{9}$$

$$(\boldsymbol{\xi}, \boldsymbol{\varepsilon}(\mathbf{v}))_\mathrm{s} = 0 \tag{10}$$

which must hold for all appropriate $(\mathbf{w}, q, \boldsymbol{\xi})$. The strong form of the problem reads

$$-2\eta_\mathrm{f} \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{v}) - \nabla \cdot \chi_\mathrm{s} \boldsymbol{\tau} + \nabla p = \mathbf{f} \tag{11}$$

$$\nabla \cdot \mathbf{v} = 0 \tag{12}$$

$$\chi_\mathrm{s} \boldsymbol{\varepsilon}(\mathbf{v}) = \mathbf{0} \tag{13}$$

DOI: 10.1002/nme

where χ_s is the characteristic function of Ω_s.

The critical point is the setting of the Lagrange multiplier $\boldsymbol{\tau}$. The equation $(\boldsymbol{\xi}, \boldsymbol{\varepsilon}(\mathbf{v}))_s = 0$ for all $\boldsymbol{\xi}$ has to have enough *restrictions* to yield $\boldsymbol{\varepsilon}(\mathbf{v}) = 0$ on Ω_s, but not too many. Loosely speaking, there are only three functions to be set (the components of $\mathbf{v}$), and therefore, there cannot be more than three scalar multipliers to be determined, whereas $\boldsymbol{\tau}$ has, in principle, six components (assuming it is symmetric).

A possible way to choose $\boldsymbol{\tau}$ that leads to a stable scheme is to take it as $\boldsymbol{\tau} = 2\eta_s \boldsymbol{\varepsilon}(\mathbf{g})$, where $\mathbf{g}$ is *a vector field to be determined* and η_s is a viscosity acting as scaling coefficient. If $\mathbf{h}$ is the test function for $\mathbf{g}$, the final weak form of the problem would be

$$2\eta_f(\boldsymbol{\varepsilon}(\mathbf{v}), \boldsymbol{\varepsilon}(\mathbf{w})) + 2\eta_s(\boldsymbol{\varepsilon}(\mathbf{g}), \boldsymbol{\varepsilon}(\mathbf{w}))_s - (p, \nabla \cdot \mathbf{w}) = \langle \mathbf{f}, \mathbf{w} \rangle \tag{14}$$

$$(q, \nabla \cdot \mathbf{v}) = 0 \tag{15}$$

$$-2\eta_s(\boldsymbol{\varepsilon}(\mathbf{h}), \boldsymbol{\varepsilon}(\mathbf{v}))_s = 0 \tag{16}$$

which must hold for all appropriate $(\mathbf{w}, q, \mathbf{h})$. It is not our intention to pursue here the analysis of the well posedness of (14)–(16), which will be presented elsewhere.

Remarks

- Note that the resulting problem is completely different to what would be obtained using a penalty method, in which $\boldsymbol{\tau} = 2\eta_s \chi_s \boldsymbol{\varepsilon}(\mathbf{v})$ with $\eta_s \to \infty$.
- Equation (15) has to be imposed only in the fluid, because in the solid, it is implied by (16). At the discrete level, this is not necessary if one can guarantee that (15) is compatible with (16).
- The vector field $\mathbf{g}$ is undetermined by infinitesimal rigid body motions, in the sense that for any constant vectors $\mathbf{a}$ and $\mathbf{b}$, $\boldsymbol{\varepsilon}(\mathbf{g} + \mathbf{a} + \mathbf{b} \times \mathbf{x}) = \boldsymbol{\varepsilon}(\mathbf{g})$, $\mathbf{x}$ being the position vector.

Equation (16) can be relaxed iteratively using an augmented Lagrangian formulation coupled with an Uzawa scheme (iterative penalization). Let $r > 1$ be a given penalty parameter. A possible iterative scheme to solve (14)–(16) would be

$$2\frac{\eta_s}{r}(\boldsymbol{\varepsilon}(\mathbf{h}), \boldsymbol{\varepsilon}(\mathbf{g}^k) - \boldsymbol{\varepsilon}(\mathbf{g}^{k-1}))_s - 2\eta_s(\boldsymbol{\varepsilon}(\mathbf{h}), \boldsymbol{\varepsilon}(\mathbf{v}^{k-1}))_s = 0 \tag{17}$$

$$2\eta_f(\boldsymbol{\varepsilon}(\mathbf{v}^k), \boldsymbol{\varepsilon}(\mathbf{w})) + 2\eta_s(\boldsymbol{\varepsilon}(\mathbf{g}^k), \boldsymbol{\varepsilon}(\mathbf{w}))_s - (p^k, \nabla \cdot \mathbf{w}) = \langle \mathbf{f}, \mathbf{w} \rangle \tag{18}$$

$$(q, \nabla \cdot \mathbf{v}^k) = 0 \tag{19}$$

where the superscript denotes the iteration counter.

Let us *call* $\boldsymbol{\tau} = 2\eta_s \boldsymbol{\varepsilon}(\mathbf{g})$, $\boldsymbol{\xi} = 2\eta_s \boldsymbol{\varepsilon}(\mathbf{h})$. This definition allows us to write (17)–(19) as

$$\frac{1}{2\eta_s r}(\boldsymbol{\xi}, \boldsymbol{\tau}^k - \boldsymbol{\tau}^{k-1})_s - (\boldsymbol{\xi}, \boldsymbol{\varepsilon}(\mathbf{v}^{k-1}))_s = 0 \tag{20}$$

$$2\eta_f(\boldsymbol{\varepsilon}(\mathbf{v}^k), \boldsymbol{\varepsilon}(\mathbf{w})) + (\boldsymbol{\tau}^k, \boldsymbol{\varepsilon}(\mathbf{w}))_s - (p^k, \nabla \cdot \mathbf{w}) = \langle \mathbf{f}, \mathbf{w} \rangle \tag{21}$$

$$(q, \nabla \cdot \mathbf{v}^k) = 0 \tag{22}$$

If P_τ is the projection onto the space of $\boldsymbol{\tau}$s, (20) can be written as

$$\boldsymbol{\tau}^k = \boldsymbol{\tau}^{k-1} + 2\eta_s r \, P_\tau(\boldsymbol{\varepsilon}(\mathbf{v}^{k-1})) \tag{23}$$

In the iterative procedure described, the velocity could be treated implicitly in the evaluation of $\boldsymbol{\tau}$, that is to say, $\mathbf{v}^{k-1}$ in (17), (20), and (23) could be replaced by $\mathbf{v}^k$. In this case, there would be an additional contribution to $\boldsymbol{\varepsilon}(\mathbf{v}^k)$ in (18) and (21).

Note that at the continuous level, (23) holds with P_τ the identity, because all tensors in (20) have L^2-components. However, for numerical purposes, it could be interesting *to interpolate* $\boldsymbol{\tau}$ *as an independent variable*. The problem to be solved would be exactly the same as the previous one, which converges to the original problem (8)–(10). However, it is expected that, *because of the iterative way to compute* $\boldsymbol{\tau}$, its space will automatically have the appropriate dimension. Note also that

DOI: 10.1002/nme

infinitesimal rigid body motions will be filtered because $\mathbf{v}$ is uniquely determined by the boundary conditions. In any case, (23) determines the space of $\boldsymbol{\tau}$ without explicitly building it, as a natural consequence of the use of an augmented Lagrangian approach.

3.2. The Navier–Stokes equations with a rigid body

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

$$\rho_{\mathrm{f}}\,(\partial_t \mathbf{v} + \mathbf{v}\cdot\nabla\mathbf{v}) - \nabla\cdot\boldsymbol{\sigma} = \mathbf{f} \quad \text{in } \Omega_{\mathrm{f}}, t > 0 \tag{24}$$

$$\nabla\cdot\mathbf{v} = 0 \quad \text{in } \Omega_{\mathrm{f}}, t > 0 \tag{25}$$

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

$$\boldsymbol{\sigma} = 2\eta_{\mathrm{f}}\,\boldsymbol{\varepsilon}(\mathbf{v}) - p\,\mathbf{I}_d \tag{26}$$

where $\mathbf{I}_d$ is the d-dimensional identity tensor. Equations (24) and (25) are subject to the boundary and initial conditions

$$\mathbf{v} = \mathbf{v}_{\Gamma,\mathrm{f}} \quad \text{on } \partial\Omega_{\mathrm{f}} \setminus \Gamma_{\mathrm{im}}, t > 0 \tag{27}$$

$$\mathbf{v} = \mathbf{v}_{\mathrm{im}} \quad \text{on } \Gamma_{\mathrm{im}}, t > 0 \tag{28}$$

$$\boldsymbol{\sigma}\cdot\mathbf{n} = \mathbf{t}_{\mathrm{im}} \quad \text{on } \Gamma_{\mathrm{im}}, t > 0 \tag{29}$$

$$\mathbf{v}(\mathbf{x}, 0) = \mathbf{v}_0(\mathbf{x}) \quad \text{in } \Omega_{\mathrm{f}} \tag{30}$$

where $\mathbf{v}_{\Gamma,\mathrm{f}}$ is a given velocity boundary condition, $\mathbf{v}_{\mathrm{im}}$ 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, $\mathbf{t}_{\mathrm{im}}$ 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. For incompressible flows, the divergence-free constraint (25) gives rise to the pressure in the fluid.

In the present formulation, we treat the rigid body as a continuous domain subjected to an additional rigidity constraint. We may write the equations of motion as the Navier–Stokes equations with this constraint as

$$\rho_{\mathrm{s}}\,(\partial_t \mathbf{v} + \mathbf{v}\cdot\nabla\mathbf{v}) - \nabla\cdot\boldsymbol{\sigma} = \mathbf{f} \quad \text{in } \Omega_{\mathrm{s}}, t > 0 \tag{31}$$

$$\nabla\cdot\mathbf{v} = 0 \quad \text{in } \Omega_{\mathrm{s}}, t > 0 \tag{32}$$

$$\boldsymbol{\varepsilon}(\mathbf{v}) = \mathbf{0} \quad \text{in } \Omega_{\mathrm{s}}, t > 0 \tag{33}$$

where ρ_{s} is 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 (33). Note also that the rigidity constraint (33) ensures that the velocity field is divergence free. Hence, (32) is a redundant equation. Nevertheless, we choose to keep it to account for the pressure term. As noted earlier, (32) gives rise to a pressure fluid. Similarly, as explained in Section 3.1 for the Stokes problem, (33) gives rise to a stress field $\boldsymbol{\tau}$. The stress tensor is then given by the following:

$$\boldsymbol{\sigma} = \boldsymbol{\tau}_{\mathrm{s}} - p\,\mathbf{I}_d \tag{34}$$

Equations (31)–(33) need to be supplied with the boundary and initial conditions

$$\mathbf{v} = \mathbf{v}_{\Gamma,\mathrm{s}} \quad \text{on } \partial\Omega_{\mathrm{s}} \setminus \Gamma_{\mathrm{im}}, t > 0 \tag{35}$$

$$\mathbf{v} = \mathbf{v}_{\mathrm{im}} \quad \text{on } \Gamma_{\mathrm{im}}, t > 0 \tag{36}$$

$$\boldsymbol{\sigma}\cdot\mathbf{n} = -\mathbf{t}_{\mathrm{im}} \quad \text{on } \Gamma_{\mathrm{im}}, t > 0 \tag{37}$$

$$\mathbf{v}(\mathbf{x}, 0) = \mathbf{v}_0(\mathbf{x}) \quad \text{in } \Omega_{\mathrm{s}} \tag{38}$$

DOI: 10.1002/nme

where $\mathbf{v}_{\Gamma,\mathrm{f}}$ is a given velocity boundary condition that needs to be compatible with a rigid body motion if $\partial\Omega_\mathrm{s} \setminus \Gamma_\mathrm{im} \neq \emptyset$, and the initial condition $\mathbf{v_0}(\mathbf{x})$ must be also compatible with a rigid body motion.

Making use of the notation introduced in Section 2, we may write problems (24)–(30) and (31)–(38) in a unified way in the whole computational domain Ω as

$$\begin{aligned} \rho\,(\partial_t \mathbf{v} + \mathbf{v}\cdot\nabla\mathbf{v}) - \nabla\cdot(2\eta\,\boldsymbol{\varepsilon}(\mathbf{v}) + \boldsymbol{\tau} - p\,\mathbf{I}_d) &= \mathbf{f} && \text{in } \Omega, t>0 \\ \nabla\cdot\mathbf{v} &= 0 && \text{in } \Omega, t>0 \\ \boldsymbol{\varepsilon}_\mathrm{s}(\mathbf{v}) &= \mathbf{0} && \text{in } \Omega, t>0 \\ \mathbf{v} &= \mathbf{v}_\Gamma && \text{on } \partial\Omega, t>0 \\ \mathbf{v}(\mathbf{x},0) &= \mathbf{v_0}(\mathbf{x}) && \text{in } \Omega \end{aligned} \tag{39}$$

where $\mathbf{v}_\Gamma = \mathbf{v}_{\Gamma,\mathrm{s}}$ on $\partial\Omega_\mathrm{s}\cap\partial\Omega$ and $\mathbf{v}_\Gamma = \mathbf{v}_{\Gamma,\mathrm{f}}$ on $\partial\Omega_\mathrm{f}\cap\partial\Omega$, $\boldsymbol{\varepsilon}_\mathrm{s}(\mathbf{v}) = H(\alpha)\boldsymbol{\varepsilon}(\mathbf{v})$, $\eta = (1-H(\alpha))\eta_\mathrm{f}$, $\rho = \rho_\mathrm{s}H(\alpha) + \rho_\mathrm{f}(1-H(\alpha))$, and $\boldsymbol{\tau} = H(\alpha)\boldsymbol{\tau}_\mathrm{s}$. The boundary conditions (28), (29), (36), and (37) 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 that are square integrable in time with values in $H^1(\Omega)^n$ 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 $\mathcal{T}_0 = L^2(\Omega)^{n\times n}$, respectively (in fact, a subspace of $L^2(\Omega)^{n\times n}$ would be enough). The corresponding test functions will be denoted $\mathbf{w} \in V_0 = H_0^1(\Omega)^n$, $q \in P_0$, and $\boldsymbol{\xi} \in \mathcal{T}_0$. Multiplying by the test functions and integrating by parts, we can state the associated standard weak form of the system (39) as follows: Find $\mathbf{v} \in V$, $p \in P$, and $\boldsymbol{\tau} \in \mathcal{T}$ such that

$$\begin{aligned} \rho(\partial_t\mathbf{v},\mathbf{w}) + \rho(\mathbf{v}\cdot\nabla\mathbf{v},\mathbf{w}) - (p,\nabla\cdot\mathbf{w}) + (2\eta\boldsymbol{\varepsilon}(\mathbf{v}),\boldsymbol{\varepsilon}(\mathbf{w})) + (\boldsymbol{\tau},\boldsymbol{\varepsilon}_\mathrm{s}(\mathbf{w})) &= \langle\mathbf{f},\mathbf{w}\rangle \\ (q,\nabla\cdot\mathbf{v}) &= 0 \\ -(\boldsymbol{\xi},\boldsymbol{\varepsilon}_\mathrm{s}(\mathbf{v})) &= 0 \end{aligned} \tag{40}$$

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

As explained in Section 3.1 for the Stokes case, the problem is not well posed without additional conditions on $\boldsymbol{\tau}$. A possible way to choose it 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 is employed. This is what we describe next.

Suppose that we discretize in time problem (40) 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 $\delta_t\mathbf{v}$ be 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 (i.e., using $\mathbf{v}^k$ in (17), (20), and (23) instead of $\mathbf{v}^{k-1}$), the iterative scheme to be performed *within each time step* reads

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

$$\begin{aligned} &\rho(\delta_t\mathbf{v}^k,\mathbf{w}) + \rho(\mathbf{v}^k\cdot\nabla\mathbf{v}^k,\mathbf{w}) - (p^k,\nabla\cdot\mathbf{w}) \\ &\qquad + 2(\eta\boldsymbol{\varepsilon}(\mathbf{v}^k) + H(\alpha)\eta_\mathrm{s}\,r\,P_\tau(\boldsymbol{\varepsilon}(\mathbf{v}^k)),\boldsymbol{\varepsilon}(\mathbf{w})) + (\boldsymbol{\tau}^{k-1},\boldsymbol{\varepsilon}_\mathrm{s}(\mathbf{w})) = \langle\mathbf{f},\mathbf{w}\rangle \\ &(q,\nabla\cdot\mathbf{v}^k) = 0 \end{aligned}$$

(5) Update $\boldsymbol{\tau}^k = \boldsymbol{\tau}^{k-1} + 2\eta_\mathrm{s}\,r\,P_\tau(\boldsymbol{\varepsilon}(\mathbf{v}^k))$ 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.

Remarks

- If the space for $\boldsymbol{\tau}$ contains all symmetric gradients of vector fields, $P_\tau = I$ (identity) and we could solve the momentum equation in step (4) with the viscosity $\eta + H(\alpha)\eta_\mathrm{s}\,r$. However, at

DOI: 10.1002/nme

the discrete level, it can be of interest to take the space for the approximation to $\boldsymbol{\tau}$ different from symmetric gradients.

- In the previous scheme, we have considered Uzawa's iterations uncoupled from the iterations required to linearize the convective term and the time stepping itself. Obviously, it could also be possible to deal with these in a coupled way.

4. STABILIZED FEM

In this section, we describe briefly the Galerkin finite element approximation and the corresponding stabilization method for the resulting discrete system of equations (40). On the basis of 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.

As it is well known, the stability of the discrete formulation depends on appropriate compatibility restrictions on the choice of the finite element spaces. According to this, standard Galerkin mixed elements with, for example, continuous equal-order linear/linear interpolation for the three fields are not stable. Lack of stability shows as uncontrollable oscillations that pollute the solution. We propose in here a VMS method (see [33]), which allows one to equal-order continuous interpolations for the three fields, apart from preventing from oscillations due to convection-dominated flows. The basic idea is to consider that the unknowns can be split in two components, coarse and fine, corresponding to different scales or levels of resolution [26]. First, we solve the fine scales in an approximate manner, and then, we replace their effect into the large scales.

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 finite element (coarse) component, whereas the prime is used for the so-called subgrid scale (fine) component of the unknowns. According to this, we have

$$\begin{aligned}\mathbf{v} &= \mathbf{v}_h + \mathbf{v}' \in V_h \oplus V' \\ p &= p_h + p' \in P_h \oplus P' \\ \boldsymbol{\tau} &= \boldsymbol{\tau}_h + \boldsymbol{\tau}' \in \mathcal{T}_h \oplus \mathcal{T}'\end{aligned}$$

If the spaces for the test functions are split likewise, with a subscript 0 to identify them, problem (40) becomes the following: 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

$$\begin{aligned}\rho(\delta_t(\mathbf{v}_h + \mathbf{v}'), \mathbf{w}_h + \mathbf{w}') + \rho((\mathbf{v}_h + \mathbf{v}') \cdot \nabla(\mathbf{v}_h + \mathbf{v}'), \mathbf{w}_h + \mathbf{w}') - (p_h + p', \nabla\cdot(\mathbf{w}_h + \mathbf{w}')) \\ +2(\eta\boldsymbol{\varepsilon}(\mathbf{v}_h + \mathbf{v}'), \boldsymbol{\varepsilon}(\mathbf{w}_h + \mathbf{w}')) + (\boldsymbol{\tau}_h + \boldsymbol{\tau}', \boldsymbol{\varepsilon}_{\text{s}}(\mathbf{w}_h + \mathbf{w}')) = \langle \mathbf{f}, \mathbf{w}_h + \mathbf{w}' \rangle\end{aligned} \tag{41}$$

$$(q_h + q', \nabla \cdot (\mathbf{v}_h + \mathbf{v}')) = 0 \tag{42}$$

$$-(\boldsymbol{\xi}_h + \boldsymbol{\xi}', \boldsymbol{\varepsilon}_{\text{s}}(\mathbf{v}_h + \mathbf{v}')) = 0 \tag{43}$$

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 [34]), we will make use of some common approximations:

(i) The subscales are not tracked in time; therefore, quasi-static subscales are considered here. However, the subscale equation remains quasi time dependent.
(ii) The convective velocity of the nonlinear term may be approximated using only the large-scale component, so that $(\mathbf{v}_h + \mathbf{v}') \cdot \nabla(\mathbf{v}_h + \mathbf{v}') \approx \mathbf{v}_h \cdot \nabla(\mathbf{v}_h + \mathbf{v}')$. Moreover, this approximation can be performed also if the convective term is written as $\nabla \cdot (\mathbf{v}_h + \mathbf{v}') \otimes (\mathbf{v}_h + \mathbf{v}')$, which is relevant when integrating by parts the convective term.
(iii) Terms involving subscales can be integrated by parts and the subscales on the element boundaries neglected.

DOI: 10.1002/nme

The equations for the coarse scales are obtained taking the subscale test functions equal to zero. Doing this and using the previous assumptions, we have

$$\rho(\delta_t \mathbf{v}_h, \mathbf{w}_h) + \rho(\mathbf{v}_h \cdot \nabla \mathbf{v}_h, \mathbf{w}_h) - (p_h + p', \nabla \cdot \mathbf{w}_h) + 2(\eta \boldsymbol{\varepsilon}(\mathbf{v}_h), \boldsymbol{\varepsilon}(\mathbf{w}_h)) + (\boldsymbol{\tau}_h + \boldsymbol{\tau}', \boldsymbol{\varepsilon}_{\mathrm{s}}(\mathbf{w}_h)) + \sum_K (\mathbf{v}', -\rho \mathbf{v}_h \cdot \nabla \mathbf{w}_h - \nabla \cdot (2\eta \boldsymbol{\varepsilon}(\mathbf{w}_h)))_K = \langle \mathbf{f}, \mathbf{w}_h \rangle \tag{44}$$

$$(q_h, \nabla \cdot \mathbf{v}_h) - \sum_K (\mathbf{v}', \nabla q_h)_K = 0 \tag{45}$$

$$-(\boldsymbol{\varepsilon}_{\mathrm{s}}(\mathbf{v}_h), \boldsymbol{\xi}_h) + \sum_K (\mathbf{v}', \chi_{\mathrm{s}} \nabla \cdot \boldsymbol{\xi}_h)_K = 0 \tag{46}$$

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 (41)–(43) and using approximations (i)–(iii) described earlier. Introducing the finite element residuals

$$\begin{aligned} \mathcal{R}_v &= \mathbf{f} - \rho \delta_t \mathbf{v}_h - \rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h - \nabla p_h + \chi_{\mathrm{s}} \nabla \cdot \boldsymbol{\tau}_h + \nabla \cdot (2\eta \boldsymbol{\varepsilon}(\mathbf{v}_h)) \\ \mathcal{R}_p &= -\nabla \cdot \mathbf{v}_h \\ \mathcal{R}_\tau &= \boldsymbol{\varepsilon}_{\mathrm{s}}(\mathbf{v}_h) \end{aligned}$$

and using the same ideas as in [35, 36], it turns out that the subscales may be approximated within each element $K \in \mathcal{K}_h$ by

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

where Π'_v, Π'_p, and Π'_τ are the projections onto V', P', and $\mathcal{T}'$, respectively, and α_v, α_p, and α_τ are the so-called stabilization parameters. The most common choice is to take the former as the identity when applied to finite element residuals (see [33, 35]), and this is what we will do here, although it is also possible to take them as the projection orthogonal to the finite element space (see [36] and references therein). Referring to the stabilization parameters, we compute them within each element as

$$\alpha_v = \left[\left(\frac{c_1 \eta}{\rho h^2} \right)^2 + \left(\frac{c_2 \|\mathbf{v}_h\|_K}{h} \right)^2 \right]^{-1/2} \tag{47}$$

$$\alpha_p = \left[\left(\frac{\eta}{\rho} \right)^2 + \left(\frac{c_2 \|\mathbf{v}_h\|_K h}{c_1} \right)^2 \right]^{1/2} \tag{48}$$

$$\alpha_\tau = c_3 \frac{h}{L} 2\eta_{\mathrm{s}} \tag{49}$$

where h is the element size, L is a characteristic length of the computational domain, $\|\mathbf{v}\|_K$ is 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 (see [37]).

Remarks

- Very often, the time-step size of the temporal discretization is included in the expression of α_v. This improves the convergence behavior of the algorithm to deal with the nonlinearity of the problem, but has several conceptual drawbacks, as explained in [37, 38]. To make α_v more uniform over the computational domain and, as a consequence, improve the behavior of the scheme, one may take

$$\alpha_v = \left[\frac{1}{\alpha_0^2} + \left(\frac{c_1 \eta}{\rho h^2} \right)^2 + \left(\frac{c_2 \|\mathbf{v}_h\|_K}{h} \right)^2 \right]^{-1/2}$$

DOI: 10.1002/nme

where α_0 is a reference value of α_v given by (47) computed over the whole mesh (e.g., the minimum over all the elements). This value in fact should be related to the time-step size of the time discretization, Δt.

- The factor h/L in (49) improves the convergence of stresses when equal interpolation is used for all variables [35, 39]. However, it is possible to take it out (that is to say, to take $L = h$) and obtain optimal convergence for velocity and pressure [36].
- 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.

Inserting the expression for the subscales obtained in (44)–(46), we finally obtain the stabilized finite element problem we were seeking. It consists of finding $(\mathbf{v}_h, p_h, \boldsymbol{\tau}_h) \in V_h \times P_h \times \mathcal{T}_h$ such that

$$\begin{aligned}
&\rho(\delta_t \mathbf{v}_h, \mathbf{w}_h) + \rho(\mathbf{v}_h \cdot \nabla \mathbf{v}_h, \mathbf{w}_h) - (p_h, \nabla \cdot \mathbf{w}_h) + 2(\eta\boldsymbol{\varepsilon}(\mathbf{v}_h), \boldsymbol{\varepsilon}(\mathbf{w}_h)) + (\boldsymbol{\tau}_h, \boldsymbol{\varepsilon}_{\mathrm{s}}(\mathbf{w}_h)) \\
&\quad + \sum_K \alpha_v (\rho\delta_t \mathbf{v}_h + \rho\mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h - \chi_{\mathrm{s}} \nabla \cdot \boldsymbol{\tau}_h - \nabla \cdot (2\eta\boldsymbol{\varepsilon}(\mathbf{v}_h)), \rho\mathbf{v}_h \cdot \nabla \mathbf{w}_h + \nabla \cdot (2\eta\boldsymbol{\varepsilon}(\mathbf{w}_h)))_K \\
&\quad + \sum_K \alpha_p (\nabla \cdot \mathbf{v}_h, \nabla \cdot \mathbf{w}_h) + \sum_K \alpha_\tau (\boldsymbol{\varepsilon}_{\mathrm{s}}(\mathbf{v}_h), \boldsymbol{\varepsilon}_{\mathrm{s}}(\mathbf{w}_h)) \\
&\quad = \langle \mathbf{f}, \mathbf{w}_h \rangle + \sum_K \alpha_v (\mathbf{f}, \rho\mathbf{v}_h \cdot \nabla \mathbf{w}_h + 2\eta \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{w}_h))_K \qquad (50) \\
&\quad (q_h, \nabla \cdot \mathbf{v}_h) + \sum_K \alpha_v (\rho\delta_t \mathbf{v}_h + \rho\mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h - \chi_{\mathrm{s}} \nabla \cdot \boldsymbol{\tau}_h - \nabla \cdot (2\eta\boldsymbol{\varepsilon}(\mathbf{v}_h)), \nabla q_h)_K \\
&\quad = \sum_K \alpha_v (\mathbf{f}, \nabla q_h)_K \qquad (51) \\
&\quad - (\boldsymbol{\varepsilon}_{\mathrm{s}}(\mathbf{v}_h), \boldsymbol{\xi}_h) + \sum_K \alpha_v (\rho\delta_t \mathbf{v}_h + \rho\mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h - \chi_{\mathrm{s}} \nabla \cdot \boldsymbol{\tau}_h - \nabla \cdot (2\eta\boldsymbol{\varepsilon}(\mathbf{v}_h)), -\chi_{\mathrm{s}} \nabla \cdot \boldsymbol{\xi}_h)_K \\
&\quad = \sum_K \alpha_v (\mathbf{f}, -\chi_{\mathrm{s}} \nabla \cdot \boldsymbol{\xi}_h)_K \qquad (52)
\end{aligned}$$

for all $(\mathbf{w}_h, q_h, \boldsymbol{\xi}_h) \in V_{h,0} \times P_{h,0} \times \mathcal{T}_{h,0}$. 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 (50)–(52) uniquely solvable. This can be circumvented by using a sort of augmented Lagrangian scheme coupled with an Uzawa iterative scheme, as explained at the continuous level. To explain how to introduce this, let us write first the matrix form of problems (50)–(52). If X_v, X_p, and X_τ are the nodal values of $\mathbf{v}_h$, p_h, and $\boldsymbol{\tau}_h$, respectively, this matrix form reads

$$\begin{pmatrix} A_{vv} & A_{vp} & A_{v\tau} \\ A_{pv} & A_{pp} & A_{p\tau} \\ A_{\tau v}^{(1)} + A_{\tau v}^{(2)} & A_{\tau p} & A_{\tau\tau} \end{pmatrix} \begin{pmatrix} X_v \\ X_p \\ X_\tau \end{pmatrix} = \begin{pmatrix} F_v \\ F_p \\ F_\tau \end{pmatrix} \qquad (53)$$

where the matrix components and the forcing terms are identified in Table I. There, we have considered the approximation to the temporal derivative written as $\delta_t \mathbf{v}_h = \gamma \mathbf{v}_h - \mathbf{f}_t$, where γ is the coefficient that multiplies the unknown velocity and $\mathbf{f}_t$ is a combination of previous velocity values. For example, for the backward Euler scheme, $\gamma = \Delta t^{-1}$ and $\mathbf{f}_t = \Delta t^{-1} \mathbf{v}_h^{n-1}$, $\mathbf{v}_h^{n-1}$ being the velocity of the previous time step.

The iterative scheme we propose to solve (53) at each time step is

$$\begin{pmatrix} A_{vv} & A_{vp} & 0 \\ A_{pv} & A_{pp} & 0 \\ A_{\tau v}^{(1)} + A_{\tau v}^{(2)} & A_{\tau p} & \frac{1}{2\eta_{\mathrm{s}} r} M \end{pmatrix} \begin{pmatrix} X_v^k \\ X_p^k \\ X_\tau^k \end{pmatrix} = \begin{pmatrix} F_v - A_{v\tau} X_\tau^{k-1} \\ F_p - A_{p\tau} X_\tau^{k-1} \\ F_\tau + \frac{1}{2\eta_{\mathrm{s}} r} M X_\tau^{k-1} - A_{\tau\tau} X_\tau^{k-1} \end{pmatrix} \qquad (54)$$

where M is the mass matrix corresponding to the interpolation of $\boldsymbol{\tau}_h$ (the integral of the product of shape functions) and, as before, r is the penalty parameter. Obviously, the first two equations in

DOI: 10.1002/nme

Table I. Contributions to the matrix form of the problem.

Matrix/vector in (53)	Term in (50)–(52)
A_{vv}	$\rho(\gamma \mathbf{v}_h, \mathbf{w}_h) + \rho(\mathbf{v}_h \cdot \nabla \mathbf{v}_h, \mathbf{w}_h) + 2(\eta \boldsymbol{\varepsilon}(\mathbf{v}_h), \boldsymbol{\varepsilon}(\mathbf{w}_h))$ $+ \sum_K \alpha_v (\rho\gamma \mathbf{v}_h + \rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h - \nabla \cdot (2\eta \boldsymbol{\varepsilon}(\mathbf{v}_h)), \rho \mathbf{v}_h \cdot \nabla \mathbf{w}_h + \nabla \cdot (2\eta \boldsymbol{\varepsilon}(\mathbf{w}_h)))_K$ $+ \sum_K \alpha_p (\nabla \cdot \mathbf{v}_h, \nabla \cdot \mathbf{w}_h) + \sum_K \alpha_\tau (\boldsymbol{\varepsilon}_s(\mathbf{v}_h), \boldsymbol{\varepsilon}_s(\mathbf{w}_h))$
A_{vp}	$-(p_h, \nabla \cdot \mathbf{w}_h) + \sum_K \alpha_v (\nabla p_h, \rho \mathbf{v}_h \cdot \nabla \mathbf{w}_h + \nabla \cdot (2\eta \boldsymbol{\varepsilon}(\mathbf{w}_h)))_K$
$A_{v\tau}$	$(\boldsymbol{\tau}_h, \boldsymbol{\varepsilon}_s(\mathbf{w}_h)) + \sum_K \alpha_v (-\chi_s \nabla \cdot \boldsymbol{\tau}_h, \rho \mathbf{v}_h \cdot \nabla \mathbf{w}_h + \nabla \cdot (2\eta \boldsymbol{\varepsilon}(\mathbf{w}_h)))_K$
A_{pv}	$(q_h, \nabla \cdot \mathbf{v}_h) + \sum_K \alpha_v (\rho\gamma \mathbf{v}_h + \rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h - \nabla \cdot (2\eta \boldsymbol{\varepsilon}(\mathbf{v}_h)), \nabla q_h)_K$
A_{pp}	$\sum_K \alpha_v (\nabla p_h, \nabla q_h)_K$
$A_{p\tau}$	$\sum_K \alpha_v (-\chi_s \nabla \cdot \boldsymbol{\tau}_h, \nabla q_h)_K$
$A^{(1)}_{\tau v}$	$-(\boldsymbol{\varepsilon}_s(\mathbf{v}_h), \boldsymbol{\xi}_h)$
$A^{(2)}_{\tau v}$	$\sum_K \alpha_v (\rho\gamma \mathbf{v}_h + \rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h - \nabla \cdot (2\eta \boldsymbol{\varepsilon}(\mathbf{v}_h)), -\chi_s \nabla \cdot \boldsymbol{\xi}_h)_K$
$A_{\tau p}$	$\sum_K \alpha_v (\nabla p_h, -\chi_s \nabla \cdot \boldsymbol{\xi}_h)_K$
$A_{\tau\tau}$	$\sum_K \alpha_v (\chi_s \nabla \cdot \boldsymbol{\tau}_h, \chi_s \nabla \cdot \boldsymbol{\xi}_h)_K$
F_v	$\langle \mathbf{f} + \rho \mathbf{f}_t, \mathbf{w}_h \rangle + \sum_K \alpha_v (\mathbf{f} + \rho \mathbf{f}_t, \rho \mathbf{v}_h \cdot \nabla \mathbf{w}_h + 2\eta \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{w}_h))_K$
F_p	$\sum_K \alpha_v (\mathbf{f} + \rho \mathbf{f}_t, \nabla q_h)_K$
F_τ	$\sum_K \alpha_v (\mathbf{f} + \rho \mathbf{f}_t, -\chi_s \nabla \cdot \boldsymbol{\xi}_h)_K$

(54) can be solved in a coupled way, and then, the values of X_v^k and X_p^k can be used in the third equation.

Scheme (54) can be slightly modified to treat part of X_τ implicitly in the first equation, in the spirit of the remark after (23). From the third equation in (54), we have

$$\begin{aligned} X_\tau^k &= X_\tau^{k-1} + 2\eta_s r M^{-1} \left[F_\tau - A^{(2)}_{\tau v} X_v^k - A_{\tau p} X_p^k - A_{\tau\tau} X_\tau^{k-1} \right] + 2\eta_s r M^{-1} \left[-A^{(1)}_{\tau v} X_v^k \right] \\ &:= X_\tau^{k-1} + \delta X_\tau^{k,1} + \delta X_\tau^{k,2} \end{aligned}$$

with the obvious definition of arrays $\delta X_\tau^{k,1}$ and $\delta X_\tau^{k,2}$. A possible modification of the first equation in (54) is to replace X_τ^{k-1} by $X_\tau^{k-1} + \delta X_\tau^{k,2}$. From the expression of $A^{(1)}_{\tau v}$ in Table I, it follows that the nodal values $\delta X_\tau^{k,2} = -2\eta_s r M^{-1} A^{(1)}_{\tau v} X_v^k$ are nothing but those of $-2\eta_s r P_\tau(\boldsymbol{\varepsilon}(\mathbf{v}_h))$. This term can therefore be moved to the left-hand side of the first equation in (54). This in fact is achieved by simply replacing $2(\eta \boldsymbol{\varepsilon}(\mathbf{v}_h), \boldsymbol{\varepsilon}(\mathbf{w}_h))$ by $2(\eta \boldsymbol{\varepsilon}(\mathbf{v}_h) + H(\alpha)\eta_s\, r\, P_\tau(\boldsymbol{\varepsilon}(\mathbf{v}_h)), \boldsymbol{\varepsilon}(\mathbf{w}_h))$, as explained for the continuous problem.

5. NUMERICAL EXPERIMENTS

In this section, we present several 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 [40, 41]). The results obtained with the proposed approach, referred as ISM, are compared with solutions obtained either by standard solution (classical boundary conditions) or by other approaches.

5.1. 2D immersion of solid bodies in an incompressible fluid

We first consider a numerical test of flow around four fixed cylinders in a square channel. This study is considered as a first step to investigate the feasibility of fluid–structure computations and could be used later on a tube bundle configuration, such as those existing in nuclear steam generators.

We consider two cases. In the first case, we apply the immersed stress method: (i) we consider a very simple square domain $[0, 1] \times [0, 1]$; (ii) we compute analytically the distance function of four

DOI: 10.1002/nme

circles located on each corner of the domain; (iii) we apply the anisotropic mesh adaptation using the variation of the gradients of the level set function; and finally, (iv) we mix and assign the physical properties. In the second case, the effort will be concentrated on the geometry and on building the fluid mesh while well respecting the curvatures. The obtained finite element meshes that will be used in the two cases are depicted in Figure 2.

We apply the same conditions on both test cases and we compare the solutions. We impose a pressure gradient at the inlet and outlet equal to $\Delta p = 100$. For simplicity, we set the density of the fluid $\rho_f = 1$, the density of the solid $\rho_s = 1$, and the viscosity of the fluid $\eta_f = 1$.

A comparison of the velocity and the pressure fields computed on the entire square domain (fluid and solid) and on only the fluid domain are shown in Figures 3 and 4. The 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. The pressure distribution caused by the interaction is more interesting, which is depicted in Figure 4.

5.2. *Immersion of solid bodies in fluid 3D*

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 through two half-spheres inside a cavity of dimension 1 m^3. As in the 2D test case, a pressure gradient is prescribed ($\Delta p = 100$). We compare, after reaching the steady state, the results to those calculated only on the fluid domain. The mesh representation of the fluid domain case is shown in Figure 5. As sketched in Figure 6, mesh adaptation is required for a good capturing of the solid–fluid interface, and a zoom of the adapted mesh is also depicted. The velocity and pressure fields computed with the proposed method are shown in Figures 7 and 8, respectively, and compared with results calculated only on the fluid

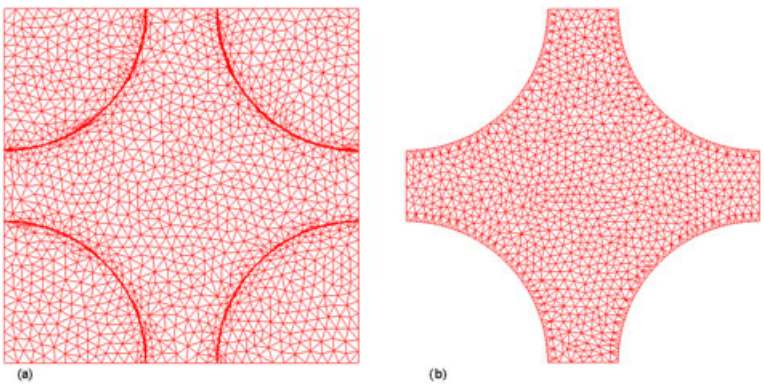


Figure 2. Finite element mesh and geometry of (a) the immersed obstacles and (b) the fluid domain.

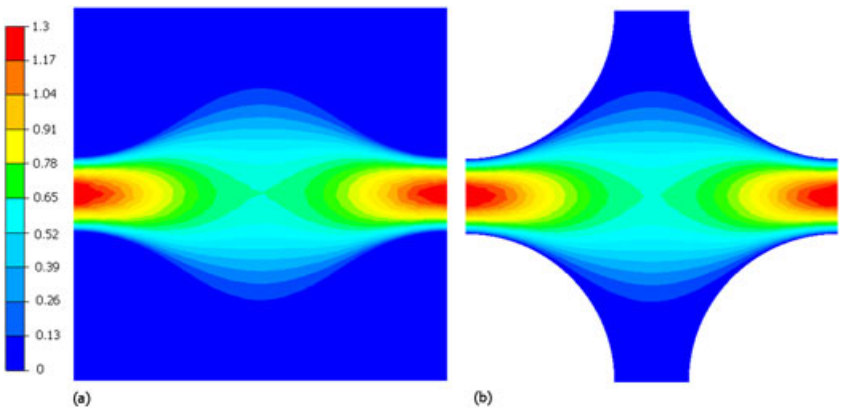


Figure 3. Norm of the velocity calculated over (a) the entire domain and (b) the fluid domain.

DOI: 10.1002/nme

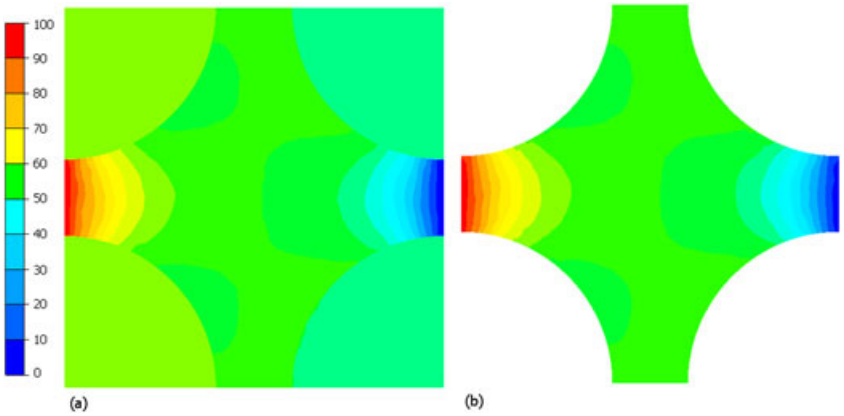


Figure 4. Pressure distribution in (a) the entire domain and (b) the fluid domain.

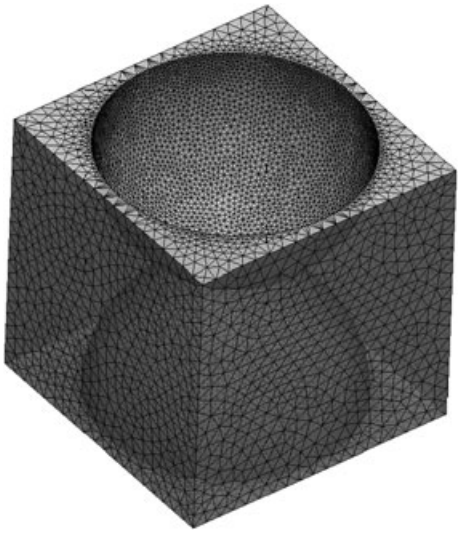


Figure 5. Finite element mesh of the fluid domain.

domain. Again, the agreement between the two numerical simulations shows that the present solver is able to predict accurately the behavior of the fluid and the presence of the solid.

5.3. Flow around a fixed circular cylinder (2D)

We consider in this section another well-studied problem, namely that of a channel flow with a cylindrical obstruction [42–45]. The setting of the problem is shown in Figure 9. It consists of a rectangular channel with a circular obstruction of diameter 0.1. Zero initial condition is considered all over the domain. On the inflow and outflow boundaries, the velocity is defined by the following:

$$u(t;0,y) = u(t;2.2,y) = 0.41^{-2} \sin\left(\frac{\pi t}{8}\right)(6y(0.41-y),0)$$

No-slip conditions are prescribed at the other boundaries.

The circular cylinder is represented by means of a signed distance function (the level set function), and an anisotropic mesh adaptation is applied to capture well the interface. Figure 9 shows the extremely refined solid–fluid interface. The Reynolds number is defined by $\mathrm{Re} = UD/\mu$, where D is the diameter of the cylinder, U is the free-stream mean inflow velocity, and μ is the kinematic viscosity set to 10^{-3} in order to obtain Reynolds numbers $0 \leqslant \mathrm{Re} \leqslant 100$.

DOI: 10.1002/nme

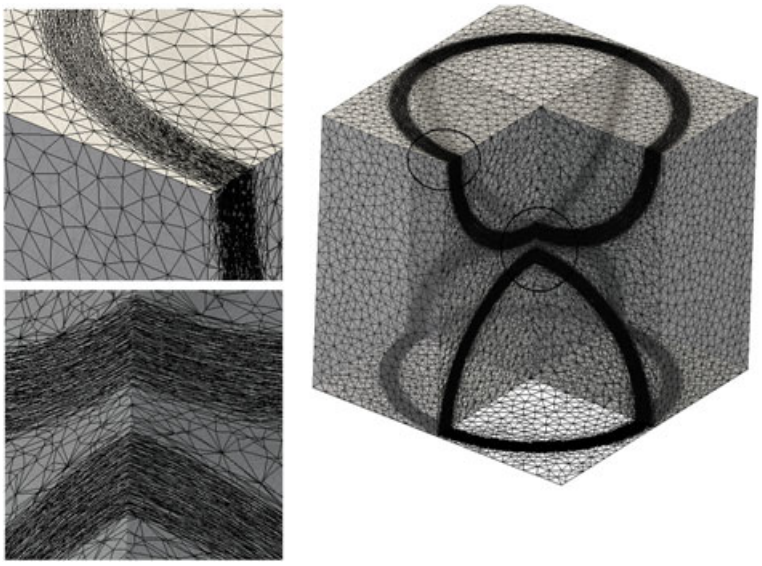


Figure 6. Finite element mesh of two half-spheres in a fluid domain.

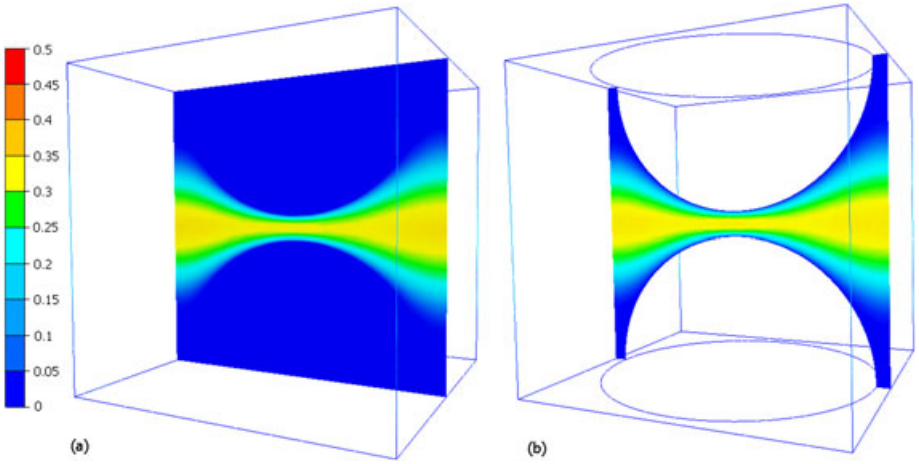


Figure 7. Velocity distribution in (a) the entire domain and (b) the fluid domain.

After a while, from the beginning of the simulation, the inflow increases, and two vortices start to develop behind the cylinder. These vortices are first attached behind the cylinder at a moderate Reynolds number, and by increasing the Reynolds number, they become starched, and the flow will be distorted and broken apart. At a time of around $(4 \sim 5)$, the separation of the vortices from the cylinder occurs, and an alternative vortex shedding known as Karmen vortex street develops. Important benchmark parameters of the flows around bodies are the drag and the difference in the pressure between two points at the edge of the obstacle.

Figure 10 shows the profiles of the benchmark parameters obtained on a 40,000 elements mesh with a constant time-step $\Delta t = 0.0025$. The plots show the good tendency of the approximated solution obtained with the numerical scheme to reproduce the profile of the coefficients.

Recall that the drag coefficient can be computed using

$$c_{\mathrm{d}} = \frac{F_x}{\frac{1}{2}\rho {u_\infty}^2 D} \tag{55}$$

DOI: 10.1002/nme

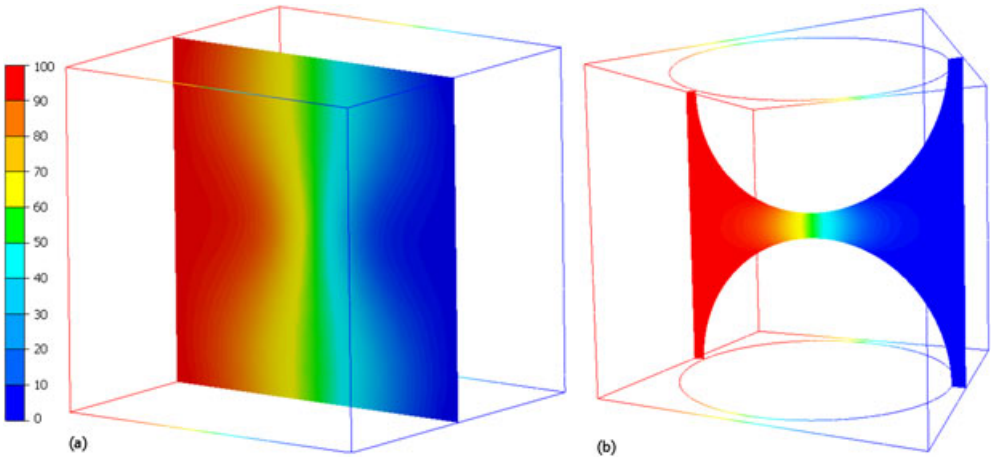


Figure 8. Pressure distribution in (a) the entire domain and (b) the fluid domain.

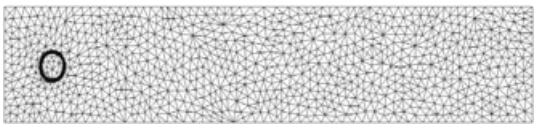
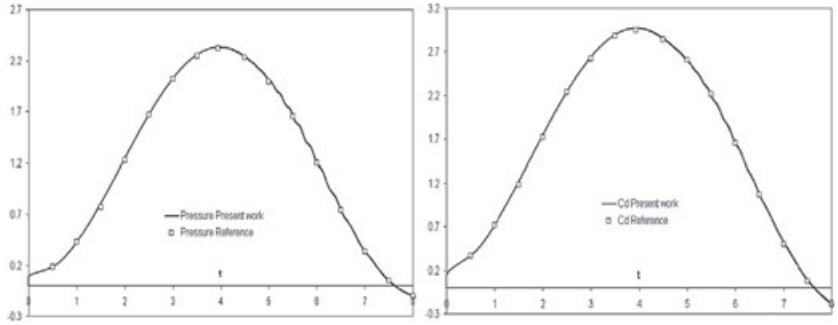


Figure 9. Finite element mesh around the circular cylinder.



(a) $P_{diff} = P(0.15, 0.2) - P(0.25, 0.2)$ (b) Drag coefficient

Figure 10. Comparisons of the pressure difference and the drag coefficients with results from [45]. (a) $P_{\text{diff}} = P(0.15, 0.2) - P(0.25, 0.2)$ and (b) drag coefficient.

where u_∞ is the maximum inflow velocity and F_x is the x-component of $\mathbf{F}$, the total force on the cylinder surface S:

$$\mathbf{F} = \int_S (-p\mathbf{I} + \mathbf{S}) \cdot \mathbf{n} \, \mathrm{d}S \tag{56}$$

$\mathbf{S}$ being the deviatoric stresses.

5.4. Oscillating circular cylinder in a channel

This test has been widely used as a benchmark for rigid particulate flows and has been analyzed by a number of authors [46]. We consider an oscillating circular cylinder of diameter 0.1 inside a channel (height $H = 0.41$, length $L = 2.2$). The cylinder moves from his initial position $(1.1, 0.2)$ and starts oscillating along with a prescribed velocity $u = 2\pi f A \cos(2\pi f t)$, with $A = 0.25$, $f = 0.25$, and $v = 0$. No-slip velocity conditions are imposed at the two walls, inlet and outlet of the channel.

DOI: 10.1002/nme

The dynamic viscosity of the fluid is $\eta = 10^{-3}$, and the density is equal to $\rho = 1$. The Reynolds number is ~ 50. The fluid in the channel is initially at rest. A time step equal to 0.005 is used.

We compute first the level set function that identifies the solid part from the fluid region and then apply the anisotropic mesh adaptation at the interface. Thus, a single set of equations is solved for the whole computational domain by treating the different subdomains as a single fluid with variable material properties. As the cylinder starts oscillating, we repeat the adaptivity steps to ensure accurate interface representation. Figures 11 and 12 show two snapshots results at two different times ($t = 18.9$ s and $t = 21$ s), which reflect the oscillating behavior of the cylinder. For comparisons, we present in Figure 13 the computed drag coefficient c_d. As expected, the presented results using the immersed stress method agree very well with the reference results.

5.5. *Falling cylinder in an incompressible fluid*

We present here the same test case as in [47] to analyze the terminal velocity v_T of a falling cylinder in an incompressible fluid. The same parameters as in this reference 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 1 000 kg/m^3, and the solid density is 2 000 kg/m^3. The gravitational force is 9.8 m s^{-1}. Note that, for comparisons, we have chosen from [47] 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}, and 0.5 kg s^{-1}.

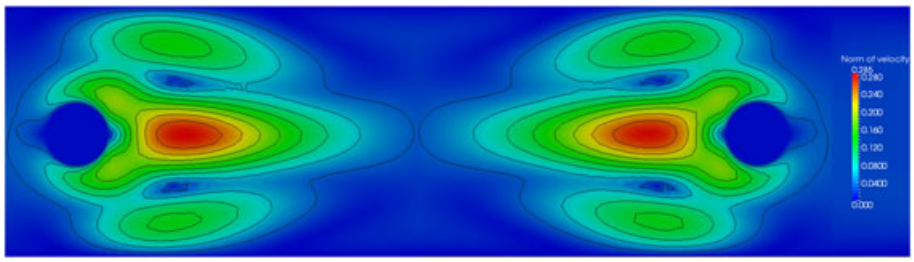


Figure 11. Norm of velocity at (left) $t = 18.9$ s and (right) $t = 21$ s.

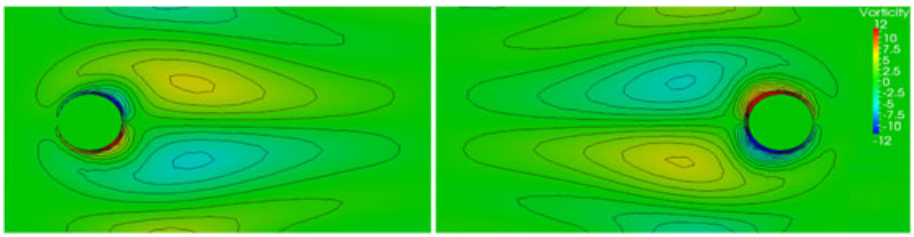


Figure 12. Local vorticity value at (left) $t = 18.9$ s and (right) $t = 21$ s.

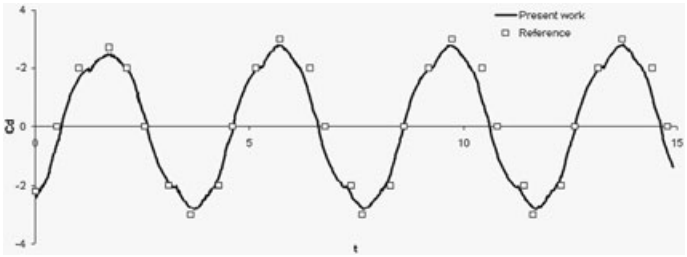


Figure 13. Drag coefficients for one oscillating circular cylinder in a channel and comparisons with results from [46].

DOI: 10.1002/nme

All the computed data are presented in Table II. Note that we are solving the Navier–Stokes equations, whereas the terminal velocity is derived under a Stokes flow assumption. As a result, the computed terminal velocity will be usually lower than the analytic solution, as stated in [47]. As observed in Table II, we obtain a closer agreement on the finest mesh and a convergence order close to 2. Robinson *et al*. [47] solved the FSI problem in a monolothic framework using a Marker and Cell (MAC) grid discretization of the fluid and a fully Lagrangian discretization of the structure and obtained an order of convergence equal to 1. Comparing our computational results to those reported in [47], we can find good agreements for the terminal velocities.

Table II. Terminal velocity for the falling cylinder problem.

η_f	Reference [47] 160×640	$h = 0.002$	$h = 0.001$	$h = 0.0005$	$h = 0.00025$	Convergence order
0.1	−0.1966	−0.1503069	−0.167770	−0.19230704	−0.196834	~ 2.3
0.2	−0.1417	−0.10756319	−0.12367	−0.13596088	−0.14087306	~ 1.8
0.5	−0.06721	−0.04301721	−0.0579240	−0.0645362	−0.066432	~ 1.7

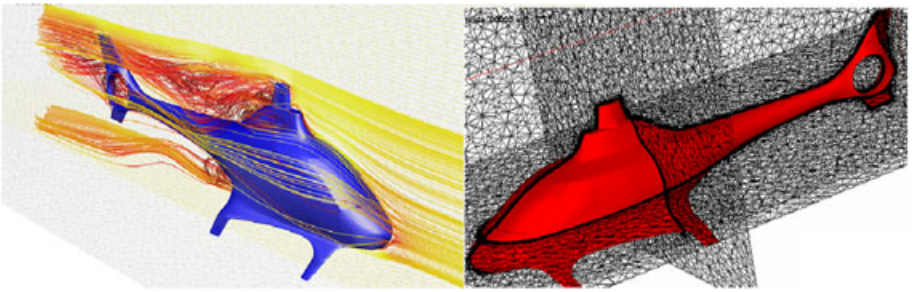


Figure 14. Numerical simulation of unsteady flow around a helicopter in forward flight.

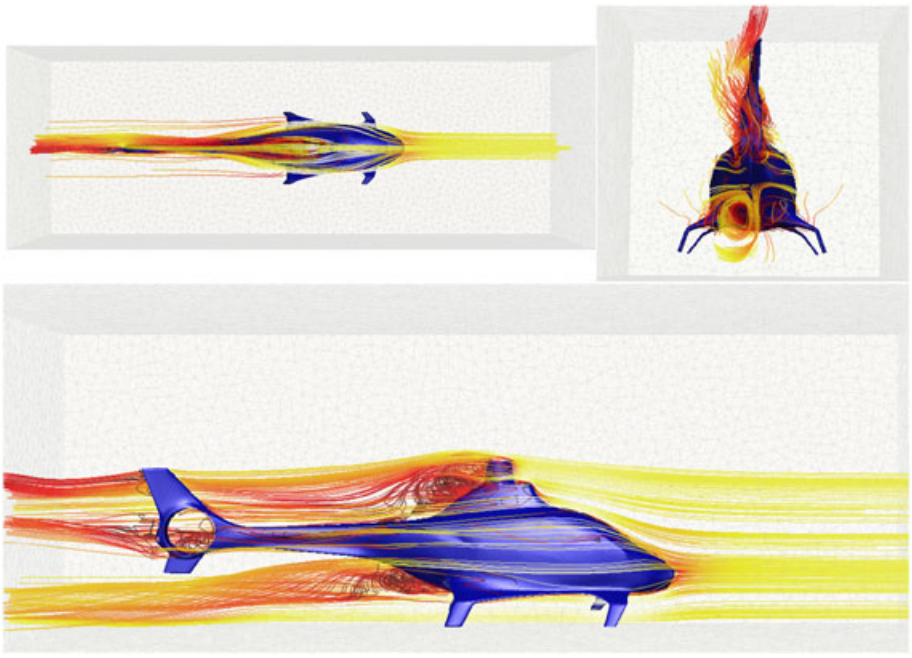


Figure 15. Different snapshots of the flow around a helicopter in forward flight.

DOI: 10.1002/nme

5.6. *Unsteady flow past an immersed helicopter in forward flight*

Figures 14 and 15 present the parallel numerical simulation of unsteady flow around a 3D helicopter in forward flight using the proposed monolithic fluid–structure approach with anisotropic mesh adaptation. The mesh generation algorithm allows the creation of extremely anisotropic elements stretched along the interface, which is an important requirement for FSI problems having internal/boundary layers. The final obtained mesh reflects the capability of the method to render a well-respected geometry in terms of curvature, angles, and complexity. Contrary to other techniques, this promising method can provide an alternative to body-fitted mesh for very complex geometry.

This simulation is obtained using 96 2.4-GHz cores in parallel. The mesh consists of $\sim 1.6M$ tetrahedral elements and $\sim 300,000$ nodes. The Reynolds number at the inlet is $\sim 1.5^7$. The reasonable nature of the results shows a good potential for the developed formulations. The purpose of this example is to confirm the motivation behind pursuing such general approach. Indeed, it allows to easily (i) deal with a large diversity of complex shapes and dimensions without mesh reconstructions; (ii) use and affect different physical properties for the surrounding fluid (air, water, and others) and for the immersed structures; and (iii) handle the interfaces through anisotropic mesh adaptation.

6. CONCLUSION

In this paper, we have described a stabilized three-field velocity–pressure–stress, designed for the computation of rigid bodies in an incompressible Navier–Stokes flow at high Reynolds number. The method is based on treating a single set of equation for the whole domain. 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 three fields. We combine this approach with the mesh adaptivity to resolve complex structure geometries. 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 at high Reynolds number was demonstrated. The results presented here show that this method can be used in a wide range of application for multibody fluid–solid problems. Further research will focus on tackling deformable solid interaction.

ACKNOWLEDGEMENT

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

REFERENCES

1. Hirt CW, Amsden AA, Cook JL. An arbitrary Lagrangian-Eulerian computing method for all speeds. *Journal of Computational Physics* 1974; **14**(3):227–253.
2. Hughes TJR, Liu WK, Zimmerman T. Lagrangian-Eulerian finite element formulation for incompressible viscous flow. *Computer Methods in Applied Mechanics and Engineering* 1981; **29**(3):239–249.
3. Donea J, Giuliani S, Halleux JP. An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering* 1982; **33**(1–3):689–723.
4. Benson D. An efficient, accurate, simple ALE method for non linear finite element programs. *Computer Methods in Applied Mechanics and Engineering* 1989; **72**(3):305–350.
5. Souli M, Ouahsine A, Lewin L. ALE formulation for fluid structure interaction problems. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**(5–7):659–675.
6. van Loon R, Anderson PD, van de Vosse FN, Sherwin SJ. Comparison of various fluid-structure interaction methods for deformable bodies. *Computers and Structures* 2007; **85**(11–14):833–843.
7. 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*, Bletzinger KU, Schweizerhof K, Wall WA (eds). CIMNE: Barcelona, 2001; 689–698.
8. Le Tallec P, Mouro J. Fluid structure interaction with large structural displacements. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**(24–25):3039–3067.
9. Mok DP, Wall WA, Ramm E. Partitioned analysis approach for the transient, coupled response of viscous fluids and flexible structures. In *Proceedings of the European Conference on Computational Mechanics, ECCM'99*, Wunderlich W (ed.). TU Munich, 1999.
10. Fernández MA, Moubachir M. A Newton method using exact Jacobians for solving fluid-structure coupling. *Computers and Structure* 2005; **83**(2–3):127–142.

DOI: 10.1002/nme

11. Gerbeau JF, Vidrascu M. A quasi-Newton algorithm based on a reduced model for fluid structure interaction problems in blood flow. *Mathematical Modelling and Numerical Analysis* 2003; **37**:631–647.
12. Gerbeau JF, Vidrascu M, Frey P. Fluid structure interaction in blood flows on geometries coming from medical imaging. *Computers and Structure* 2005; **83**(2–3):155–165.
13. Küttler U, Wall AW. Fixed-point fluidstructure interaction solvers with dynamic relaxation. *Computational Mechanics* 2008; **43**:61–72.
14. Caussin P, Gerbeau JF, Nobile F. Added-mass effect in the design of partioned algorithms for fluid-structure problems. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**:4506–4527.
15. Farhat C, Rallu A, Wang K, Belytschko T. Robust and provably second-order explicit-explicit and implicit-explicit staggered time-integrators for highly non-linear compressible fluid-structure interaction problems. *International Journal for Numerical Methods in Fluids* 2010; **84**(1):73–107.
16. Farhat C, Lesoinne M, LeTallec P. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering* 1998; **157**(1–2):95–114.
17. Kreiss HO, Petersson A. A second order accurate embedded boundary method for the wave equation with dirichlet data. *SIAM Journal on Scientific Computing* 2006; **27**(4):1141–1167.
18. Wang K, Rallu A, Gerbeau JF, Farhat C. Algorithms for interface treatment and load computation in embedded boundary methods for fluid and fluidstructure interaction problems. *International Journal for Numerical Methods in Fluids* 2011; **67**(9):1175–1206.
19. Peskin CS. The immersed boundary method. *Acta Numerica* 2002; **11**:479–517.
20. Glowinski R, Pan TW, Helsa TI, Joseph DD. A distributed Lagrange multiplier/fictious domain methods for particulate flows. *International Journal of Multiphase Flow* 1999; **25**(5):755–794.
21. Codina R, Houzeaux J, Coppola-Owen H, Baiges J. The fixed-mesh ALE approach for the numerical approximation of flows in moving domains. *Journal of Computational Physics* 2009; **228**:1591–1611.
22. Baiges J, Codina R. The fixed-mesh ALE approach applied to solid mechanics and fluid-structure interaction problems. *International Journal for Numerical Methods in Engineering* 2010; **81**:1529–1557.
23. Baiges J, Codina R, Coppola-Owen H. The fixed-mesh ALE approach for the numerical simulation of floating solids. *International Journal for Numerical Methods in Fluids* 2011; **67**:1004–1023.
24. Feghali S, Hachem E, Coupez T. Monolithic stabilized finite element method for rigid body motions in the incompressible Navier-Stokes flow. *European Journal of Computational Mechanics* 2010; **19**(5–7):547–573.
25. Coupez T, Hugues D, Hachem E, Laure P, Silva L, Valette R. Multidomain finite element computations: application to multiphasic problems. In *Arbitrary Lagrangian-Eulerian and Fluid-Structure Interaction*, Souli M, Benson DJ (eds), Numerical Simulation. Wiley, 2010; 221–289.
26. Hachem E, Rivaux B, Kloczko T, Digonnet H, Coupez T. Stabilized finite element method for incompressible flows with high Reynolds number. *Journal of Computational Physics* 2010; **229**(23):8643–8665.
27. Hachem E, Kloczko T, Digonnet H, Coupez T. Stabilized finite element solution to handle complex heat and fluid flows in industrial furnace using the immersed volume method. *International Journal for Numerical Methods in Fluids* 2012; **68**(1):99–121.
28. 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* 2009; **78**(8):980–1008.
29. Codina R, Soto O. A numerical model to track two-fluid interfaces based on a stabilized finite element method and the level set technique. *International Journal for Numerical Methods in Fluids* 2002; **40**:293–301.
30. van der Pijl SP, 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* 2005; **47**:339–361.
31. 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* 2006; **194**:4951–4976.
32. Coupez T. Génération de maillage et adaptation de maillage par optimisation locale. *Revue Européenne des Éléments Finis* 2000; **9**:403–423.
33. Hughes TRJ. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origin of stabilized methods. *Computer Methods in Applied Mechanics and Engineering* 1995; **127**:387–401.
34. Codina R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:4295–4321.
35. Badia S, Codina R. Stabilized continuous and discontinuous Galerkin techniques for Darcy flow. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**:1654–1667.
36. Codina R. Finite element approximation of the three field formulation of the Stokes problem using arbitrary interpolations. *SIAM Journal on Numerical Analysis* 2009; **47**:699–718.
37. Codina R, Principe J, Guasch O, Badia S. Time dependent subscales in the stabilized finite element approximation of incompressible flow problems. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**:2413–2430.
38. Badia S, Codina R. On a multiscale approach to the transient Stokes problem. Transient subscales and anisotropic space-time discretization. *Applied Mathematics and Computation* 2009; **207**:415–433.
39. Cervera M, Chiumenti M, Codina R. Mixed stabilized finite element methods in nonlinear solid mechanics: Part I: formulation. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**:2559–2570.

DOI: 10.1002/nme

40. Digonnet H, Coupez T. Object-oriented programming for fast and easy development of parallel applications in forming processes simulation. In *Second MIT Conference on Computational Fluid and Solid Mechanics*, Bathe KJ (ed.). Massachusetts Institute of Technology: Cambridge, MA 02139 U.S.A., 2003; 1922–1924.
41. Mesri Y, Digonnet H, Coupez T. Advanced parallel computing in material forming with CIMLIB. *European Journal of Computational Mechnanics* 2009; **18**:669–694.
42. Schäfer M, Turek S. The benchmark problem 'flow around a cylinder'. In flow simulation with high-performance computers II. *Notes on Numerical Fluid Mechanics* 1996; **52**:547–566.
43. Volker J, Gunar M. Higher order finite element discretizations in a benchmark problem for incompressible flows. *International Journal for Numerical Methods in Fluids* 2001; **37**:885–903.
44. Frochte J, Heinrichs W. A splitting technique of higher order for the Navier-Stokes equations. *Journal of Computational and Applied Mathematics* 2009; **228**(1):373–390.
45. Volker J. Reference values for drag and lift of a two-dimensional time-dependent flow around a cylinder. *International Journal for Numerical Methods in Fluids* 2004; **44**(7):777–788.
46. Wan D, Turek S. Fictitious boundary and moving mesh methods for the numerical simulation of rigid particulate flows. *Journal of Computational Physics* 2007; **222**:28–56.
47. Robinson A, Schroeder C, Fedkiw R. A symmetric positive definite formulation for monolithic fluid structure interaction. *Journal of Computational Physics* 2010; **230**(4):1547–1566.

DOI: 10.1002/nme