Fluid-rigid body interaction using the finite element method and ale formulation : framework, implementation and benchmarking

Luca Berti, Vincent Chabannes, Laetitia Giraldi, Christophe Prud'Homme

FLUID-RIGID BODY INTERACTION USING THE FINITE ELEMENT METHOD AND ALE FORMULATION: FRAMEWORK, IMPLEMENTATION AND BENCHMARKING

Luca Berti
Cemosis, IRMA UMR 7501, CNRS
Université de Strasbourg, France
berti@math.unistra.fr

Vincent Chabannes
Cemosis, IRMA UMR 7501, CNRS
Université de Strasbourg, France
chabannes@math.unistra.fr

Laetitia Giraldi
CALISTO team, INRIA
Université Côte d'Azur, France
laetitia.giraldi@inria.fr

Christophe Prud'homme
Cemosis, IRMA UMR 7501, CNRS
Université de Strasbourg, France
prudhomm@math.unistra.fr

ABSTRACT

We consider the problem of moving rigid bodies in a Newtonian fluid. The fluid-solid problem is solved using the finite element method, and the solution is based on the Arbitrary-Lagrangian-Eulerian description of the fluid with conforming treatment of moving interfaces. Numerical experiments are discussed, and the results are validated with the literature. The results are obtained with the `Feel++` open source library [17].

Introduction

The simulation of moving rigid particles immersed in a fluid is a fundamental component in numerical simulations of industrial systems like fluidised bed reactors [9] or oil wells [18]. Recovering the motion of rigid bodies is also a first step in the numerical simulation of deformable immersed bodies, e.g. biological organisms or swimming robots.

Different methods have been proposed to simulate moving rigid bodies in a fluid, as various difficulties are associated to this problem.

When the Reynolds number is high and the fluid is turbulent, in order to capture the vortices and their effects on the immersed object, spectral methods are often used, since they solve accurately the different modes that appear in the flow [19, 2, 21]. Even when the flow is laminar and the Reynolds number is low, spectral methods are effective (see for instance [15], where the authors investigate the sedimentation of particles with different shapes). Other approaches allowing to solve the fluid-solid interaction problem are the Lattice Boltzmann method, where the discrete Boltzmann equations are solved to obtain the fluid velocity distribution [23], or the finite volume method [3].

In order to follow the motion of the immersed object, different coupling strategies have been devised. The immersed boundary method is one of the possible approaches [16, 20]. It represents the object via an implicit equation, and then recovers its displacement by using penalty methods on a fixed grid. In [11], an extension of the immersed boundary method (penalized immersed boundary method) is proposed to handle the motion of massive rigid bodies. In this case, the rigid body is represented by two components, linked together with springs of large stiffness: the first component, which interacts with the fluid, is massless, and it transfers the fluid forces and torques to the second massive component, which instead moves as if it was in vacuum.

In this paper, we focus on the Arbitrary-Lagrangian-Eulerian formulation of the fluid problem [5], and we solve it using the finite element method. This approach was also proposed in [13, 12] to study the motion of multiple rigid particles in a Navier-Stokes flow. It consists in partially decoupling the motion of the fluid domain from the motion

of the fluid continuum to follow the motion of the solid and to deform the mesh in agreement with the motion of the latter. This choice leads us to address questions of mesh adaptation, interpolation between meshes and interface discretization in order to ensure the conformity of the fluid-solid interface and good accuracy of our results. The method presented below is validated on benchmarks and examples from the literature, and its implementation is open source in the `Feel++` GitHub repository[1][17].

1 Rigid body moving in a fluid

1.1 Mathematical formulation

The fluid model In this article, we consider rigid bodies immersed in a Newtonian incompressible fluid where the fluid-solid interface is fitted and conforming, and the motion of the fluid domain is described using the Arbitrary-Lagrangian-Eulerian (ALE) frame [8].

Let $\mathcal{F}_t \subseteq \mathbb{R}^d, d \in \{2,3\}$, denote the domain occupied by the fluid at time t, where $t \in [0,T]$. Let $\mathcal{A}_t : \mathcal{F}_0 \to \mathcal{F}_t$ be the ALE map relating the current fluid domain with the reference domain, defined as $\mathcal{A}_t(X) = \mathcal{A}(t,X) = X + x(t,X)$ with $x(t,X)$ the domain's displacement from its reference position. Let $u : \mathcal{F}_t \times [0,T] \to \mathbb{R}^d$ and $p : \mathcal{F}_t \times [0,T] \to \mathbb{R}$ the fluid velocity and (hydrostatic) pressure. Let ρ_f and μ be the fluid constant density and dynamic viscosity.

Using the ALE frame, the evolution of the fluid continuum is partially decoupled from the evolution of the domain and the ALE time derivative is substituted to the Eulerian one $\partial_t u$ as $\partial_t u = \partial_t u|_{\mathcal{A}} - (\partial_t x \cdot \nabla)u = \partial_t u|_x - (u_{\mathcal{A}} \cdot \nabla)u$. The first term corresponds to the time variation of the fluid's velocity as seen from the ALE frame, while the second accounts for the relative velocity between the fluid continuum and the ALE frame.

The Navier-Stokes equations in moving domain, describing the evolution of the incompressible fluid continuum, are

$$\begin{aligned} \rho_f \partial_t u|_{\mathcal{A}} + \rho_f \Big((u - u_{\mathcal{A}}) \cdot \nabla\Big) u &= -\nabla p + \mu \Delta u, && \text{in } \mathcal{F}_t, \\ \nabla \cdot u &= 0, && \text{in } \mathcal{F}_t. \end{aligned} \tag{1}$$

The rigid body equations The dynamics of a rigid body is described by the motion of its center of mass and the evolution of its orientation. In the cases we will consider, the motion is caused by gravity and fluid stresses, that act as the external forces F_e and torques T_e in the Newton and Euler equations describing the time evolution of the center of mass and body orientation.

Let $\mathcal{S} \subset \mathbb{R}^d$ be the domain occupied by the rigid solid, $\rho \in \mathbb{R}_{>0}$ its density and $m = \int_{\mathcal{S}} \rho$ its mass. Let $x^{CM} = m^{-1} \int_{\mathcal{S}} \rho x$ be its center of mass, $I = \int_{\mathcal{S}} \rho (x - x^{CM}) \otimes (x - x^{CM})$ its inertia tensor. Let $U : [0,T] \to \mathbb{R}^d$ and $\omega : [0,T] \to \mathbb{R}^{d^*}$, where $d^* = 1$ if $d = 2$ or $d^* = 3$ if $d = 3$, be the linear and angular velocity of the rigid body as seen from the laboratory frame, $F_e \in \mathbb{R}^d, T_e \in \mathbb{R}^{d^*}$ be the external forces and torques acting on the body. Let $\theta \in \Theta$, where $\Theta = [-\pi, \pi]$ if $d = 2$, or $\Theta = [-\pi,\pi] \times [0,\pi] \times [0,\pi/2]$ if $d = 3$, be the angles that provide the orientation of the body. Let $R(\theta) : \Theta \to \mathbb{R}^{d \times d}$ be the orthogonal matrix describing the change of reference frame between the body frame and the laboratory frame. The Newton and Euler equations, describing the dynamics of a three-dimensional rigid body, are

$$\begin{aligned} m \frac{d}{dt} U &= F_e - \int_{\partial \mathcal{S}} -p\vec{n} + 2\mu(\nabla u + \nabla u^T)\vec{n}, \\ \frac{d}{dt}(RIR^T\omega) &= T_e - \int_{\partial \mathcal{S}} [-p\vec{n} + 2\mu(\nabla u + \nabla u^T)\vec{n}] \times (x - x^{CM}), \end{aligned} \tag{2}$$

where

$$\begin{aligned} \frac{d}{dt}\theta_i &= \omega_i, \quad \text{for } i \in \{x,y,z\}, \\ R &= R_z(\theta_z) R_y(\theta_y) R_x(\theta_x), \end{aligned} \tag{3}$$

and $R(\theta_i)$ denotes the rotation matrix around axis $i \in \{x,y,z\}$ of angle θ_i. In the two-dimensional case, I can be considered as a scalar quantity and $R(\theta)$ has the form

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}, \tag{4}$$

[1] https://github.com/feelpp/feelpp.

while in the three-dimensional case

$$R_z(\theta_z) = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_y(\theta_y) = \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix},$$

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix}. \quad (5)$$

Fluid-solid interaction The interaction between the fluid and the rigid body is described by two continuity conditions: the matching of the fluid and rigid body velocity at the common interface, and the transmission of fluid forces and torques to the immersed solid. The first condition translates as

$$u = U + \omega \times (x - x^{CM}) \quad \text{on } \partial\mathcal{F}_t \cap \partial\mathcal{S}, \quad (6)$$

while the second requires that, in absence of any other external forces,

$$m\frac{d}{dt}U = \int_{\partial\mathcal{S}} p\vec{n} - 2\mu(\nabla u + \nabla u^T)\vec{n}, \quad \frac{d}{dt}(RIR^T\omega) = \int_{\partial\mathcal{S}} [p\vec{n} - 2\mu(\nabla u + \nabla u^T)\vec{n}] \times (x - x^{CM}), \quad (7)$$

where $\vec{n}$ is the unit outward normal to $\partial\mathcal{S}$. The velocity at the body's interface is also needed to compute the displacement $x(t, X)$ in the definition of $\mathcal{A}_t(X)$. In order to find $x(t, X)$ we solve

$$\begin{aligned} \nabla \cdot ([1 + \tau(X)]\nabla_X x(t, X)) = 0 \qquad & \text{in } \mathcal{F}_0, \\ x(t, X) = g(t, X) \qquad & \text{in } \partial\mathcal{F}_0, \end{aligned} \quad (8)$$

where $g(t, X) = \int_0^t U + \omega \times (X - X^{CM})$ is the rigid displacement of $\mathcal{S}$ and $\tau(X)$ is a space-dependent coefficient whose definition is reported to subsection 1.2, where we discuss the discretized problem. In this last equations one sees how the rigid body impacts the motion of the fluid domain, and that satisfying the coupling condition will require to iterate between the solution of the fluid equations and the geometry of the fluid domain.

Variational formulation The variational formulation of equations (1)-(2)-(8) is proposed in this paragraph to illustrate the choice of test functions that avoids the numerical computation of fluid stresses at the boundary of the solid. The formulation will also be the starting point for the finite element discretization of the problem.

Following [12], let us consider the trial functions $u \in [H^1(\mathcal{F}_t)]^d$, $p \in L^2(\mathcal{F}_t)$, $U \in [L^2([0, T])]^d$, $\omega \in [L^2([0, T])]^{d^*}$, and test functions $\tilde{u} \in [H^1(\mathcal{F}_t)]^d, \tilde{p} \in L^2(\mathcal{F}_t), \tilde{U} \in [L^2([0, T])]^d, \tilde{\omega} \in [L^2([0, T])]^{d^*}$ such that $\tilde{u} = \tilde{U} + \tilde{\omega} \times (x - x^{CM})$ on $\partial\mathcal{S}$. By encoding velocity continuity at the interface in the test space, the following equality holds

$$\begin{aligned} \int_{\partial\mathcal{S}} [-p\vec{n} + 2\mu(\nabla u + \nabla u^T)\vec{n}] \cdot \tilde{u} \, \mathrm{d}S = \\ \int_{\partial\mathcal{S}} [-p\vec{n} + 2\mu(\nabla u + \nabla u^T)\vec{n}] \cdot \tilde{U} \, \mathrm{d}S + \int_{\partial\mathcal{S}} [-p\vec{n} + 2\mu(\nabla u + \nabla u^T)\vec{n}] \cdot [\tilde{\omega} \times (x - x^{CM})] \, \mathrm{d}S, \end{aligned} \quad (9)$$

which relates the boundary terms containing fluid stresses in the fluid momentum, Newton and Euler equations. The numerical computation of these boundary integrals is avoided thanks to the previous equality, as they can be substituted by the volume integrals of the corresponding equations, that is

$$\begin{aligned} \int_{\mathcal{F}t} \rho_f \Big(\partial_t u|_{\mathcal{A}} + (u - u_{\mathcal{A}}) \cdot \nabla u \Big) \cdot \tilde{u} \, dx + 2\mu \int_{\mathcal{F}_t} [(\nabla u + \nabla u^T)] : [(\nabla \tilde{u} + \nabla \tilde{u}^T)] \, dx - \int_{\mathcal{F}_t} p \nabla \cdot \tilde{u} \, dx \\ + m\frac{dU}{dt} \cdot \tilde{U} + \frac{d[R(t)IR(t)^T\omega]}{dt} \cdot \tilde{\omega} = 0. \end{aligned} \quad (10)$$

The variational formulation of (8) reads: find $x \in [H^1(\mathcal{F}_0)]^d$ such that, for all $\phi \in [H^1(\mathcal{F}_0)]^d$, the following equation holds

$$\int_{\mathcal{F}_0} ([1 + \tau(X)]\nabla_X x(t, X)) : \nabla\phi(X) = 0, \quad (11)$$

and $x(t, X) = \int_0^t U + \omega \times (X - X^{CM})$ on $\partial\mathcal{S}$.

1.2 Numerical method

Let $\mathcal{T}_0$ be the triangulation discretizing the fluid domain $\mathcal{F}_0$ at time $t = 0$, which coincides with the computational domain $\mathcal{F}_0$. Since the computational domain $\mathcal{F}_t$ is moving according to the ALE maps $\mathcal{A}_t$, also the triangulation will be moving according to a discrete approximation of the ALE maps, that we denote $\mathcal{A}_h^t$. For each time instant t_n, $0 \leq n \leq n_T$, in which the time interval $[0, T]$ is discretized, it is possible to define the triangulation $\mathcal{T}_{t_n} = \mathcal{A}_h^{t_n}(\mathcal{T}_0)$.

Let u_h^n and p_h^n denote the discrete approximations of the velocity and pressure fields at time t_n. The discrete approximation spaces for the fluid variables are now defined: as the domain is time dependent, the functional spaces are time dependent as well via the discrete ALE maps. In particular, they are related to the discrete approximation spaces on the reference domain via $\mathcal{A}_h^t$. The velocity and pressure spaces are, respectively,

$$\begin{aligned} V_h^t &= \{v : \mathcal{F}_t \to \mathbb{R}^d,\, v = \hat{v} \circ (\mathcal{A}_h^t)^{-1},\, \hat{v} \in [H^1(\mathcal{F}_0)]^d \cap [\mathbb{P}_N(\mathcal{F}_0)]^d\} \\ Q_h^t &= \{p : \mathcal{F}_t \to \mathbb{R},\, p = \hat{p} \circ (\mathcal{A}_h^t)^{-1},\, \hat{p} \in \mathbb{P}_{N-1}(\mathcal{F}_0)\}. \end{aligned} \tag{12}$$

We choose the Taylor-Hood finite element spaces $V_h^t - Q_h^t$ with $N = 2$ for our finite element simulations. Therefore, the velocity is discretized using continuous piecewise quadratic finite elements, and the pressure is discretized using continuous piecewise affine finite elements. The discrete variational formulation of the Navier-Stokes equations at time t_{n+1} requires finding $(u_h^{n+1}, p_h^{n+1}) \in V_h^{t_{n+1}} \times Q_h^{t_{n+1}}$, $(U^{n+1}, \omega^{n+1}) \in \mathbb{R}^d \times \mathbb{R}^{d^*}$ such that, for all $(\tilde{u}, \tilde{p}) \in V_h^{t_{n+1}} \times Q_h^{t_{n+1}}, (\tilde{U}, \tilde{\omega}) \in \mathbb{R}^d \times \mathbb{R}^{d^*}$

$$\begin{aligned} \int_{\mathcal{F}_{t_{n+1}}} \rho_f \partial_t u_h^{n+1}|_{\mathcal{A}} \cdot \tilde{u} + \int_{\mathcal{F}_{t_{n+1}}} \rho_f((u_h^{n+1} - u_{\mathcal{A}_h}^{n+1}) \cdot \nabla_x u_h^{n+1}) \cdot \tilde{u} + \frac{d[R(t) I R(t)^T \omega]}{dt}^{n+1} \cdot \tilde{\omega} + m \frac{dU}{dt}^{n+1} \cdot \tilde{U} \\ + 2\mu \int_{\mathcal{F}_{t_{n+1}}} [(\nabla u_h^{n+1} + \nabla u_h^{n+1,T})] : [(\nabla \tilde{u} + \nabla \tilde{u}^T)]\, dx - \int_{\mathcal{F}_{t_{n+1}}} p_h^{n+1} \nabla_x \cdot \tilde{u} = \int_{\mathcal{F}_{t_{n+1}}} f^{n+1} \cdot \tilde{u}, \end{aligned} \tag{13}$$

$$\int_{\mathcal{F}_{t_{n+1}}} \tilde{p} \nabla_x \cdot u_h^{n+1} = 0. \tag{14}$$

Let us now denote the degrees of freedom that belong to the boundary of the rigid body by the subscript Γ as u_Γ, p_Γ and the others by the subscript I as u_I, p_I.

As we have previously said, boundary terms of the form $\int_{\partial \mathcal{S}} (-p\mathbb{I} + 2\mu D(u))\vec{n} \cdot \tilde{u}\, dS$ are never computed in the assembly of the system matrix. Instead of building the finite element basis spanning the constrained test space of $(\tilde{u}, \tilde{U}, \tilde{\omega}) \in [H^1(\mathcal{F}_{t_{n+1}})]^d \times \mathbb{R}^d \times \mathbb{R}^{d^*}$ that satisfies boundary conditions $u = U + \omega \times (x - x^{CM})$ on $\partial \mathcal{S}$, we first use the standard finite element bases to discretize equations (13)-(14), getting

$$\begin{bmatrix} A_{II} & A_{I\Gamma} & 0 & 0 & B_I^T \\ A_{\Gamma I} & A_{\Gamma\Gamma} & 0 & 0 & B_\Gamma^T \\ 0 & 0 & T & 0 & 0 \\ 0 & 0 & 0 & M & 0 \\ B_I & B_\Gamma & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_I \\ u_\Gamma \\ U \\ \omega \\ p \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{15}$$

where

$$\begin{aligned} A_{JK} = \int_{\mathcal{F}_{t_{n+1}}} \rho_f (\partial_t u_h^{n+1}|_{\mathcal{A}})_J \cdot \tilde{u_K} + \int_{\mathcal{F}_{t_{n+1}}} \rho_f((u_h^{n+1} - u_{\mathcal{A}_h}^{n+1}) \cdot \nabla_x u_h^{n+1})_J \cdot \tilde{u_K} + \\ 2\mu \int_{\mathcal{F}_{t_{n+1}}} D(u_h^{n+1})_J : D(\tilde{u_K})\, dx, \quad \text{for } J, K \in \{I, \Gamma\} \end{aligned}$$

$$B_I = -\int_{\mathcal{F}_{t_{n+1}}} p_h^{n+1} \nabla \cdot \tilde{u_I}\, dx \qquad B_\Gamma = -\int_{\mathcal{F}_{t_{n+1}}} p_h^{n+1} \nabla \cdot \tilde{u_\Gamma}\, dx,$$

$$T = m\mathbb{I}, \qquad M = R^n I^n (R^n)^T.$$

Then, the operator

$$P = \begin{bmatrix} \mathbb{I} & 0 & 0 \\ 0 & \tilde{P}_U & \tilde{P}_\omega \\ 0 & \mathbb{I} & 0 \\ 0 & 0 & \mathbb{I} \end{bmatrix},$$

that satisfies the equation

$$(u_I, u_\Gamma, U, \omega)^T = P(u_I, U, \omega)^T$$

is built, as it performs the change of finite element basis from the standard Lagrange basis to the one satisfying $\tilde{u} = \tilde{U} + \tilde{\omega} \times (x - x^{CM})$ on $\partial\mathcal{S}$. In the previous matrix, $\tilde{P}_U$ and $\tilde{P}_\omega$ are the interpolation operators that enable the expression of u_Γ as a function of U and ω. In order to detail them, let us define the spaces

$$V_0 = \{v \in \mathcal{C}^0(\partial\mathcal{S}^h), v\big|_E \in [\mathbb{P}_0(E)]^d \ \forall E \in \partial\mathcal{S}^h\},$$
$$V_1 = \{v \in \mathcal{C}^0(\partial\mathcal{S}^h), v\big|_E \in [\mathbb{P}_1(E)]^d \ \forall E \in \partial\mathcal{S}^h\}.$$

Let us define the polynomial space $[\mathbb{P}_N(\partial\mathcal{S}^h)]^d$, where $N = 2$ is the local polynomial degree of the fluid velocity approximation. The operator $\tilde{P}_U\big| : [\mathbb{P}_N(\partial\mathcal{S}^h)]^d \to V_0$ is defined by

$$\int_{\partial\mathcal{S}^h} u \cdot \varphi \, \mathrm{d}x = \int_{\partial\mathcal{S}^h} U \cdot \varphi \, \mathrm{d}x, \qquad \forall \varphi \in V_0, \tag{16}$$

and the operator $\tilde{P}_\omega\big| : [\mathbb{P}_N(\partial\mathcal{S}^h)]^d \to V_1$ is defined in an analogous manner by

$$\int_{\partial\mathcal{S}^h} u \cdot \varphi \, \mathrm{d}x = \int_{\partial\mathcal{S}^h} \omega \times (x - x^{CM}) \cdot \varphi \, \mathrm{d}x, \qquad \forall \varphi \in V_1. \tag{17}$$

After extending the operator P to $\mathcal{P}$ in order to include the pressure degrees of freedom,

$$\mathcal{P} = \begin{bmatrix} \mathbb{I} & 0 & 0 & 0 \\ 0 & \tilde{P}_U & \tilde{P}_\omega & 0 \\ 0 & \mathbb{I} & 0 & 0 \\ 0 & 0 & \mathbb{I} & 0 \\ 0 & 0 & 0 & \mathbb{I} \end{bmatrix},$$

it is possible to compute the matrix expressing the coupled fluid-rigid body problem by conjugation with $\mathcal{P}$

$$\mathcal{P}^T \begin{bmatrix} A_{II} & A_{I\Gamma} & 0 & 0 & B_I^T \\ A_{\Gamma I} & A_{\Gamma\Gamma} & 0 & 0 & B_\Gamma^T \\ 0 & 0 & T & 0 & 0 \\ 0 & 0 & 0 & M & 0 \\ B_I & B_\Gamma & 0 & 0 & 0 \end{bmatrix} \mathcal{P} = \begin{bmatrix} A_{II} & A_{I\Gamma}\tilde{P}_U & A_{I\Gamma}\tilde{P}_\omega & B_I^T \\ \tilde{P}_U^T A_{\Gamma I} & \tilde{P}_U^T A_{\Gamma\Gamma}\tilde{P}_U + T & \tilde{P}_U^T A_{\Gamma\Gamma}\tilde{P}_\omega & \tilde{P}_U^T B_\Gamma^T \\ \tilde{P}_\omega^T A_{\Gamma I} & \tilde{P}_\omega^T A_{\Gamma\Gamma}\tilde{P}_U & \tilde{P}_\omega^T A_{\Gamma\Gamma}\tilde{P}_\omega + M & \tilde{P}_\omega^T B_\Gamma^T \\ B_I & B_\Gamma\tilde{P}_U & B_\Gamma\tilde{P}_\omega & 0 \end{bmatrix},$$

and the right-hand side of the coupled problem by multiplying it by $\mathcal{P}^T$

$$\mathcal{P}^T \begin{bmatrix} 0 \\ 0 \\ F_e \\ M_e \\ 0 \end{bmatrix}. \tag{18}$$

As the rigid body moves, the geometry of the fluid domain changes. In our framework we supposed the interface between the solid and fluid domains to be conforming, and the evolution of the fluid domain to be described in the ALE frame. In the case of small mesh deformations, equation (11) is solved using piecewise linear continuous finite elements over the triangulated reference fluid domain $\hat{\mathcal{F}}_{t_0}^h$, where t_0 is the reference time, i.e. the latest time instant in which remeshing has been realised. Let us define the finite element spaces

$$\begin{aligned} X_{\bar{\phi}}^h &= \{\phi\big|\phi \in [H^1(\hat{\mathcal{F}}_{t_0}^h)]^d \cap [\mathbb{P}^1(\hat{\mathcal{F}}_{t_0}^h)]^d, \phi = \bar{\phi} \text{ on } \partial\hat{\mathcal{F}}_{t_0}^h\}, \\ X_0^h &= \{\phi\big|\phi \in [H_0^1(\hat{\mathcal{F}}_{t_0}^h)]^d \cap [\mathbb{P}^1(\hat{\mathcal{F}}_{t_0}^h)]^d\}. \end{aligned} \tag{19}$$

The computational domain $\mathcal{F}_{t_{n+1}}^h = \mathcal{A}_h^{t_{n+1}}(\hat{\mathcal{F}}_{t_0}^h)$ is obtained by computing the discrete ALE map $\mathcal{A}_h^{t_{n+1}}(X) = X + \sum_{t=t_0}^{t_{n+1}} \phi_h^t(X_t)$ via the solution of

$$\begin{aligned} &\int_{\mathcal{F}_0^h} (1 + \tau(X)) \nabla\phi_h^{t_{n+1}}(X) : \nabla v \ \mathrm{d}x = 0, && \forall v \in X_0^h, \\ &\phi_h^{t_{n+1}} = \bar{\phi}^{t_{n+1}}, && \text{on } \partial\hat{\mathcal{F}}_{t_0}^h. \end{aligned} \tag{20}$$

Here $\bar{\phi}^{t_{n+1}}(X_{t_0}) = \int_{t_n}^{t_{n+1}} U + \omega \times (X_{t_0} + \phi^{t_n}(X_{t_0}) - X_{t_0}^{CM} - \phi^{t_n}(X_{t_0}^{CM}))\,\mathrm{d}t$ and τ is a piecewise constant coefficient, defined on each element e of the domain's discretization as $\tau\big|_e = (1 - V_{min}/V_{max})/(V_e/V_{max})$, where V_{max}, V_{min} and V_e are the volumes of the largest, smallest and current element of the domain discretization [10]. This discontinuous coefficient takes large values for elements of smaller volume, so that the mesh deformation is mainly supported on larger elements. The time integration of $\bar{\phi}^{t_{n+1}}$ is performed numerically in two steps: first, the contributions coming from the linear velocity are integrated to compute the new center of mass $\bar{\phi}_1^{t_{n+1}}(X_{t_0}^{CM})$

$$\begin{aligned} \theta_{n+1} &= (t_{n+1} - t_n)\omega^n + \theta_n, \quad R^{n+1} = R(\theta_{n+1}), \\ \bar{\phi}_1^{t_{n+1}}(X_{t_0}) &= (t_{n+1} - t_n)U^n; \end{aligned} \tag{21}$$

second, the orientation of the body is computed using the new center of mass

$$\bar{\phi}^{t_{n+1}}(X_{t_0}) \approx R(\theta_{n+1})(X_{t_0} + \phi_1^{t_{n+1}}(X_{t_0}) - \phi_1^{t_{n+1}}(X_{t_0}^{CM})) + \phi_1^{t_{n+1}}(X_{t_0}^{CM}) - X_{t_0}. \tag{22}$$

Rigid body velocities U and ω are approximated at t_n since their values at t_{n+1} are computed during the solution of the fluid problem on $\mathcal{F}_{t_{n+1}}^h$. Few fixed point iterations are performed at each time step to ensure the convergence of the body's position.

2 Computational aspects

Mesh management The ALE formulation of the problem with conforming interface forces us to monitor the quality of the mesh, since following the motion of the rigid bodies deforms the triangulation of the fluid's computational domain. Mesh quality is computed for each two dimensional simplex, using the fair measure

$$q_{2D} = \frac{4\sqrt{3}A}{|e_1|^2 + |e_2|^2 + |e_3|^2}, \tag{23}$$

that depends on the area A of the simplex, and the measures of its edges $|e_i|, i \in \{1, 2, 3\}$ [7]. For a regular simplex $q_{2D} = 1$, and in general $q_{2D} \leq 1$.

Until the minimum of the mesh quality field falls below a certain threshold value (0.4 in the simulations that we will present), the mesh is just deformed by extending the boundary deformation inside the computational domain. Since remeshing is a computationally expensive task, we exploit few techniques to maintain a good mesh quality as long as possible. The first one was already discussed above: when computing the ALE map, a piecewise constant coefficient $\tau(X)$ is prescribed to distribute most of the domain deformation on larger simplexes; in conjunction to this, a mesh size of increasing value, smaller on the boundary of the solid and larger on the boundaries of the fluid domain, can be prescribed. This second technique has a twofold advantage: a better description of the flow in proximity of the body and a larger simplex deformation far from the immersed solid, which delays remeshing further. In our case, the computation of the variable mesh size is based on the Fast Marching Method for the computation of the distance function from the immersed body [14].

When mesh quality falls below the prescribed threshold, a remeshing procedure is initiated. Our implementation relies on *Mmg*[2] and *parMmg* software for mesh adaptation, in sequential and parallel respectively [4]. We choose to keep the same interface and volume discretization for the immersed solid, in order to ease the interpolation of the fluid solution onto the new geometry and keep the volume and surface measures constant across the simulation. Every time the mesh is reconstructed, the current fluid velocity $u(t, x)$, pressure $p(t, x)$ and displacement $x(t, X)$ fields are interpolated onto the new computational domain. Moreover, depending on the order n of the time stepping scheme, also the previous $n - 1$ approximations of the velocity field are interpolated on the computational domains at the appropriate time instants.

Preconditioning There are two types of preconditioners we have explored for the solution of the coupled fluid and rigid body problem: monolithic preconditioners and block preconditioners. In the first group we considered LU preconditioning from the $MUMPS$ library. For larger problems, especially in 3d, the computational effort associated to the use of this solver becomes prohibitive and other solvers, based on block preconditioning and more adapted to the fluid-structure problems at hand, are preferred. In this second group, we considered preconditioners based on algebraic factorisation with different preconditioning strategies on each of the blocks, ranging from algebraic multigrid to LU factorization. These preconditioning strategies are extensively described in [6].

[2] https://www.mmgtools.org

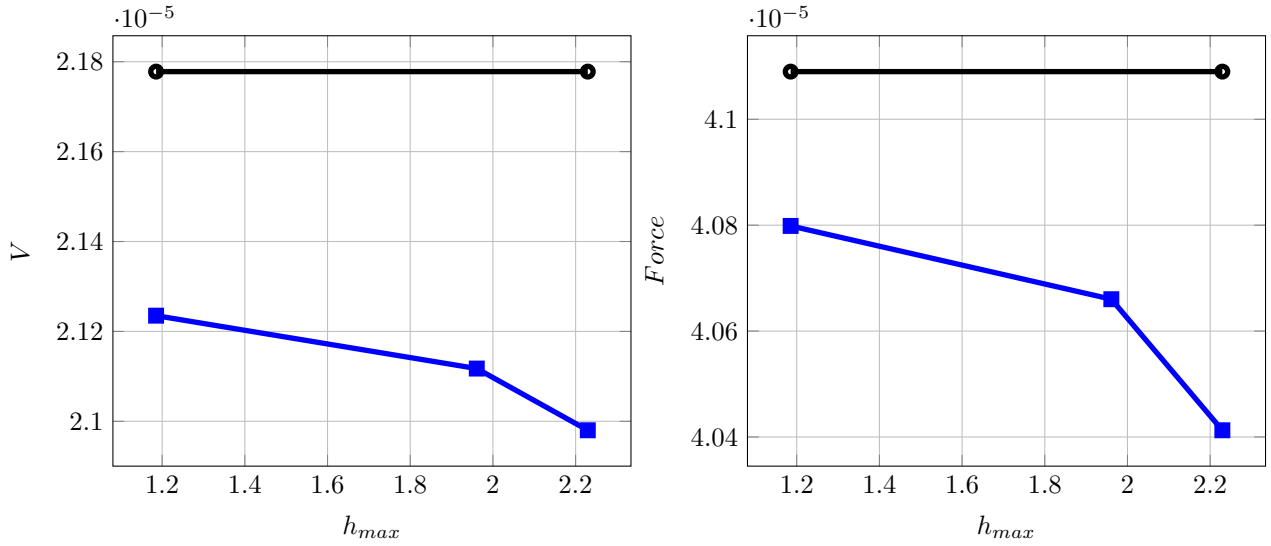


Figure 1: Translational velocity of a sphere subject to gravity (left) and associated drag force (right). The computed values are in blue, the theoretical value is in green. The value being obtained is in line with the prediction of (24) for $\rho_s = 10^{-3}$, $\rho_f = 10^{-6}$, $g = 9.81$, $R = 0.1$, $\mu = 1$, and it converges to the theoretical value as the mesh is refined.

3 Performance and verification on the falling sphere case

In this section, we evaluate the performance of our numerical method on the classical example of the sedimentation of a sphere in Stokes regime, where no inertial effects are present. We evaluate its settling velocity V and fluid forces F, and compare them to the analytical formulas that are valid in this regime. The first one we consider is the Stokes formula $F = 6\pi\mu RV$, relating drag force and linear velocity of a sphere, where R is the radius of the body, V its velocity, μ is the viscosity of the fluid. When specialized to a settling sphere under gravity, the formula can be inverted to compute the settling velocity as

$$V = \frac{2(\rho_s - \rho_f)gR^2}{9\mu}, \tag{24}$$

where ρ_s and ρ_f are the sphere and fluid densities, respectively, and g is the gravity acceleration.
We use these relationships as benchmarks for the fluid-rigid-body coupling solver to assess both the value of the linear velocity and of the fluid forces. Homogeneous Dirichlet boundary conditions are imposed on the external boundaries of the computational domain, together with zero average pressure using a Lagrange multiplier and gravity as the driving force. The results are reported in Figure 1, which shows the values of the settling velocity (left figure) and the drag force (right figure) converging to their analytical values when refining the domain discretization. In Table 1 we report the geometrical characteristics of the mesh we used, namely the number of elements, points and degrees of freedom. An analysis of the preconditioning strategies is also carried out: Table 2 reports the comparison of an LU preconditioning strategy with block preconditioning. The numbers refer to the ratio between the average time for the solution of a single time-step with LU preconditioning, where the preconditioner is constructed at every time-step, and the selected strategy. Table 2, on the left, shows that preconditioning based on algebraic factorisation can provide faster solution than a simple LU preconditioning. However, the performance of the preconditioner depends on the solution strategy for each of the blocks: we can see that choosing LU factorisation for the velocity block ($FS_1^{LU-Pmm-GAMG}$) gives similar results to LU factorisation on the whole matrix. On the other hand, if an algebraic multigrid solver, like GAMG, is chosen for the velocity block ($FS_1^{GAMG-Pmm-jacobi}$), a larger speed-up is obtained. Table 2, on the right, shows that the number of Krylov iterations, necessary to solve the algebraic system, depend strongly on the way that degrees of freedom are split into blocks. In fact, when putting in the same block the degrees of freedom relative to fluid and rigid-body velocities (choice FS_1), while leaving pressure degrees of freedom in a separate block, the number of Krylov iterations is of the order of unity; on the other hand, if the first block contains only the fluid velocities, and rigid-body velocities are put in the second block together with pressure degrees of freedom, the number of iterations necessary to solve the algebraic system grows considerably.

Mesh	N_u	N_p	$N_{elements}$	h_{max}
M_0	163773	7277	48340	2.22947
M_1	250350	11015	78867	1.96095
M_2	1044645	44096	298478	1.18534

Table 1: Mesh characteristics for the falling sphere example.

Mesh	LU	$FS_1^{LU-Pmm-GAMG}$	$FS_1^{GAMG-Pmm-jacobi}$
M_0	1	0.86	0.65
M_1	1	0.61	1.18
M_2	1	1.031	5.2

Mesh	LU	FS_1	FS_2
M_0	-	3	23
M_1	-	7	21
M_2	-	3	12

Table 2: Comparison of the preconditioning strategies on $N = 24$ processors. On the left, numbers represent the ratios of average solution time t_{LU}/t_{prec}. The block preconditioning approach proves more effective than LU preconditioning with more refined meshes. On the right, the number of iterations of the Krylov subspace solver is reported. This table shows that the number of iterations depends strongly on how the degrees of freedom are split into blocks.

4 Test cases

We compare the results of the algorithm presented above with various results from the literature. The testcases we consider concern single rigid bodies moving under gravity: as two-dimensional cases we consider a circle and ellipse subject to gravity and as a three-dimensional case we consider a falling spheroid under gravity.

4.1 Falling ellipse

In this section we consider an example of elliptical body falling under the effect of gravity. This benchmark was presented in [23].

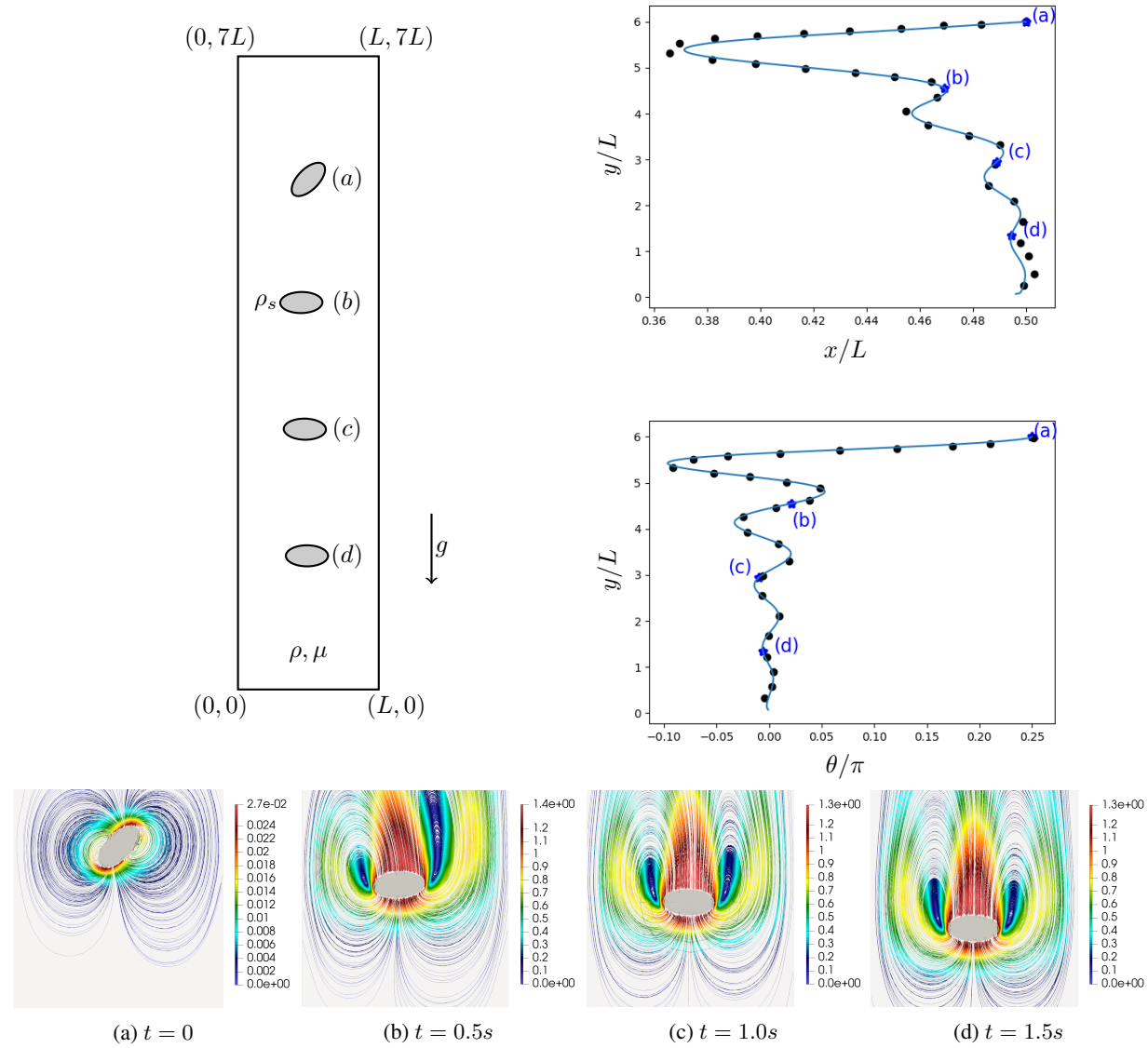
Let us define the characteristic length of the domain as $L = 0.4\,\mathrm{cm}$. We consider an ellipse of semi-axes $a = L/4$ and $b = L/8$ evolving in the rectangular cavity $[0, L] \times [0, 7L]$ filled with fluid. At time $t = 0\,\mathrm{s}$, the center of the ellipse is located at $(0.5L, 6L)$ and the major semi-axis a forms an angle of $\pi/4$ with the horizontal direction. The fluid density is $\rho = 1\,\mathrm{g/cm^3}$, the kinematic viscosity is $\mu = 0.01\,\mathrm{cm^2/s}$ and the ellipse density is $\rho_s = 1.1\,\mathrm{g/cm^3}$. The gravity acceleration is $g = 980\,\mathrm{cm/s^2}$. Homogeneous Dirichlet boundary conditions are applied on the walls and the bottom of the cavity, while homogeneous Neumann conditions are applied on the top. Our results are obtained with a time-step $\Delta t = 0.001\,\mathrm{s}$, , a second order BDF scheme and with mesh size $h \approx 0.01$. The resulting mesh is composed of about 18600 points and 36500 elements.

The center of mass of the ellipse describes an oscillatory trajectory as the solid settles on the bottom of the cavity; during the fall, the orientation of the body changes as well in an oscillatory manner. Figures 2 collect the initial geometry of the problem (on the left) and the comparison of our results regarding the trajectory and orientation of the elliptical body with [23]. Our results (blue continuous line) are in good agreement with the reference (black dots). Both the trajectory of the center of mass and the angular dynamics show an accurate approximation of the reference results.

4.2 Falling cylinder

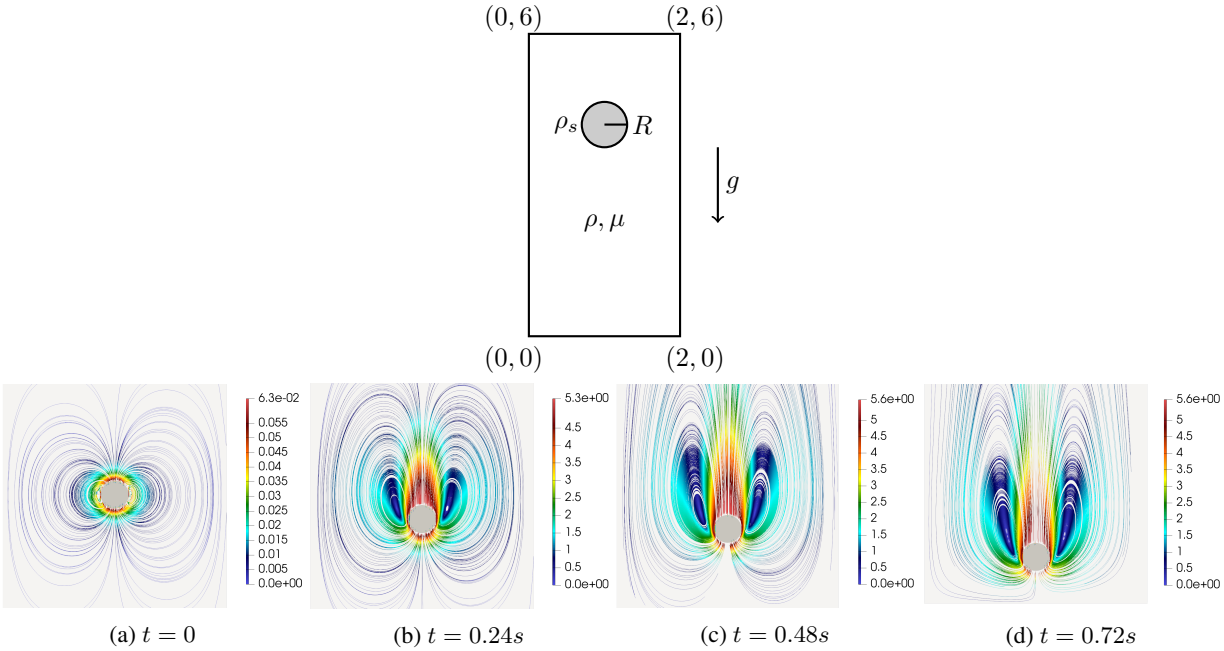
In this section we consider a two dimensional cylinder falling under gravity in a confined environment and compare our results with [22]. The computational domain is a rectangle of $[0, 2] \times [0, 6]\,\mathrm{cm}$ filled with fluid. The radius of the body is $R = 0.125\,\mathrm{cm}$ and, at $t = 0\,\mathrm{s}$, the center of mass of the cylinder is located at $(1, 4)$. Figure 3 presents a schematic representation of the testcase. The fluid density is $\rho = 1\,\mathrm{g/cm^3}$, the fluid viscosity is $\mu = 0.1\,\mathrm{g/(cm \cdot s)}$ and the solid density is $\rho_s = 1.25\,\mathrm{g/cm^3}$. The gravity acceleration is $g = 980\,\mathrm{cm/s^2}$ also in this case. As before, homogeneous Dirichlet boundary conditions are applied on the bottom and lateral walls, while homogeneous Neumann conditions are imposed on the top of the cavity. Our results are obtained with a time-step of $\Delta t = 0.001\,\mathrm{s}$, a second order BDF scheme and a mesh size $h \approx 0.06$. The initial mesh is composed of approximately 5700 nodes and 11200 elements.

We compare our results to [22], which display the time evolution of four quantities: the coordinate y^{CM} of the center of mass of the body, the y component of the body velocity U, the Reynolds number and the kinetic energy of the solid. These two aggregated quantities are computed, respectively, as $Re = 2R\rho_s\sqrt{U_x^2 + U_y^2}/\mu$ and $E_k = 0.5\rho_s\pi R^2(U_x^2 + U_y^2)$. Figure 4 reports the comparison between our results (blue continuous line) and [22] (black



(a) $t = 0$ (b) $t = 0.5s$ (c) $t = 1.0s$ (d) $t = 1.5s$

Figure 2: On the left, schematic representation of different configurations of the falling ellipse, starting in the tilted position. On the left, trajectory and orientation of the elliptical particle. The continuous line corresponds to our results, while the black dots correspond to results taken from [23]. The positions and orientations corresponding to the starred values are represented in the left figure.



(a) $t = 0$ (b) $t = 0.24s$ (c) $t = 0.48s$ (d) $t = 0.72s$

Figure 3: Schematic representation of the falling cylinder benchmark and streamlines at four different time instants.

dots), showing a good agreement between the two. A slight overestimation of the falling velocity U_y is the reason why the matching is not perfect. This small difference is amplified in the kinetic energy plot, because $E_k \propto U_y^2$.

4.3 Three-dimensional spheroid

In this section we consider the simulation of a falling three-dimensional spheroid in a cylindrical domain. Following [15], three non-dimensional parameters are identified and used to describe the flow and particle properties: the aspect ratio is defined by $\chi = d/a$, where a is the axis length of the spheroid in the local z_{loc} direction and d is its diameter in the local $x_{loc} - y_{loc}$ plane; the density ratio is denoted by $\tilde{\rho} = \rho_f/\rho_s$ between the fluid and the particle; the Galileo number is given by $Ga = U_g d/\nu$, where $U_g = \sqrt{(\tilde{\rho} - 1)g/d^2(\pi/\chi)(d^3/6)}$ is a gravitational velocity scale defined for heavy particles and ν is the kinematic viscosity of the fluid. The gravitational velocity was taken to be $U_g = 1$, so that the kinematic viscosity $\nu = 1/Ga$. In Figure 5 the parametrization is shown.

The case considered here corresponds to the triplet $(\chi, Ga, \tilde{\rho}) = (1.5, 150, 2.14)$, which produces a vertical oscillating regime. However, differently from [15], our problem was cast in the global reference frame (and not in the spheroid's local reference frame), which led to a different settling path and behaviour as the spheroid approached the walls of the cylindrical domain. The radius of the computational cylinder in which the spheroid falls down was fixed to be $R_c = 5.34d$ and its height was taken to be $h = 75d$ in order to avoid contact with the walls and observe its settling behavior during a time window of $40\,\mathrm{s}$. In figure 6 we plot the settling velocity and the z coordinate of the spheroid's center of mass. Despite the different approaches, the results are similar to those obtained in [15].

Conclusions

In this paper we presented a mathematical and numerical formulation for the solution of Navier-Stokes equations in moving domain with immersed, moving rigid bodies. Our description is based on the ALE formulation of the fluid equations, coupled with the Newton-Euler equations for rigid body motion. The computational aspects involved in the numerical solution of the problem are discussed, especially mesh adaptation and the preconditioning strategies.

Our numerical results are benchmarked with results from the literature, and a good agreement is found. The code is open source [17] and the configuration files of the benchmarks are available as well [1]. The testcases that were considered validate our fluid-solid interaction solver for moving rigid bodies. This validation is a first step towards the construction of a computational tool simulating swimming and the motion of deformable bodies in fluids. In an

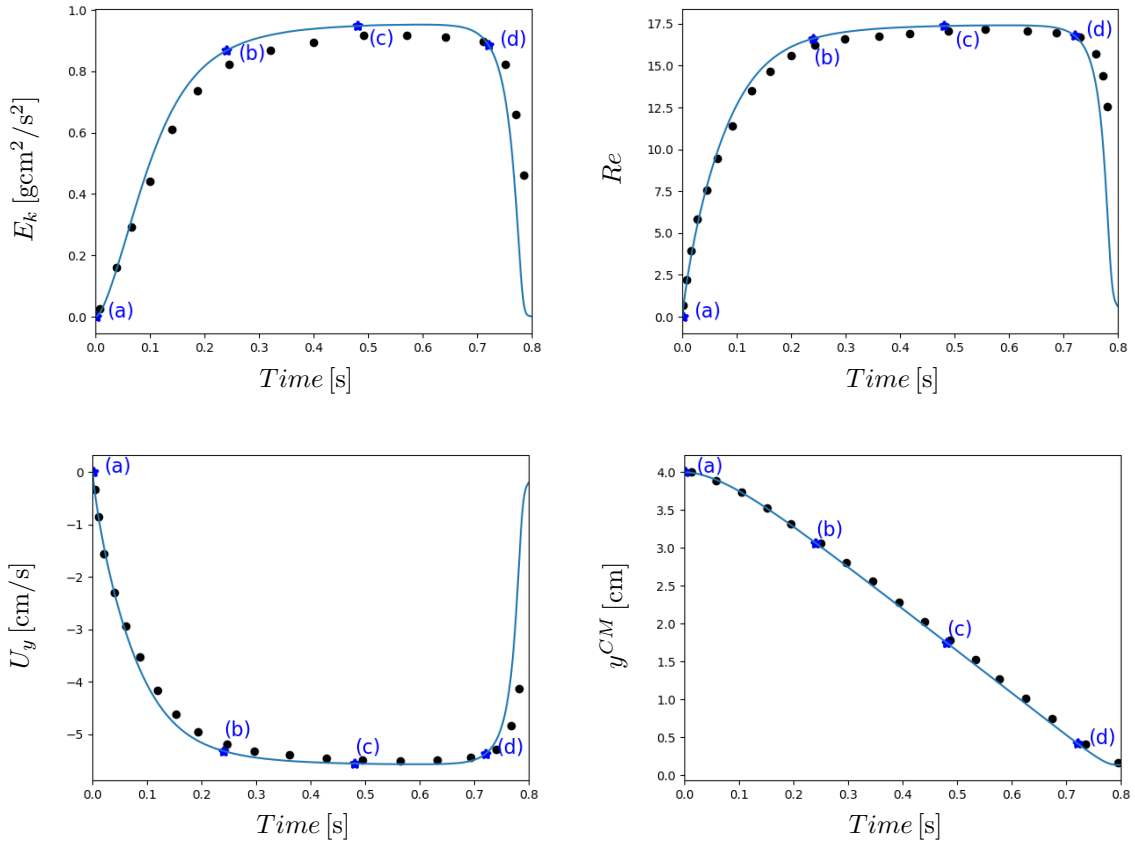


Figure 4: Time evolution of different quantities relative to the confined cylinder example. From the top left: kinetic linear energy, Reynolds number, vertical component of the linear velocity and y coordinate of the cylinder's center of mass. The blue continuous line corresponds to our results, while the black dots correspond to results from [22]. The starred values correspond to the time instants $t = 0, 0.24, 0.48, 0.72s$.

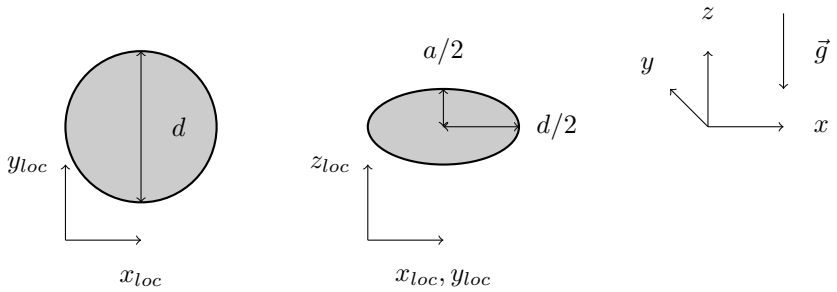


Figure 5: On the left, parametrization of the $3D$ spheroid in its local reference frame. On the right, global reference frame and gravity vector.

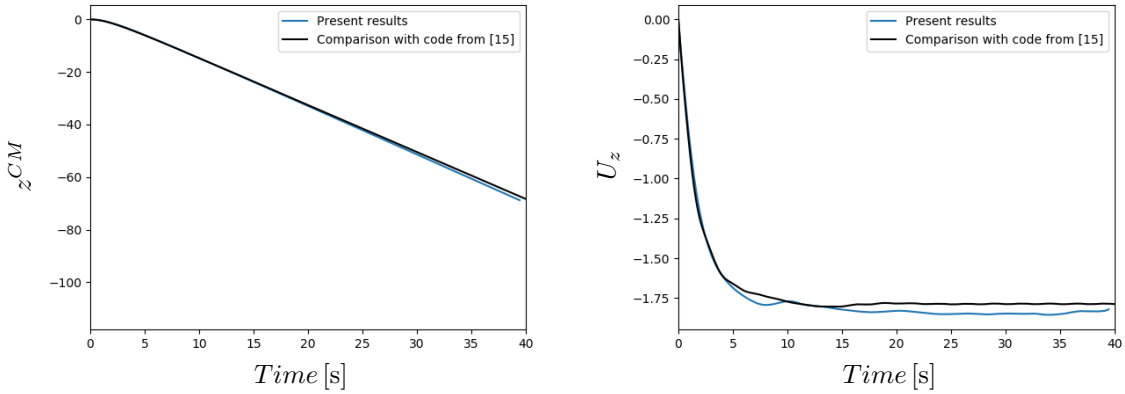


Figure 6: Trajectory in the z direction of the center of mass and z component of the linear velocity for the 3d freely falling spheroid testcase.

upcoming paper we will turn to the simulation of swimming bodies with prescribed deformation strategies and to the coupling with elasticity models.

Acknowledgements

The authors acknowledge the financial support of Labex Irmia, ITI Irmia++ and the French Agence Nationale de la Recherche (ANR), under grant ANR-21-CE45-0013 (project NEMO). The authors also thank Yannick Hoarau, Thibaut Metivet and Jan Dušek for the fruitful discussions and the data they provided.

Reproducibility

The benchmarks described in this paper are publicly available in the Github repository [1] and can be reproduced by following the instructions.

A Implementation details

The results presented in this paper were obtained via the Finite Element Embedded Library in `C++` called `Feel++` [17]. `Feel++` allows to use a very wide range of advanced methods for numerical simulations and provides a mathematical kernel for solving partial differential equations using arbitrary order Galerkin methods, together with a set of mono and multi-physics toolboxes. A collection of Python modules wrapping the library and its multi-physics toolboxes is also available. In this section we will briefly present the classes that compose the fluid mechanics toolbox, the configuration and results files.

A.1 The fluid mechanics toolbox

The fluid mechanics toolbox is an application that solves the Stokes and Navier-Stokes equations in moving domain using the Arbitrary-Lagrangian-Eulerian formalism. In its present state [17], it encompasses the possibility to simulate fluid flows around moving objects or inside domains with deforming boundaries. The `FluidMechanics` base class extends four fundamental classes, namely `ModelBase`, which handles the execution of the toolbox at its lowest level (creating folders for data, and the interface for parallel computations); `ModelAlgebraic` and `ModelNumerical`, which contain the algebraic data structures used to solve the discrete problem, to accomplish the post-processing and time-stepping; `ModelPhysics`, which provides the data structures that specify the physics of the problem (e.g. material properties, turbulence models). The `FluidMechanics` class fills the general interface proposed by the previous fundamental classes with the variational problem coming from the discretisation of the Navier-Stokes equations. An UML diagram is provided in Figure 7 to appreciate the dependencies of the class. Additional classes are interfaced with `FluidMechanics` to enable the simulation of fluids in moving domain (`MeshALE`), the simulation of rigid bodies

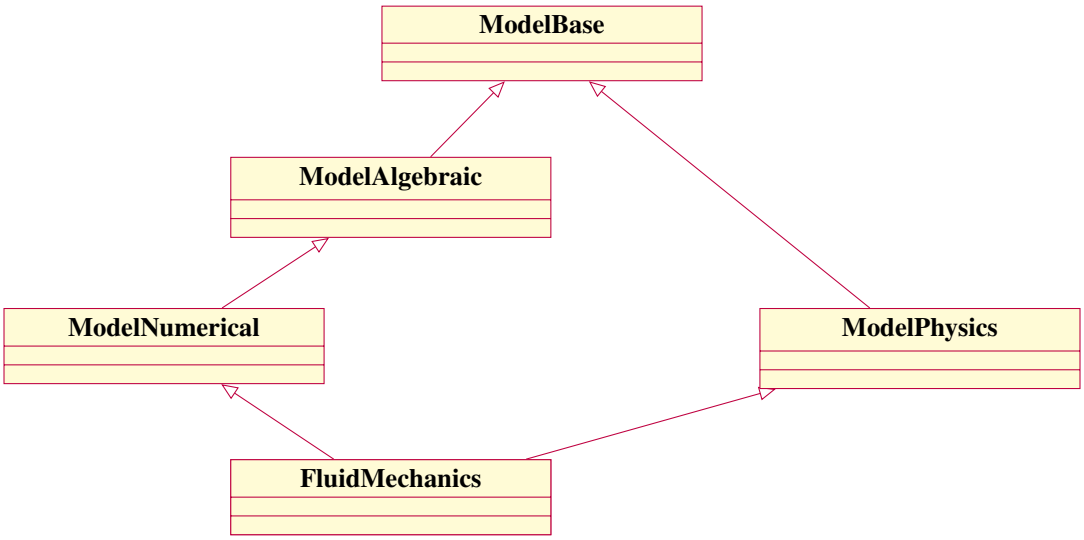


Figure 7: UML diagram of the `FluidMechanics` class.

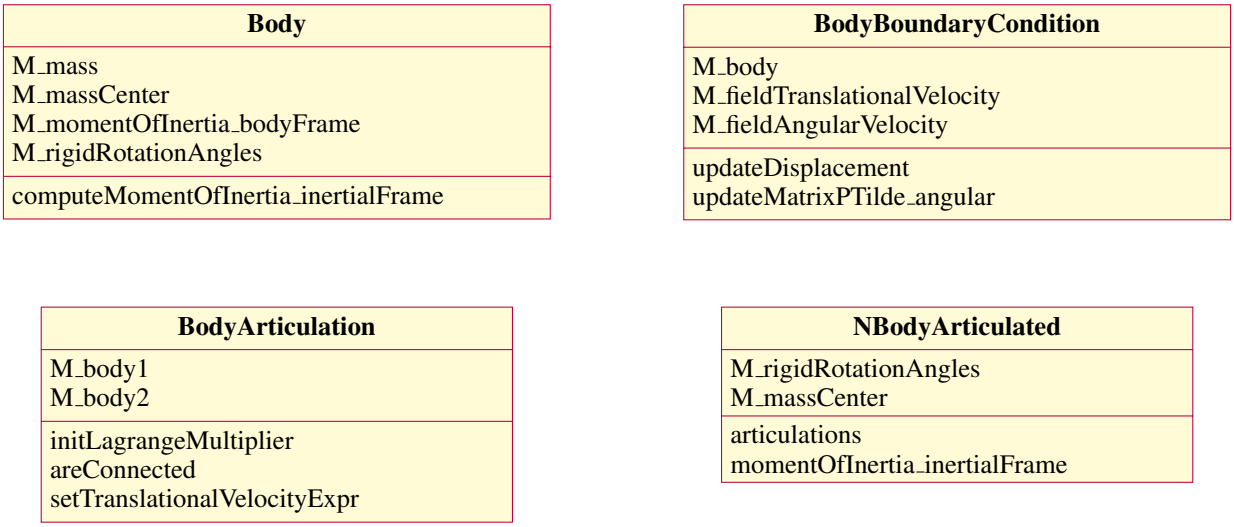


Figure 8: Synthetic UML diagrams for the classes that handle the simulation of rigid body motion inside `FluidMechanics`. The `Body` class collects the variables describing the position of the body and its dynamical properties, as well as the routines to update them. `BodyBoundaryCondition` contains the variables that describe the rigid and deformation velocity of the body, while `NBodyArticulated` allows to describe a body which is made of several `Body` components, connected and moving relatively to each other as prescribed in `BodyArticulation`.

(`Body`, `BodyBoundaryCondition`), the simulation of articulated bodies (`BodyArticulation`, `NBodyArticulated`). Synthetic information about the relevant methods and attributes of these classes are provided in Figure 8.

A.2 Launching the tests

The numerical results in this paper can be reproduced by following the instructions contained in the repository. The fluid toolbox is initialised and executed, and at each time step the mesh quality measure (23) is computed by the function `etaQ`, and if its minimum falls below a predefined threshold, the computational domain is rebuilt. In order to have a conforming interface between the solid and the fluid, the user prescribes the surface and volume markers to be kept by the remeshing algorithm.

A.3 Configuration and results files

Fluid mechanics simulations are parametrized by three configuration files: a geometry file, describing the computational domain and characterised by the `.geo` extension; a `.json` file, prescribing the quantitative information about the problem, namely the fluid model, the material properties and the boundary conditions; a `.ini` file, that contains the location of the other configuration files, parametrizes the algebraic solver and the time-stepping algorithm. In order to differentiate between Dirichlet boundary conditions at fluid-particle interfaces and on external boundaries, the json configuration file has a specific section `body` to treat the first case. In this section, linear or angular velocities can also be prescribed at the surface of the object, but the simplest case is reported in Listing 1, where only the boundary condition $u = U + \omega \times (x - x^{CM})$ is prescribed on $\partial\mathcal{F}_t \cap \partial\mathcal{S}$.

Listing 1: Body section in the .json configuration file

```
"BoundaryConditions":
{
        "fluid":
        {
                ...,
                "body":
                {
                        "bodyName":
                        {
                                "markers":"SurfaceMarkers",
                                "materials":{"names":"RigidBodyMaterial"}

                        }
                }
        }
}
```

The data that describe the motion of the rigid body (the coordinates of the center of mass, the orientation angles, the moment of inertia, the net fluid forces and torques) are collected in a csv file and are preceded by the name of the body itself.

References

[1] *Supplementary material: https://github.com/feelpp/article.fluid_rigid_body_ale*.

[2] S. ALLENDE, C. HENRY, AND J. BEC, *Stretching and buckling of small elastic fibers in turbulence*, Physical review letters, 121 (2018), p. 154501.

[3] F. AUGUSTE AND J. MAGNAUDET, *Path oscillations and enhanced drag of light rising spheres*, Journal of Fluid Mechanics, 841 (2018), pp. 228–266.

[4] C. DAPOGNY, C. DOBRZYNSKI, AND P. FREY, *Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems*, Journal of Computational Physics, 262 (2014), pp. 358–378.

[5] J. DONEA, A. HUERTA, J.-P. PONTHOT, AND A. RODRÍGUEZ-FERRAN, *Arbitrary lagrangian-eulerian methods*, Encyclopedia of computational mechanics, (2004).

[6] H. ELMAN, D. SILVESTER, AND A. WATHEN, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, Oxford University Press, June 2014.

[7] D. A. FIELD, *Qualitative measures for initial meshes*, International Journal for Numerical Methods in Engineering, 47 (2000), pp. 887–906.

[8] L. FORMAGGIA AND A. QUARTERONI, eds., *Cardiovascular mathematics: modeling and simulation of the circulatory system*, no. 1 in MS & A : modeling, simulation & applications, Springer, Milano, 2009.

[9] K. KAFUI, C. THORNTON, AND M. ADAMS, *Discrete particle-continuum fluid modelling of gas–solid fluidised beds*, Chemical Engineering Science, 57 (2002), pp. 2395–2410.

[10] H. KANCHI AND A. MASUD, *A 3D adaptive mesh moving scheme*, International Journal for Numerical Methods in Fluids, 54 (2007), pp. 923–944.

[11] Y. KIM AND C. S. PESKIN, *A penalty immersed boundary method for a rigid body in fluid*, Physics of Fluids, 28 (2016), p. 033603.

[12] B. MAURY, *Direct Simulations of 2D Fluid-Particle Flows in Biperiodic Domains*, Journal of Computational Physics, 156 (1999), pp. 325–351.

[13] B. MAURY AND R. GLOWINSKI, *Fluid-particle flow: a symmetric formulation*, Comptes Rendus de l'Académie des Sciences - Series I - Mathematics, 324 (1997), pp. 1079–1084.

[14] T. METIVET, V. CHABANNES, M. ISMAIL, AND C. PRUD'HOMME, *High-Order Finite-Element Framework for the Efficient Simulation of Multifluid Flows*, Mathematics, 6 (2018), p. 203.

[15] M. MORICHE, M. UHLMANN, AND J. DUŠEK, *A single oblate spheroid settling in unbounded ambient fluid: a benchmark for simulations in steady and unsteady wake regimes*, 2020.

[16] C. S. PESKIN, *The immersed boundary method*, Acta numerica, 11 (2002), pp. 479–517.

[17] C. PRUD'HOMME, V. CHABANNES, T. METIVET, R. HILD, TROPHIME, A. SAMAKE, T. SAIGRE, P. RICKA, AND L. BERTI, *feelpp/feelpp; doi: 10.5281/zenodo.5718297*.

[18] A. RAKHIMZHANOVA, C. THORNTON, Y. AMANBEK, AND Y. ZHAO, *Numerical simulations of sand production in oil wells using the CFD-DEM-IBM approach*, Journal of Petroleum Science and Engineering, 208 (2022), p. 109529.

[19] A. G. TOMBOULIDES AND S. A. ORSZAG, *Numerical investigation of transitional and weak turbulent flow past a sphere*, Journal of Fluid Mechanics, 416 (2000), pp. 45–73.

[20] M. UHLMANN, *An immersed boundary method with direct forcing for the simulation of particulate flows*, Journal of Computational Physics, 209 (2005), pp. 448–476.

[21] M. UHLMANN AND J. DUŠEK, *The motion of a single heavy sphere in ambient fluid: A benchmark for interface-resolved particulate flow simulations with significant relative velocities*, International Journal of Multiphase Flow, 59 (2014), pp. 221–243.

[22] D. WAN AND S. TUREK, *Direct numerical simulation of particulate flow via multigrid FEM techniques and the fictitious boundary method*, International Journal for Numerical Methods in Fluids, 51 (2006), pp. 531–566.

[23] Z. XIA, K. W. CONNINGTON, S. RAPAKA, P. YUE, J. J. FENG, AND S. CHEN, *Flow patterns in the sedimentation of an elliptical particle*, Journal of Fluid Mechanics, 625 (2009), pp. 249–272.