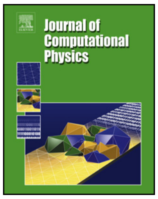
Journal of Computational Physics 374 (2018) 625–653

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp



Non-body-fitted fluid–structure interaction: Divergence-conforming B-splines, fully-implicit dynamics, and variational formulation

Hugo Casquero [a,*], Yongjie Jessica Zhang [a], Carles Bona-Casas [b], Lisandro Dalcin [c], Hector Gomez [d]

[a] *Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, United States*
[b] *Departament de Física & IAC^3, Universitat de les Illes Balears, Palma de Mallorca, 07122, Spain*
[c] *Extreme Computing Research Center, King Abdullah University of Science and Technology (KAUST), Thuwal, 23955, Saudi Arabia*
[d] *School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, United States*

ARTICLE INFO

Article history:
Received 27 November 2017
Received in revised form 2 July 2018
Accepted 12 July 2018
Available online 19 July 2018

Keywords:
Fluid–structure interaction
Immersed boundary method
Incompressibility
Isogeometric analysis
Divergence-conforming B-splines
Divergence-conforming immersed boundary method

ABSTRACT

Immersed boundary (IB) methods deal with incompressible visco-elastic solids interacting with incompressible viscous fluids. A long-standing issue of IB methods is the challenge of accurately imposing the incompressibility constraint at the discrete level. We present the divergence-conforming immersed boundary (DCIB) method to tackle this issue. The DCIB method leads to completely negligible incompressibility errors at the Eulerian level and various orders of magnitude of increased accuracy at the Lagrangian level compared to other IB methods. Furthermore, second-order convergence of the incompressibility error at the Lagrangian level is obtained as the discretization is refined. In the DCIB method, the Eulerian velocity–pressure pair is discretized using divergence-conforming B-splines, leading to *inf–sup* stable and *pointwise* divergence-free Eulerian solutions. The Lagrangian displacement is discretized using non-uniform rational B-splines, which enables to robustly handle large mesh distortions. The data transfer needed between the Eulerian and Lagrangian descriptions is performed at the quadrature level using the same spline basis functions that define the computational meshes. This conduces to a fully variational formulation, sharp treatment of the fluid–solid interface, and a 0.5 increase in the convergence rate of the Eulerian velocity and the Lagrangian displacement measured in L^2 norm in comparison with using discretized Dirac delta functions for the data transfer. By combining the generalized-α method and a block-iterative solution strategy, the DCIB method results in a fully-implicit discretization, which enables to take larger time steps. Various two- and three-dimensional problems are solved to show all the aforementioned properties of the DCIB method along with mesh-independence studies, verification of the numerical method by comparison with the literature, and measurement of convergence rates.

---

[*] Corresponding author.
E-mail address: hugocp@andrew.cmu.edu (H. Casquero).

https://doi.org/10.1016/j.jcp.2018.07.020
0021-9991/

1. Introduction

In 1972, Peskin proposed the immersed boundary (IB) method to tackle fluid–structure interaction (FSI) problems [1–3]. This contribution was a significant breakthrough in the field of FSI since it opened the way for simulating problems that involve large translational and rotational motions of several deformable solids in an automatic way. The original IB method may be seen as both a mathematical model and a numerical method, in which incompressible visco-elastic solids defined on co-dimension one manifolds interact with an incompressible viscous fluid. The mathematical model manages to link Eulerian descriptions of the velocity and the pressure defined in the fluid and solid domains with a Lagrangian description of the displacement defined in the solid domain, that is, it enables to treat the viscous and elastic parts of the Cauchy stress tensor in their natural descriptions. In the original IB numerical method, finite differences were used to approximate the Navier–Stokes equations, the solids were represented by a fiber network, and the data transfer needed between the Eulerian and Lagrangian descriptions was performed using discretized Dirac delta functions. Regarding the time discretization, the position of the solids was updated in an explicit manner. In [4], the mathematical model of the IB method was generalized to deal with co-dimension zero viscous hyperelastic solids. In [5,4,6,7], finite elements were used to discretize both the Navier–Stokes equations and the solid. Regarding the data transfer between descriptions, discretized Dirac delta functions were still used in [6,7] while the basis functions of the finite element meshes were used instead in [5,4], thus leading to a fully-variational formulation. The generalization of the IB method to work with rigid solids was done in [8,9], to work with variable densities and viscosities in [10,11], and to work with two-fluid flows in [12,13]. In the last decades, diverse numerical methods have been developed to discretize the mathematical model proposed by the IB method, such as a hybrid finite difference/finite element discretization [14], a NURBS-based discretization [15], a discretization based on T-splines [16], finite volume discretization for the Navier–Stokes equations [17], and lattice Boltzmann discretization for the Navier–Stokes equations [18,19]. These numerical methods have been applied to a variety of problems, such as heart valve analysis and design [20–23], cell-scale blood flow [24–26], aquatic animal locomotion [27,28], tissue cryofreezing [29], capsule dynamics [30], vesicle dynamics [31], particle laden flows [32], and floating structures [33].

In [34], theory and numerical examples show how weakly divergence-free discretizations lead to extremely poor mass conservation in incompressible flow problems with large forcings verifying that the irrotational part of the forcing is large relative to the divergence-free part. Moreover, the exact solution of the Eulerian pressure is usually discontinuous at the fluid–solid interface, which leads to poor approximation properties of the discrete pressure spaces used in immersed FSI methods. As a result, immersed FSI methods, whether they follow Peskin's idea or not (see [35–39]), often have profound difficulties to accurately impose the incompressibility constraint at both Eulerian and Lagrangian levels [40,41,6,34,38,25,42]. This issue is extremely important since large errors in the incompressibility constraint are able to even alter the qualitative behavior of numerical solutions in challenging FSI applications, such as heart valves [40,38] and cell-scale blood flow [25]. A number of work-arounds have been proposed in the literature to try to deal with this issue. In [38], following what was proposed for incompressible flow problems in [34], stabilized formulations for the Navier–Stokes equations are locally modified near the fluid–solid interface to add heavy grad–div stabilization. This heavy grad–div stabilization near the fluid–solid interface is able to impose the incompressibility constraint accurately at the Eulerian level. However, as the authors mentioned in [34,38], the modified stabilization will lead to locking and other instabilities if the scaling parameters are large enough and there is no rule to know *a priori* when this will happen. In addition, the modified stabilization increases the condition number of the final system of equations and it becomes a challenge to find a scalable solver in order to work with highly-refined three-dimensional meshes [38,42,25]. In [43], an *ad hoc* volume correction strategy at the Lagrangian level, to be carried out every ten time steps, was proposed to reduce the spurious volume change of co-dimension zero solids in immersed FSI methods. Although the authors acknowledge that the introduction of this correction strategy, which is not derived from physical laws, may negatively impact the accuracy of numerical solutions. Interesting options to capture the pressure discontinuity at the interface are to combine the fictitious domain method with adaptive meshing following the interface as done in [37] and to use an extended finite element discretization as done in [44,42]. However, the automatic application of these two methods to complex three-dimensional problems remains an open problem. Recently in [45], a novel IB method, based on finite differences and discretized Dirac delta functions, was proposed. This method defines velocity-interpolation and force-spreading schemes in such a way that the interpolated velocity field in which the structure moves is at least $\mathcal{C}^1$-continuous and satisfies a continuous divergence-free condition. In [45], various benchmark problems involving co-dimension one solids are solved and the new IB method leads to various orders of magnitude of increased accuracy in volume conservation at the Lagrangian level in comparison with other IB methods [40,41]. However, as the authors explained in [45], their method is restricted to periodic domains up to now, which limits its potential application to real-word FSI problems.

In [34], incompressible flow benchmark problems involving large irrotational forcings were solved using several mixed finite elements that are *inf–sup* stable. Scott–Vogelius elements, which lead to *pointwise* divergence-free velocity solutions, completely outperformed Taylor–Hood elements, P2P0 elements, and P1Bubble-P1 elements regarding mass conservation, thus showing the great advantages of pointwise divergence-free discretizations for this kind of problems. To the best of our knowledge, the only work that has tried pointwise divergence-free Eulerian discretization in the context of immersed FSI methods is the recent paper [46]. In [46], divergence-conforming B-splines [47–50] were applied to the non-boundary-fitted FSI method developed in [38,39]. The method used in [38,46] defines a fluid subproblem and a Kirchhoff–Love shell subproblem with no-tailored discretizations. The no-penetration and no-slip conditions at the interface are imposed through a generalization of Nitsche's method, which involves adding penalty terms and Lagrangian multipliers to the formulation. Divergence-conforming B-splines take advantage of the higher inter-element continuity of B-splines [51,52] to construct velocity–pressure pairs that are *inf–sup* stable, *pointwise* divergence-free, and H^1 conforming. The pointwise divergence-free velocity solutions are obtained by posing the variational form on spaces forming a discrete de Rahm complex [53,47] in which the divergence operator maps velocity solutions into the pressure space. In the last decade, the use of spline functions, such as B-splines, non-uniform rational B-splines (NURBS), analysis-suitable T-splines (ASTS), and hierarchical B-splines, has become widespread in computational mechanics thanks to isogeometric analysis (IGA) [51,54]. In the field of immersed methods for FSI, IGA has already been used to perform NURBS-based and ASTS-based generalizations of the IB method [15,16], develop the immersogeometric method [38,55,39], couple shells with Stokes flows using the boundary integral method [56,57], solve air-blast problems [58], develop a fictitious domain approach [59], and a stabilized cut-cell immersed framework [60].

In this paper, we propose an IB numerical method that uses divergence-conforming B-splines for the discretization of the Eulerian velocity and the Eulerian pressure together with NURBS for the discretization of the Lagrangian displacement. Our method benefits from the higher inter-element continuity of splines, which leads to higher accuracy per degree of freedom in fluid mechanics [61] and increased robustness when it comes to handle large deformations in solid mechanics [62]. The data transfer between the Eulerian and Lagrangian descriptions is performed using the spline basis functions that define the meshes, which leads to a fully variational formulation. With respect to the time discretization, the generalized-α method along with a block iterative solution strategy leads to a fully-implicit method that enables to take larger time steps than semi-implicit algorithms [63,64]. We also develop a scalable parallel implementation of the proposed method. Here, we focus on co-dimension zero solids since this is the case where imposing the incompressibility constraint is more challenging. This is due to the fact that even with negligible errors in the Eulerian velocity, the volume of the solids may still vary along the simulation due to the discretization error introduced when the Lagrangian displacement is computed from the Eulerian velocity. Nevertheless, this framework can be used to deal with co-dimension one solids, which is an appealing direction of future research to the authors. Since the main aim of this method is to significantly improve the accuracy with which the incompressibility constraint is satisfied in the context of immersed boundary methods, the method presented in this manuscript will be called from now on the divergence-conforming immersed boundary (DCIB) method.

The outline of this paper is as follows. In Section 2 we define basic notation that will be used throughout the paper. Section 3 presents the mathematical model of the IB method, expressed in both strong form and weak form. Section 4 describes the discretization process proposed by the authors. Section 4.1 starts explaining how to construct B-splines and NURBS in different dimensions. After that, the spatial discretization is performed. Section 4.2 sorts out the time discretization to reach a fully-discrete formulation and Section 4.3 explains the steps followed to solve the discrete system. Section 5 displays several numerical examples in two- and three-dimensional layouts. The first example is a two-dimensional problem consisting of a soft disk in a lid-driven cavity flow. This example is a widespread benchmark problem to evaluate how accurately immersed FSI methods are able to impose the incompressibility constraint at the discrete level, the DCIB method vastly outperforms the immersed FSI methods that have solved this benchmark before. Additionally, we use this example to verify the DCIB method by comparing our results with those obtained with a fully Eulerian finite difference method for FSI problems [65]. The second example is a two-dimensional benchmark problem to study the accuracy with which immersed FSI methods satisfy energy conservation at the discrete level. It consists of an oscillating disk surrounded by fluid. The third example is a hollow disk in a shear flow in a two-dimensional layout. We use this example to compute the convergence rates of our numerical method. In the fourth example, we consider a three-dimensional problem consisting of eight hollow spheres in a gravity-driven flow. Taking advantage of our scalable parallel implementation, we perform a mesh-independence study for this problem. The last example tests the robustness of our numerical method and it consists of a red blood cell passing through a constriction, which mimics what happens in the human spleen. Finally, in Sections 6 and 7, some concluding remarks and directions of future research are drawn, respectively.

2. Notation and kinematics

Let $d = \{2, 3\}$ and $(0, T)$ be the number of spatial dimensions and the time interval of interest, respectively. Let $\Omega_1^t \subset \mathbb{R}^d$ and $\Omega_2^t \subset \mathbb{R}^d$ be two open sets that represent the time-dependent domains occupied by a viscous incompressible fluid and a viscous hyperelastic incompressible solid, respectively. $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ are the outward unit normal vectors to Ω_1^t and Ω_2^t, respectively. Γ_1^t and Γ_2^t are the boundaries of Ω_1^t and Ω_2^t, respectively. These two regions meet at the fluid–solid interface $\Gamma_I^t = \overline{\Omega_1^t} \cap \overline{\Omega_2^t}$. For simplicity, we assume that the solid is completely embedded in the fluid as this holds true for all the examples presented in this paper. Therefore, the fluid–solid interface coincides with the boundary of the solid, that is, $\Gamma_I^t = \Gamma_2^t$. Let $\Omega \subset \mathbb{R}^d$ be an open set that represents the domain occupied by both the fluid and the solid ($\overline{\Omega} = \overline{\Omega_1^t \cup \Omega_2^t}$). Note that Ω is considered to be time independent in this exposition since that is always the case in the simulations of Section 5. The boundary of Ω is denoted by Γ, which coincides with $\Gamma_1^t = \Gamma_1^0$ since the solid is fully embedded in the fluid.

The three main variables of our mathematical model are the Eulerian velocity $\boldsymbol{v} : \Omega \times (0, T) \mapsto \mathbb{R}^d$, the Eulerian pressure $p : \Omega \times (0, T) \mapsto \mathbb{R}$, and the Lagrangian displacement $\boldsymbol{u} : \Omega_2^0 \times (0, T) \mapsto \mathbb{R}^d$, where Ω_2^0 is the material domain of Ω_2^t. Let $\boldsymbol{X} \in \Omega_2^0$, $\boldsymbol{x} \in \Omega$, and $\boldsymbol{\varphi} : \Omega_2^0 \times (0, T) \mapsto \Omega_2^t$ be a material particle, a spatial position, and the deformation mapping, respectively. The deformation mapping verifies that $\boldsymbol{\varphi}(\boldsymbol{X}, t) = \boldsymbol{X} + \boldsymbol{u}(\boldsymbol{X}, t)$.

3. Immersed formulation of the governing equations

In this section, the immersed FSI formulation is presented. At the continuous level, this formulation is equivalent to considering the Navier–Stokes equations in the region Ω_1^t, the incompressible visco-elasticity equations in the region Ω_2^t, and coupling them by imposing the continuity of velocity and traction at the fluid–solid interface [4].

3.1. Strong form

The strong form of the immersed formulation may be stated as: Given $\rho_1 \in \mathbb{R}^+$, $\rho_2 \in \mathbb{R}^+$, $\mu_1 \in \mathbb{R}^+$, $\mu_2 \in \mathbb{R}^+ \cup \{0\}$, $G \in \mathbb{R}^+$, $\boldsymbol{g}_M \in \Omega \times (0,T) \mapsto \mathbb{R}^d$, $\boldsymbol{g}_V \in \Omega \times (0,T) \mapsto \mathbb{R}^d$, $\boldsymbol{v}_0 : \Omega \mapsto \mathbb{R}^d$, $\boldsymbol{u}_0 : \Omega_2^0 \mapsto \mathbb{R}^d$, and $\boldsymbol{v}_B : \Gamma \times (0,T) \mapsto \mathbb{R}^d$, find $\boldsymbol{v} : \Omega \times (0,T) \mapsto \mathbb{R}^d$, $p : \Omega \times (0,T) \mapsto \mathbb{R}$, and $\boldsymbol{u} : \Omega_2^0 \times (0,T) \mapsto \mathbb{R}^d$, such that,

$$\rho_1 \left(\left.\frac{\partial \boldsymbol{v}}{\partial t}\right|_{\boldsymbol{x}} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v} \right) = \nabla_{\boldsymbol{x}} \cdot \boldsymbol{\sigma}_1 + \rho_1 \boldsymbol{g}_M + \boldsymbol{g}_V + \mathcal{F} \qquad \text{in} \quad \Omega \times (0,T), \tag{1}$$

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v} = 0 \qquad \text{in} \quad \Omega \times (0,T), \tag{2}$$

$$\left.\frac{\partial \boldsymbol{u}}{\partial t}\right|_{\boldsymbol{X}} = \boldsymbol{v} \qquad \text{in} \quad \overline{\Omega_2^t} \times (0,T), \tag{3}$$

$$\boldsymbol{\sigma}_1 \boldsymbol{n}_1 = -\boldsymbol{\sigma}_2 \boldsymbol{n}_2 \qquad \text{on} \quad \Gamma_I^t \times (0,T), \tag{4}$$

$$\boldsymbol{v} = \boldsymbol{v}_0 \qquad \text{on} \quad \Omega \times \{0\}, \tag{5}$$

$$\boldsymbol{u} = \boldsymbol{u}_0 \qquad \text{on} \quad \Omega_2^0 \times \{0\}, \tag{6}$$

$$\boldsymbol{v} = \boldsymbol{v}_B \qquad \text{on} \quad \Gamma \times (0,T), \tag{7}$$

with

$$\mathcal{F} = \begin{cases} 0, & \boldsymbol{x} \in \Omega_1^t, \\ (\rho_1 - \rho_2) \left(\left.\frac{\partial \boldsymbol{v}}{\partial t}\right|_{\boldsymbol{x}} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v} - \boldsymbol{g}_M \right) + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{\sigma}_2 - \boldsymbol{\sigma}_1), & \boldsymbol{x} \in \Omega_2^t, \end{cases} \tag{8}$$

and constitutive laws

$$\boldsymbol{\sigma}_1 = -p\boldsymbol{I} + \boldsymbol{\sigma}_1^v, \tag{9}$$

$$\boldsymbol{\sigma}_2 = -p\boldsymbol{I} + \boldsymbol{\sigma}_2^e + \boldsymbol{\sigma}_2^v. \tag{10}$$

Here, ρ_1 and ρ_2 are the fluid and solid densities, respectively, μ_1 and μ_2 are the fluid and solid dynamic viscosities, respectively, G is the shear modulus of the solid, $\boldsymbol{g}_M$ is an external force per unit of mass acting on the system, $\boldsymbol{g}_V$ is an external force per unit of volume acting on the system, $\boldsymbol{v}_0$ is the velocity initial data, $\boldsymbol{u}_0$ is the displacement initial data, $\boldsymbol{v}_B$ is the velocity boundary data, and $\boldsymbol{I}$ denotes the identity tensor in $\mathbb{R}^{d \times d}$. $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ are the fluid and solid Cauchy stress tensors, respectively, $\boldsymbol{\sigma}_1^v = 2\mu_1 \nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v}$ is the viscous part of the fluid Cauchy stress tensor, $\boldsymbol{\sigma}_2^v = 2\mu_2 \nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v}$ is the viscous part of the solid Cauchy stress tensor, $\nabla_{\boldsymbol{x}}^{\text{sym}}(\cdot)$ is the symmetric gradient operator given by $\nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v} = (\nabla_{\boldsymbol{x}} \boldsymbol{v} + \nabla_{\boldsymbol{x}} \boldsymbol{v}^T)/2$, $\boldsymbol{\sigma}_2^e = \boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T/J$ is the elastic part of the solid Cauchy stress tensor, $\boldsymbol{F} : \Omega_2^0 \times (0,T) \mapsto \mathbb{R}^{d \times d}$ is the deformation gradient given by $\boldsymbol{F} = \nabla_{\boldsymbol{X}} \boldsymbol{\varphi} = \boldsymbol{I} + \nabla_{\boldsymbol{X}} \boldsymbol{u}$, $\boldsymbol{S} = 2\partial\psi/\partial\boldsymbol{C}$ denotes the second Piola–Kirchhoff stress tensor of a hyperelastic incompressible material, ψ is the strain-energy density function, $\boldsymbol{C} = \boldsymbol{F}^T \boldsymbol{F}$ denotes the right Cauchy–Green deformation tensor, and $J = \det(\boldsymbol{F})$ is the Jacobian determinant, which is equal to 1 as the solid is incompressible. Note that $\mu_2 \geq 0$, that is, we assume that μ_2 might be equal to zero, in which case the solid is purely elastic. Unless mentioned otherwise, the following incompressible Neo-Hookean material will be our choice to construct the elastic part of the solid Cauchy stress tensor

$$\psi = \frac{G}{2}(I_1 - d), \tag{11}$$

$$\boldsymbol{S} = G\boldsymbol{I}, \tag{12}$$

where I_1 is the first invariant of the right Cauchy–Green deformation tensor.

Eqs. [illegible] represent the linear momentum balance equation, the mass conservation equation, and the kinematic equation that relates the Lagrangian displacement with the Eulerian velocity, respectively. Note that Eq. (2) imposes the incompressibility constraint in the whole domain Ω. Eq. (4) imposes the continuity of traction at the fluid–solid interface. Continuity of velocity at the fluid–solid interface is imposed through Eq. (3). Eqs. (5)–(6) define the initial condition for the velocity and the displacement, respectively. Eq. (7) represents the boundary condition for the fluid. For shortness, Dirichlet boundary conditions are assumed to be applied on the whole boundary of the fluid in Sections 3 and 4, but Neumann and periodic boundary conditions can be applied in the DCIB method following standard procedures as done in Section 5.

3.2. Weak form

Let $\mathcal{S}_v$, $\mathcal{S}_p$, and $\mathcal{S}_u$ denote trial solution spaces for velocity, pressure, and displacement, respectively, defined as follows

$$\mathcal{S}_v = \left\{ \boldsymbol{v} \mid \boldsymbol{v}(\cdot, t) \in \left(H^1(\Omega)\right)^d, \ \boldsymbol{v} = \boldsymbol{v}_B \quad \text{on} \quad \Gamma \times (0, T) \right\}, \tag{13}$$

$$\mathcal{S}_p = \left\{ p \mid p(\cdot, t) \in L^2(\Omega), \quad \int_\Omega p \, \mathrm{d}\Omega = 0 \right\}, \tag{14}$$

$$\mathcal{S}_u = \left\{ \boldsymbol{u} \mid \boldsymbol{u}(\cdot, t) \in \left(H^1\left(\Omega_2^0\right)\right)^d \right\}. \tag{15}$$

Here, $L^2(\Omega)$ denotes the space of scalar-valued functions that are square-integrable on Ω, while $\left(H^1(\Omega)\right)^d$ denotes the space of square-integrable $\mathbb{R}^d$-valued functions with square-integrable derivatives on Ω. Analogously, we define weighting function spaces $\mathcal{V}_v$, $\mathcal{V}_p$, and $\mathcal{V}_u$ for the momentum, continuity, and kinematic equations as follows

$$\mathcal{V}_v = \left\{ \boldsymbol{w} \mid \boldsymbol{w}(\cdot) \in \left(H^1(\Omega)\right)^d, \ \boldsymbol{w} = 0 \quad \text{on} \quad \Gamma \times (0, T) \right\}, \tag{16}$$

$$\mathcal{V}_p = \left\{ q \mid q(\cdot) \in L^2(\Omega) \right\}, \tag{17}$$

$$\mathcal{V}_u = \left\{ \boldsymbol{s} \mid \boldsymbol{s}(\cdot) \in \left(H^1\left(\Omega_2^0\right)\right)^d \right\}. \tag{18}$$

The weak form of the immersed formulation may be stated as: Find $\boldsymbol{v} \in \mathcal{S}_v$, $p \in \mathcal{S}_p$, and $\boldsymbol{u} \in \mathcal{S}_u$, such that,

$$B\left((\boldsymbol{w}, q, \boldsymbol{s}), (\boldsymbol{v}, p, \boldsymbol{u})\right) - L(\boldsymbol{w}) = 0 \qquad \forall (\boldsymbol{w}, q, \boldsymbol{s}) \in \mathcal{V}_v \times \mathcal{V}_p \times \mathcal{V}_u, \tag{19}$$

with

$$\begin{aligned} B\left((\boldsymbol{w}, q, \boldsymbol{s}), (\boldsymbol{v}, p, \boldsymbol{u})\right) = {} & \left(\boldsymbol{w}, \rho_1 \left.\frac{\partial \boldsymbol{v}}{\partial t}\right|_{\boldsymbol{x}} \right)_\Omega - \left(\nabla_{\boldsymbol{x}} \boldsymbol{w}, \rho_1 \boldsymbol{v} \otimes \boldsymbol{v}\right)_\Omega \\ & - \left(\nabla_{\boldsymbol{x}} \cdot \boldsymbol{w}, p\right)_\Omega + \left(\nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{w}, 2\mu_1 \nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v}\right)_\Omega \\ & + \left(q, \nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}\right)_\Omega - \left(\boldsymbol{w}, (\rho_1 - \rho_2) \left.\frac{\partial \boldsymbol{v}}{\partial t}\right|_{\boldsymbol{x}} \right)_{\Omega_2^0} \\ & + \left(\nabla_{\boldsymbol{x}} \boldsymbol{w}, (\rho_1 - \rho_2)\, \boldsymbol{v} \otimes \boldsymbol{v}\right)_{\Omega_2^0} + \left(\nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{w}, \boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T\right)_{\Omega_2^0} \\ & + \left(\nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{w}, 2\mu_2 \nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v}\right)_{\Omega_2^0} - \left(\nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{w}, 2\mu_1 \nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v}\right)_{\Omega_2^0} \\ & + \left(\boldsymbol{s}, \left.\frac{\partial \boldsymbol{u}}{\partial t}\right|_{\boldsymbol{X}} - \boldsymbol{v} \right)_{\Omega_2^0}, \end{aligned} \tag{20}$$

$$L(\boldsymbol{w}) = \left(\boldsymbol{w}, \rho_1 \boldsymbol{g}_M\right)_\Omega - \left(\boldsymbol{w}, (\rho_1 - \rho_2)\, \boldsymbol{g}_M\right)_{\Omega_2^0} + \left(\boldsymbol{w}, \boldsymbol{g}_V\right)_\Omega, \tag{21}$$

where $(\cdot, \cdot)_\Omega$ and $(\cdot, \cdot)_{\Omega_2^0}$ denote the L^2 inner product over the domain Ω and Ω_2^0, respectively. In Eq. (20), note that the convection term has been brought into the conservative form and that the continuity of traction at the fluid–solid interface is naturally enforced.

4. Discretization

In this section, we describe the DCIB method, which discretizes the variational form defined in Eqs. (19)–(21). We first discretize in space leading to a semi-discrete formulation, and then perform time discretization.

4.1. Spatial discretization

The spatial discretization of the Eulerian velocity, the Eulerian pressure, and their weighting functions is performed using divergence-conforming B-splines while the spatial discretization of the Lagrangian displacement and its associated weighting function is performed using NURBS.

4.1.1. Univariate B-splines

A knot vector is a finite non-decreasing sequence of real numbers $\Xi_1 = \{\xi_{1,1}, \xi_{2,1}, \ldots, \xi_{n_1+p_1+1,1}\}$, where $\xi_{i,1}$ is the ith knot, p_1 is the polynomial degree, and n_1 is the number of univariate B-spline basis functions defined by the knot vector. Given Ξ_1, univariate B-spline basis functions are constructed recursively starting with piecewise constants ($p_1 = 0$)

$$\widehat{B}_i^0(\xi_1) = \begin{cases} 1 & \text{if } \xi_{i,1} \le \xi_1 < \xi_{i+1,1}, \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

For $p_1 > 0$, univariate B-spline basis functions are defined by

$$\widehat{B}_i^{p_1}(\xi_1) = \frac{\xi_1 - \xi_{i,1}}{\xi_{i+p_1,1} - \xi_{i,1}} \widehat{B}_i^{p_1-1}(\xi_1) + \frac{\xi_{i+p_1+1,1} - \xi_1}{\xi_{i+p_1+1,1} - \xi_{i+1,1}} \widehat{B}_{i+1}^{p_1-1}(\xi_1), \tag{23}$$

which is the Cox–de Boor recursion formula (see [52]). Note that when $\xi_{i+p_1,1} - \xi_{i,1}$ is zero, $\frac{\xi_1 - \xi_{i,1}}{\xi_{i+p_1,1} - \xi_{i,1}}$ is set to be zero, and analogously, when $\xi_{i+p_1+1,1} - \xi_{i+1,1}$ is zero, $\frac{\xi_{i+p_1+1,1} - \xi_1}{\xi_{i+p_1+1,1} - \xi_{i+1,1}}$ is set to be zero. The support of a univariate B-spline basis function of order p_1 is always $p_1 + 1$ knot spans.

Since the knots can be repeated, we define a sequence of knots without repetitions (henceforth known as breakpoints) $\boldsymbol{\eta}_1 = \{\eta_{1,1}, \eta_{2,1}, \ldots, \eta_{m,1}\}$ and another sequence with their multiplicities $\boldsymbol{r}_1 = \{r_{1,1}, r_{2,1}, \ldots, r_{m,1}\}$. The univariate B-spline basis functions are p_1-degree piecewise polynomials on $\boldsymbol{\eta}_1$ and the subintervals $\left[\eta_{j,1}, \eta_{j+1,1}\right]$ constitute a set of non-empty knot spans, analogous to one-dimensional elements in the finite element method (FEM). Knot multiplicity enables to control the continuity of splines, namely, if a breakpoint $\eta_{j,1}$ has multiplicity $r_{j,1}$, the basis functions will have at least $\alpha_{j,1} = p_1 - r_{j,1}$ continuous derivatives at $\eta_{j,1}$. Let us denote the sequence collecting the regularities as $\boldsymbol{\alpha}_1 = \{\alpha_{1,1}, \alpha_{2,1}, \ldots, \alpha_{m,1}\}$. In this exposition, we restrict ourselves to the case of open knot vectors (i.e., $r_{1,1} = r_{m,1} = p_1 + 1$ and $1 \le r_{j,1} \le p_1$ for $j = 2, \ldots, m-1$). This leads to $n_1 \ge p_1 + 1$, $\alpha_{j,1} \ge 0$ for $j = 2, \ldots, m-1$, and $\alpha_{1,1} = \alpha_{m,1} = -1$. Let us also define $\boldsymbol{\alpha}_1 + \mathbf{1} = \{-1, \alpha_{2,1}+1, \alpha_{3,1}+1, \ldots, \alpha_{m-1,1}+1, -1\}$ and $|\boldsymbol{\alpha}_1| = \min\{\alpha_{2,1}, \alpha_{3,1}, \ldots, \alpha_{m-1,1}\}$. For convenience, we consider $\eta_{1,1} = 0$ and $\eta_{m,1} = 1$, that is, the parametric domain is $(0, 1)$.

The basis functions $\{\widehat{B}_i^{p_1}(\xi_1)\}_{i=1}^{n_1}$ form a partition of unity, have local support, and are non-negative [54]. The univariate B-spline space spanned by these basis functions is denoted by

$$\mathcal{S}_{\boldsymbol{\alpha}_1}^{p_1} = \operatorname{span}\{\widehat{B}_i^{p_1}(\xi_1)\}_{i=1}^{n_1}. \tag{24}$$

For $p_1 > 1$ and $\alpha_{j,1} \ge 0$ with $j = 2, \ldots, m-1$, the derivative of a spline is another spline. Furthermore, the derivative is a surjective operator, that is,

$$\mathcal{S}_{\boldsymbol{\alpha}_1-\mathbf{1}}^{p_1-1} \equiv \left\{ \frac{df}{d\xi} \mid f \in \mathcal{S}_{\boldsymbol{\alpha}_1}^{p_1} \right\}. \tag{25}$$

B-spline spaces support two types of refinement, namely, knot insertion and degree elevation, which are akin to h-refinement and p-refinement in FEM. These two types of refinement do not commute, and the application of degree elevation followed by knot insertion yields a refinement procedure with no analogue in FEM, the so-called k-refinement. Moreover, knot insertion and degree elevation always lead to a sequence of nested B-spline spaces. For additional details and examples, see [54,51].

4.1.2. Multivariate B-splines

Given the polynomial degrees p_l in each direction, the number of univariate B-spline basis functions in each direction n_l, the knot vectors $\Xi_l = \{\xi_{1,l}, \xi_{2,l}, \ldots, \xi_{n_l+p_l+1,l}\}$, and the associated vectors $\boldsymbol{\eta}_l = \{\eta_{1,l}, \eta_{2,l}, \ldots, \eta_{m_l,l}\}$, $\boldsymbol{r}_l = \{r_{1,l}, r_{2,l}, \ldots, r_{m_l,l}\}$, $\boldsymbol{\alpha}_l = \{\alpha_{1,l}, \alpha_{2,l}, \ldots, \alpha_{m_l,l}\}$ with $l = 1, \ldots, d$, multivariate B-spline basis functions in the parametric domain $(0,1)^d$ are defined by tensor products of the univariate ones as

$$\widehat{B}_{i_1,\ldots,i_d}^{p_1,\ldots,p_d}(\xi_1, \ldots, \xi_d) = \prod_{l=1}^{d} \widehat{B}_{i_l}^{p_l}(\xi_l), \tag{26}$$

with $i_1 = 1, \ldots, n_1, \ldots, i_d = 1, \ldots, n_d$. As their univariate counterparts, multivariate B-spline basis functions are piecewise polynomial, form a partition of unity, have local support, and are non-negative. The multivariate B-spline space spanned by these basis functions is denoted by

$$\mathcal{S}_{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_d}^{p_1,\ldots,p_d} = \operatorname{span}\{\widehat{B}_{i_1,\ldots,i_d}^{p_1,\ldots,p_d}(\xi_1, \ldots, \xi_d)\}_{i_1=1,\ldots,i_d=1}^{n_1,\ldots,n_d}. \tag{27}$$

This space is fully characterized by the polynomial degree in each direction, the number of univariate B-spline basis functions in each direction, and the knot vector in each direction. The global continuity of the space is defined as $\alpha = \min\{|\boldsymbol{\alpha}_1|, \ldots, |\boldsymbol{\alpha}_d|\}$, that is, the B-spline functions that belong to the space are at least $\mathcal{C}^\alpha$-continuous throughout the parametric domain.

The sets of breakpoints $\boldsymbol{\eta}_l = \{\eta_{1,l}, \eta_{2,l}, \ldots, \eta_{m_l,l}\}$ with $l = 1, \ldots, d$ define a structured mesh

$$\mathcal{M}_h = \left\{ Q = (\eta_{i_1,1}, \eta_{i_1+1,l}) \times \ldots \times (\eta_{i_d,d}, \eta_{i_d+1,d}), 1 \le i_l \le m_l - 1, l = 1, \ldots, d \right\} \tag{28}$$

on the parametric domain, known as parametric mesh, where $h = \max_{Q \in \mathcal{M}_h} h_Q$ is the global mesh size and h_Q is a characteristic size of element Q. For future reference, we also denote the total number of elements in a mesh as $n_{el} = \prod_{l=1}^{d} (m_l - 1)$.

4.1.3. NURBS

In certain situations, it is useful to employ rational basis functions, typically to reproduce exactly a particular geometry. Given a set of positive real numbers called weights $\{w_{i_1,\ldots,i_d}\}_{i_1=1,\ldots,i_d=1}^{n_1,\ldots,n_d}$, the NURBS basis functions are obtained from the B-spline basis functions as follows

$$\widehat{N}_{i_1,\ldots,i_d}^{p_1,\ldots,p_d}(\xi_1, \ldots, \xi_d) = \frac{w_{i_1,\ldots,i_d} \widehat{B}_{i_1,\ldots,i_d}^{p_1,\ldots,p_d}(\xi_1, \ldots, \xi_d)}{\sum_{\hat{i}_1=1,\ldots,\hat{i}_d=1}^{n_1,\ldots,n_d} w_{\hat{i}_1,\ldots,\hat{i}_d} \widehat{B}_{\hat{i}_1,\ldots,\hat{i}_d}^{p_1,\ldots,p_d}(\xi_1, \ldots, \xi_d)} . \tag{29}$$

As their polynomial counterparts, these rational basis functions form a partition of unity, have local support, and are non-negative. Note that if all the weights equal 1, we recover the B-spline basis functions. The NURBS space $\mathrm{span}\{\widehat{N}_{i_1,\ldots,i_d}^{p_1,\ldots,p_d}(\xi_1, \ldots, \xi_d)\}_{i_1=1,\ldots,i_d=1}^{n_1,\ldots,n_d}$ is fully characterized by the polynomial degree in each direction, the number of univariate B-spline basis functions in each direction, the knot vector in each direction, and the weights.

In the next sections, for the sake of shortness, we will use a global numbering for the basis functions that goes from 1 to $n = \prod_{l=1}^{d} n_l$ and vector notation for the parametric coordinates $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_d)$.

4.1.4. Eulerian discretization

In a divergence-conforming B-spline discretization, the velocity and pressure spaces are defined on the parametric domain $\widehat{\Omega}$ as follows

$$\widehat{\mathcal{S}}_v^{h^E} = \left\{ \widehat{\boldsymbol{v}}^{h^E} \mid \widehat{\boldsymbol{v}}^{h^E}(\cdot, t) \in \widehat{VEL}^{h^E}, \ \widehat{\boldsymbol{v}}^{h^E} \cdot \widehat{\boldsymbol{n}} = \widehat{\boldsymbol{v}_B} \cdot \widehat{\boldsymbol{n}} \quad \text{on} \quad \widehat{\Gamma} \times (0, T) \right\}, \tag{30}$$

$$\widehat{\mathcal{S}}_p^{h^E} = \left\{ \widehat{p}^{h^E} \mid \widehat{p}^{h^E}(\cdot, t) \in \widehat{PRE}^{h^E}, \quad \int_{\widehat{\Omega}} \widehat{p}^{h^E} \, \mathrm{d}\widehat{\Omega} = 0 \right\}, \tag{31}$$

with

$$\widehat{VEL}^{h^E} = \begin{cases} \mathcal{S}_{\boldsymbol{\alpha}_1+1,\boldsymbol{\alpha}_2}^{p_1+1,p_2} \times \mathcal{S}_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2+1}^{p_1,p_2+1} & \text{if } d = 2, \\ \mathcal{S}_{\boldsymbol{\alpha}_1+1,\boldsymbol{\alpha}_2,\boldsymbol{\alpha}_3}^{p_1+1,p_2,p_3} \times \mathcal{S}_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2+1,\boldsymbol{\alpha}_3}^{p_1,p_2+1,p_3} \times \mathcal{S}_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\alpha}_3+1}^{p_1,p_2,p_3+1} & \text{if } d = 3, \end{cases} \tag{32}$$

$$\widehat{PRE}^{h^E} = \begin{cases} \mathcal{S}_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2}^{p_1,p_2} \times \mathcal{S}_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2}^{p_1,p_2} & \text{if } d = 2, \\ \mathcal{S}_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\alpha}_3}^{p_1,p_2,p_3} \times \mathcal{S}_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\alpha}_3}^{p_1,p_2,p_3} \times \mathcal{S}_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\alpha}_3}^{p_1,p_2,p_3} & \text{if } d = 3, \end{cases} \tag{33}$$

where the superscript E stands for “Eulerian”. When $\alpha \ge 0$, the velocity and pressure discrete spaces belong to H^1. Note that only the normal Dirichlet boundary condition has been imposed on the velocity discrete space, the tangential Dirichlet boundary conditions will be imposed weakly using Nitsche’s method in the semi-discrete form.

Regarding the weighting functions $\boldsymbol{w}$ and q, we use the Bubnov–Galerkin method, which leads to

$$\widehat{\mathcal{V}}_v^{h^E} = \left\{ \widehat{\boldsymbol{w}}^{h^E} \mid \widehat{\boldsymbol{w}}^{h^E}(\cdot) \in \widehat{VEL}^{h^E}, \ \widehat{\boldsymbol{w}}^{h^E} \cdot \widehat{\boldsymbol{n}} = 0 \quad \text{on} \quad \widehat{\Gamma} \times (0, T) \right\}, \tag{34}$$

$$\widehat{\mathcal{V}}_p^{h^E} = \left\{ \widehat{q}^{h^E} \mid \widehat{q}^{h^E}(\cdot) \in \widehat{PRE}^{h^E} \right\}. \tag{35}$$

For future reference, we define the basis functions $\widehat{N}_{v_l,A_l}^{E}$ such that $\mathrm{span}\{\widehat{N}_{v_1,A_1}^{E}(\boldsymbol{\xi}^E)\}_{A_1=1}^{n^{v_1}} \times \ldots \times \mathrm{span}\{\widehat{N}_{v_d,A_d}^{E}(\boldsymbol{\xi}^E)\}_{A_d=1}^{n^{v_d}} = \widehat{\mathcal{V}}_v^{h^E}$ and $\widehat{N}_{p,B}^{E}$ such that $\mathrm{span}\{\widehat{N}_{p,B}^{E}(\boldsymbol{\xi}^E)\}_{B=1}^{n^p} = \widehat{\mathcal{V}}_p^{h^E}$, where n^{v_l} is the total number of degrees of freedom that the lth component of the velocity has and n^p is the number of degrees of freedom that the pressure has.

The above choices of spaces lead to discrete solutions that are both *inf–sup* stable and *pointwise* divergence-free. The latter property follows from the surjectivity of the divergence operator at the discrete level, namely, $\widehat{\nabla} \cdot \widehat{\mathcal{S}}_v^{h^E} = \widehat{\mathcal{S}}_p^{h^E}$. As we impose the condition $\left(\widehat{q}^{h^E}, \widehat{\nabla} \cdot \widehat{\boldsymbol{v}}^{h^E}\right)_{\widehat{\Omega}} = 0 \ \ \forall \widehat{q}^{h^E} \in \widehat{\mathcal{V}}_p^{h^E}$, we can take $\widehat{q}^{h^E} = \widehat{\nabla} \cdot \widehat{\boldsymbol{v}}^{h^E}$, obtaining $\|\widehat{\nabla} \cdot \widehat{\boldsymbol{v}}^{h^E}\|_{\mathcal{L}^2(\widehat{\Omega})} = 0$. Then, we have $\widehat{\nabla} \cdot \widehat{\boldsymbol{v}}^{h^E} = 0$ pointwise (see [48,50] for additional details). Note that obtaining an exact pointwise divergence-free discrete velocity field would imply to exactly solve the final system of algebraic equations, which is impractical, but approximately solving the final systems of algebraic equations with strict tolerances leads to a highly accurate satisfaction of the incompressibility constraint as we will show in the numerical examples of this paper.

In order to define the discrete trial and weighting function spaces on the physical domain Ω^{h^E}, let us start defining the geometrical mapping $\boldsymbol{F}^E : \widehat{\Omega} \mapsto \Omega^{h^E}$ as

$$\boldsymbol{F}^E\left(\boldsymbol{\xi}^E\right) = \sum_{D=1}^{n^E} \boldsymbol{P}^E_D \widehat{N}^E_D\left(\boldsymbol{\xi}^E\right) \quad \boldsymbol{\xi}^E \in \widehat{\Omega}\,, \tag{36}$$

where $\boldsymbol{P}^E$ are the control points that define the geometry of the physical domain and $\widehat{N}^E_D$ are a set of NURBS basis functions used to define the geometry of the physical domain. The parametric mesh defined in $\widehat{\Omega}$, called the Eulerian mesh, can be pushed forward to the physical domain Ω^{h^E} using the geometric mapping $\boldsymbol{F}^E$. This computational mesh will be used to compute all the integrals that are posed in the domain Ω^{h^E}.

The discrete trial and weighting function spaces on the physical domain Ω^{h^E} are defined using the following transformations

$$\mathcal{S}^{h^E}_v = \left\{\boldsymbol{v}^{h^E} \mid \boldsymbol{v}^{h^E} \circ \boldsymbol{F}^E = \frac{1}{\det(D\boldsymbol{F}^E)} D\boldsymbol{F}^E \widehat{\boldsymbol{v}}^{h^E}, \quad \widehat{\boldsymbol{v}}^{h^E} \in \widehat{\mathcal{S}}^{h^E}_v\right\}, \tag{37}$$

$$\mathcal{S}^{h^E}_p = \left\{p^{h^E} \mid p^{h^E} \circ \boldsymbol{F}^E = \frac{1}{\det(D\boldsymbol{F}^E)} \widehat{p}^{h^E}, \quad \widehat{p}^{h^E} \in \widehat{\mathcal{S}}^{h^E}_p\right\}, \tag{38}$$

$$\mathcal{V}^{h^E}_v = \left\{\boldsymbol{w}^{h^E} \mid \boldsymbol{w}^{h^E} \circ \boldsymbol{F}^E = \frac{1}{\det(D\boldsymbol{F}^E)} D\boldsymbol{F}^E \widehat{\boldsymbol{w}}^{h^E}, \quad \widehat{\boldsymbol{w}}^{h^E} \in \widehat{\mathcal{V}}^{h^E}_v\right\}, \tag{39}$$

$$\mathcal{V}^{h^E}_p = \left\{q^{h^E} \mid q^{h^E} \circ \boldsymbol{F}^E = \frac{1}{\det(D\boldsymbol{F}^E)} \widehat{q}^{h^E}, \quad \widehat{q}^{h^E} \in \widehat{\mathcal{V}}^{h^E}_p\right\}, \tag{40}$$

where $D\boldsymbol{F}^E$ is the gradient of the geometrical mapping $\boldsymbol{F}^E$. For the velocity, we are using the Piola transform, which is a standard choice to construct discrete spaces in $H(\mathrm{div};\Omega)$ within the context of mixed finite elements [66]. The Piola transform preserves divergence and the normal component of the transformed vector field, which leads to pointwise divergence-free discrete velocities on the physical domain Ω^{h^E} that satisfy the normal Dirichlet boundary condition. For the pressure, we are using an integral-preserving transformation, which guarantees satisfaction of the zero-mean pressure constraint on the physical domain Ω^{h^E}. Note that the total number of Eulerian degrees of freedom $(EDOF)$ is $EDOF = \dim \mathcal{S}^{h^E}_v + \dim \mathcal{S}^{h^E}_p = n^{v_1} + \ldots + n^{v_d} + n^p$. In the simulations of this paper, we always work with uniform parametric Eulerian meshes with $p_1 = \ldots = p_d = k$ and $\boldsymbol{\alpha}_1 = \ldots = \boldsymbol{\alpha}_d$ equal to constant sequences with constant component $\alpha = k-1$.[1]

The weak form of our immersed formulation involves the first spatial derivatives of the velocity field, these are obtained applying the chain rule to the Piola transform resulting in

$$\frac{\partial v^{h^E}_i}{\partial x_j} = \frac{(DF^E_{i\hat{i}} \frac{\partial \widehat{v}^{h^E}_{\hat{i}}}{\partial \xi^E_{\hat{l}}} - (DF^E)^{-1}_{\hat{j}l} \frac{\partial DF^E_{l\hat{j}}}{\partial \xi^E_{\hat{l}}} DF^E_{i\hat{i}} \widehat{v}^{h^E}_{\hat{i}} + \frac{\partial DF^E_{i\hat{i}}}{\partial \xi^E_{\hat{l}}} \widehat{v}^{h^E}_{\hat{i}})(DF^E)^{-1}_{\hat{l}j}}{\det(D\boldsymbol{F}^E)}, \tag{41}$$

where i, j, l are spatial coordinate indexes and $\hat{i}, \hat{j}, \hat{l}$ are parametric coordinate indexes, all of them go from 1 to d.

4.1.5. Lagrangian discretization

Let us consider a set of NURBS basis functions $\{\widehat{N}^L_C(\boldsymbol{\xi}^L)\}^{n^L}_{C=1}$ defined over the parametric domain $\widehat{\Omega}^0_2$, where the superscript L stands for "Lagrangian". The displacement field is defined on this parametric domain as follows

$$\widehat{\boldsymbol{u}}^{h^L}(\boldsymbol{\xi}^L, t) = \sum_{C=1}^{n^L} \boldsymbol{u}_C(t) \widehat{N}^L_C(\boldsymbol{\xi}^L) \quad \boldsymbol{\xi}^L \in \widehat{\Omega}^0_2\,, \tag{42}$$

where $\boldsymbol{u}_C(t)$ are the degrees of freedom, known as control variables in IGA parlance, of the displacement field. Invoking the isoparametric concept, the basis functions $\{\widehat{N}^L_C(\boldsymbol{\xi}^L)\}^{n^L}_{C=1}$ are also used to define a mapping $\boldsymbol{F}^L : \widehat{\Omega}^0_2 \mapsto \Omega^{0,h^L}_2$ as

$$\boldsymbol{F}^L\left(\boldsymbol{\xi}^L\right) = \sum_{C=1}^{n^L} \boldsymbol{P}^L_C \widehat{N}^L_C\left(\boldsymbol{\xi}^L\right) \quad \boldsymbol{\xi}^L \in \widehat{\Omega}^0_2\,, \tag{43}$$

---

[1] Note that $\alpha = -1$ gives rise to the standard Raviart–Thomas mixed finite elements, which are not H^1 conforming.

where $\boldsymbol{P}^L$ are the control points that define the material domain of the solid. The parametric mesh defined in $\widehat{\Omega}_2^0$, called the Lagrangian mesh, can be pushed forward to the material domain Ω_2^{0,h^L} using the mapping $\boldsymbol{F}^L$. This computational mesh will be used to compute all the integrals that are posed in the domain Ω_2^{0,h^L}.

Using the mapping $\boldsymbol{F}^L$, the displacement field defined on the material domain ($\boldsymbol{u}^{h^L} \in \mathcal{S}_u^{h^L} = \text{span}\{N_C^L(\boldsymbol{X})\}_{C=1}^{n^L} \subset \mathcal{S}_{\boldsymbol{u}}$) can be obtained with the following pushforward operation

$$\boldsymbol{u}^{h^L}(\boldsymbol{X},t) = \widehat{\boldsymbol{u}}^{h^L} \circ (\boldsymbol{F}^L)^{-1} = \sum_{C=1}^{n^L} \boldsymbol{u}_C(t) N_C^L(\boldsymbol{X}), \tag{44}$$

where $(\boldsymbol{F}^L)^{-1}$ is the inverse mapping of $\boldsymbol{F}^L$ and $N_C^L(\boldsymbol{X})$ is the Cth basis function defined on the material domain as $N_C^L(\boldsymbol{X}) = \widehat{N}_C^L \circ (\boldsymbol{F}^L)^{-1}$. When needed, the Lagrangian mesh can be pushed forward to the physical space using the discrete mapping $\boldsymbol{\varphi}^{h^L}(\boldsymbol{X},t)$. The total number of Lagrangian degrees of freedom ($LDOF$) is $LDOF = \text{dim}\mathcal{S}_u^{h^L} = dn^L$. In the simulations of this paper, we always work with uniform parametric Lagrangian meshes with the same degree p in all directions.

Regarding the weighting function $\boldsymbol{s}$, we use the Bubnov–Galerkin method. This leads to the same underlying space to discretize the trial and weighting functions ($\mathcal{V}_{\boldsymbol{u}}^{h^L} = \mathcal{S}_u^{h^L} \subset \mathcal{V}_{\boldsymbol{u}}$). Therefore, we have

$$\boldsymbol{s}^{h^L}(\boldsymbol{X}) = \sum_{C=1}^{n^L} \boldsymbol{s}_C N_C^L(\boldsymbol{X}). \tag{45}$$

4.1.6. *Semi-discrete form*

The semi-discrete form of the immersed formulation may be stated as: Find $\boldsymbol{v}^{h^E} \in \mathcal{S}_v^{h^E}$, $p^{h^E} \in \mathcal{S}_p^{h^E}$, and $\boldsymbol{u}^{h^L} \in \mathcal{S}_u^{h^L}$, such that for all $\boldsymbol{w}^{h^E} \in \mathcal{V}_v^{h^E}$, $q^{h^E} \in \mathcal{V}_p^{h^E}$, and $\boldsymbol{s}^{h^L} \in \mathcal{V}_u^{h^L}$

$$B\left((\boldsymbol{w}^{h^E}, q^{h^E}, \boldsymbol{s}^{h^L}), (\boldsymbol{v}^{h^E}, p^{h^E}, \boldsymbol{u}^{h^L})\right) - b\left(\boldsymbol{w}^{h^E}, \boldsymbol{v}^{h^E}\right) - L\left(\boldsymbol{w}^{h^E}\right) + l\left(\boldsymbol{w}^{h^E}\right) = 0, \tag{46}$$

with

$$\begin{aligned} b\left(\boldsymbol{w}^{h^E}, \boldsymbol{v}^{h^E}\right) = & \sum_{F \in \Gamma^{h^E}} \int_F 2\mu_1 (\nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{w}^{h^E} \boldsymbol{n}^{h^E}) \cdot (\boldsymbol{v}^{h^E})_{||} d\Gamma + \sum_{F \in \Gamma^{h^E}} \int_F 2\mu_1 (\nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v}^{h^E} \boldsymbol{n}^{h^E}) \cdot (\boldsymbol{w}^{h^E})_{||} d\Gamma \\ & - \sum_{F \in \Gamma^{h^E}} \int_F 2\mu_1 \frac{C_{pen}}{h_F} (\boldsymbol{v}^{h^E})_{||} \cdot (\boldsymbol{w}^{h^E})_{||} d\Gamma, \end{aligned} \tag{47}$$

$$l\left(\boldsymbol{w}^{h^E}\right) = \sum_{F \in \Gamma^{h^E}} \int_F 2\mu_1 (\nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{w}^{h^E} \boldsymbol{n}^{h^E}) \cdot (\boldsymbol{v}_B)_{||} d\Gamma - \sum_{F \in \Gamma^{h^E}} \int_F 2\mu_1 \frac{C_{pen}}{h_F} (\boldsymbol{v}_B)_{||} \cdot (\boldsymbol{w}^{h^E})_{||} d\Gamma, \tag{48}$$

where the terms $b\left(\boldsymbol{w}^{h^E}, \boldsymbol{v}^{h^E}\right)$ and $l\left(\boldsymbol{w}^{h^E}\right)$ have been added to the semi-discrete form to weakly impose the tangential Dirichlet boundary conditions using Nitsche's method, $\boldsymbol{n}^{h^E}$ is the outward unit normal vector to Ω^{h^E}, $(\cdot)_{||} = (\cdot) - ((\cdot) \cdot \boldsymbol{n}^{h^E})\boldsymbol{n}^{h^E}$ is the vector tangential component, h_F is the mesh size in the direction normal to the face F, and C_{pen} is the Nitsche's penalization parameter. All the numerical results of this paper use the value $C_{pen} = 5(k+1)$ as proposed in [48].

The discretization of the integrals that are posed on Ω_2^0 is not standard. Gaussian quadrature rules defined on the elements of the Lagrangian mesh are used to compute these integrals. However, their integrands contain functions defined in Eulerian coordinates, namely, the weighting functions $\boldsymbol{w}^{h^E}$, the velocity $\boldsymbol{v}^{h^E}$, and their first derivatives. The procedure that we use is as follows. For each Gauss point of the Lagrangian mesh with parametric coordinates $\boldsymbol{\xi}_G^L$, we first need to compute its physical location $\boldsymbol{x}_G$ through the use of $\boldsymbol{\varphi}^{h^L}$. Then, we have to invert the geometrical mapping of the Eulerian mesh to obtain the parametric coordinates $\boldsymbol{\xi}_G^E$ associated with the physical point $\boldsymbol{x}_G$ in the Eulerian mesh. In many cases of practical interest, NURBS geometrical mappings can be inverted analytically and when that is not the case, we invert the mapping by solving a nonlinear system using the Newton–Raphson algorithm. Once we know $\boldsymbol{\xi}_G^E$, we can evaluate functions defined in Eulerian coordinates through standard procedures of finite elements. This procedure, as any other procedure that the authors are aware of to compute these integrals, is suboptimal because the functions defined in Eulerian coordinates are not being integrated in a fully conforming manner. Note that the functions defined in Eulerian coordinates will generally have lines of reduced continuity in the interior of the Lagrangian elements. Nevertheless, the higher inter-element continuity of spline functions enables to alleviate this issue in comparison with using classical C^0 finite elements, namely, using $k \geq 2$ leads to continuous viscous stresses in the whole computational domain Ω. Assuming that Gauss quadrature with $(p+1)^d$

and $(k+1)^d$ quadrature points[2] are used per Lagrangian and Eulerian element, respectively, the Lagrangian elements should be at least as small as the Eulerian elements underneath to have accurate results since this would be the configuration needed to obtain conforming quadrature rules in case the elements of the two meshes were aligned with each other. In all the simulations of this paper, we have used Lagrangian elements slightly smaller than the Eulerian elements together with $(p+1)^d$ quadrature points per Lagrangian element. Moreover, we show in Section 5.2 how increasing the number of quadrature points per Lagrangian element leads to negligible variations in both the physical quantities of interest and the accuracy with which the incompressibility constraint is satisfied. In [67], a process was developed to compute integrals as the ones posed on Ω_0^2 in a conforming manner for one-dimensional problems, but as the authors mention, its extension to two- and three-dimensional settings together with an efficient implementation remains an open problem.

4.2. *Time discretization*

In this section, we set forth our fully-implicit time discretization based on the generalized-α method. The generalized-α algorithm was first proposed by Chung and Hulbert [68] for the second-order structural mechanics equations, and later extended by Jansen et al. [69] to the first-order fluid mechanics equations. Note that the mathematical model of the IB method is first-order in time. To illustrate our method, let us divide $[0,T]$ into a sequence of subintervals (t_n, t_{n+1}) with fixed time-step size $\Delta t = t_{n+1} - t_n$ and define the following residual vectors

$$\boldsymbol{R}^M = \left\{\boldsymbol{R}^M_l\right\}, \quad \boldsymbol{R}^M_l = \left\{R^M_{l,A_l}\right\}, \quad \boldsymbol{R}^I = \left\{R^I_B\right\}, \quad \boldsymbol{R}^K = \left\{\boldsymbol{R}^K_l\right\}, \quad \boldsymbol{R}^K_l = \left\{R^K_{l,C}\right\}, \tag{49}$$

where $A_l \in \{1,\ldots,n^{v_l}\}$, $B \in \{1,\ldots,n^p\}$, $C \in \{1,\ldots,n^L\}$, and l is a dimension index which runs from 1 to d. The components of the residual vectors are given by

$$R^M_{l,A_l} = B\left((\boldsymbol{w}_{l,A_l}, 0, \boldsymbol{0}), (\boldsymbol{v}^{h^E}, p^{h^E}, \boldsymbol{u}^{h^L})\right) - b\left(\boldsymbol{w}_{l,A_l}, \boldsymbol{v}^{h^E}\right) - L\left(\boldsymbol{w}_{l,A_l}\right) + l\left(\boldsymbol{w}_{l,A_l}\right), \tag{50}$$

$$R^I_B = B\left((\boldsymbol{0}, q_B, \boldsymbol{0}), (\boldsymbol{v}^{h^E}, p^{h^E}, \boldsymbol{u}^{h^L})\right) - b\left(\boldsymbol{0}, \boldsymbol{v}^{h^E}\right) - L\left(\boldsymbol{0}\right) + l\left(\boldsymbol{0}\right), \tag{51}$$

$$R^K_{l,C} = B\left((\boldsymbol{0}, 0, \boldsymbol{s}_{l,C}), (\boldsymbol{v}^{h^E}, p^{h^E}, \boldsymbol{u}^{h^L})\right) - b\left(\boldsymbol{0}, \boldsymbol{v}^{h^E}\right) - L\left(\boldsymbol{0}\right) + l\left(\boldsymbol{0}\right), \tag{52}$$

where $\boldsymbol{w}_{l,A_l} \circ \boldsymbol{F}^E = D\boldsymbol{F}^E \widehat{N}^E_{v_l,A_l} \widehat{\boldsymbol{e}}^E_l / \det(D\boldsymbol{F}^E)$, $q_B \circ \boldsymbol{F}^E = \widehat{N}^E_{p,B} / \det(D\boldsymbol{F}^E)$, and $\boldsymbol{s}_{l,C} \circ \boldsymbol{F}^L = \widehat{N}^L_C \widehat{\boldsymbol{e}}^L_l$. The vectors $\widehat{\boldsymbol{e}}^E_l$ and $\widehat{\boldsymbol{e}}^L_l$ represent the standard basis in the Eulerian and Lagrangian parametric domains, respectively.

Let us now define $\boldsymbol{V}_n$, $\boldsymbol{P}_n$, $\boldsymbol{A}_n$, $\overline{\boldsymbol{U}}_n$, and $\overline{\boldsymbol{V}}_n$ as the global vectors of control variables of $\boldsymbol{v}^{h^E}(\cdot, t_n)$, $p^{h^E}(\cdot, t_n)$, $\frac{\partial \boldsymbol{v}^{h^E}}{\partial t}(\cdot, t_n)$, $\boldsymbol{u}^{h^L}(\cdot, t_n)$, and $\frac{\partial \boldsymbol{u}^{h^L}}{\partial t}(\cdot, t_n)$ respectively. Using this notation, our time-integration algorithm is defined as follows: given $\boldsymbol{V}_n$, $\boldsymbol{A}_n$, $\overline{\boldsymbol{U}}_n$, and $\overline{\boldsymbol{V}}_n$, find $\boldsymbol{V}_{n+1}$, $\boldsymbol{A}_{n+1}$, $\boldsymbol{V}_{n+\alpha_f}$, $\boldsymbol{A}_{n+\alpha_m}$, $\boldsymbol{P}_{n+1}$, $\overline{\boldsymbol{U}}_{n+1}$, $\overline{\boldsymbol{V}}_{n+1}$, $\overline{\boldsymbol{U}}_{n+\alpha_f}$, and $\overline{\boldsymbol{V}}_{n+\alpha_m}$ such that

$$\boldsymbol{R}^M(\boldsymbol{V}_{n+\alpha_f}, \boldsymbol{A}_{n+\alpha_m}, \boldsymbol{P}_{n+1}, \overline{\boldsymbol{U}}_{n+\alpha_f}) = 0, \tag{53}$$

$$\boldsymbol{R}^I(\boldsymbol{V}_{n+\alpha_f}) = 0, \tag{54}$$

$$\boldsymbol{R}^K(\overline{\boldsymbol{V}}_{n+\alpha_m}, \boldsymbol{V}_{n+\alpha_f}) = 0, \tag{55}$$

$$\boldsymbol{V}_{n+\alpha_f} = \boldsymbol{V}_n + \alpha_f(\boldsymbol{V}_{n+1} - \boldsymbol{V}_n), \tag{56}$$

$$\boldsymbol{A}_{n+\alpha_m} = \boldsymbol{A}_n + \alpha_m(\boldsymbol{A}_{n+1} - \boldsymbol{A}_n), \tag{57}$$

$$\overline{\boldsymbol{U}}_{n+\alpha_f} = \overline{\boldsymbol{U}}_n + \alpha_f(\overline{\boldsymbol{U}}_{n+1} - \overline{\boldsymbol{U}}_n), \tag{58}$$

$$\overline{\boldsymbol{V}}_{n+\alpha_m} = \overline{\boldsymbol{V}}_n + \alpha_m(\overline{\boldsymbol{V}}_{n+1} - \overline{\boldsymbol{V}}_n), \tag{59}$$

$$\boldsymbol{V}_{n+1} = \boldsymbol{V}_n + \Delta t((1-\gamma)\boldsymbol{A}_n + \gamma \boldsymbol{A}_{n+1}), \tag{60}$$

$$\overline{\boldsymbol{U}}_{n+1} = \overline{\boldsymbol{U}}_n + \Delta t((1-\gamma)\overline{\boldsymbol{V}}_n + \gamma \overline{\boldsymbol{V}}_{n+1}), \tag{61}$$

where α_m, α_f and γ are real-valued parameters that control the accuracy, stability, and dissipation of the algorithm. Jansen et al. [69] showed that second-order accuracy can be attained by taking

$$\gamma = \frac{1}{2} + \alpha_m - \alpha_f, \tag{62}$$

while unconditional stability (for a linear problem) requires

---

[2] In divergence conforming B-splines, $(k+2)^d$ quadrature points should be used to guarantee optimal convergence rates in problems with enough regularity. However, as it will be explained in Section 5, the exact solution of the FSI problem will generally have low regularity at the fluid–solid interface and we find that using $(k+2)^d$ or $(k+1)^d$ quadrature points leads to essentially the same results.

$$\alpha_m \geq \alpha_f \geq \frac{1}{2}. \tag{63}$$

The real-valued parameters of the generalized-α method can be expressed in terms of ϱ_∞ (the spectral radius of the amplification matrix as $\Delta t \to \infty$) in such a way that conditions (62)–(63) are automatically satisfied. This is achieved by taking

$$\alpha_m = \frac{1}{2}\left(\frac{3-\varrho_\infty}{1+\varrho_\infty}\right), \tag{64}$$

$$\alpha_f = \gamma = \frac{1}{1+\varrho_\infty}, \tag{65}$$

with $\varrho_\infty \in [0, 1]$. By using Eqs. (64)–(65), all the eigenvalues of the amplification matrix take on the value $-\varrho_\infty$ when $\Delta t \to \infty$, which leads to optimal high frequency damping. Therefore, ϱ_∞ can be used to control high-frequency dissipation with the guarantee that second-order accuracy and unconditional stability for a linear problem are satisfied. In this paper, we choose $\varrho_\infty = 1/2$, which represents an adequate balance between accuracy and robustness in solving our nonlinear system of algebraic equations.

4.3. Block-iterative solution strategy

In order to compute the unknown control variables at time t^{n+1} in our fully-discrete form given by Eqs. (53)–(61), we use a block-iterative approach [70], namely, we derive two separate tangent matrices: one tangent matrix for the linear momentum and mass conservation residuals ($\boldsymbol{R}^M$ and $\boldsymbol{R}^I$) where the Lagrangian control variables $\overline{\boldsymbol{U}}_{n+\alpha_f}$ are considered to be constant and the other tangent matrix for the kinematic equation residual ($\boldsymbol{R}^K$) where the Eulerian control variables $\boldsymbol{V}_{n+\alpha_f}$ are considered to be constant. This fully-implicit approach follows two stages:

Predictor stage:

1. Set

$$\boldsymbol{V}_{n+1,(0)} = \boldsymbol{V}_n, \tag{66}$$

$$\boldsymbol{A}_{n+1,(0)} = \frac{(\gamma-1)}{\gamma}\boldsymbol{A}_n, \tag{67}$$

$$\boldsymbol{P}_{n+1,(0)} = \boldsymbol{P}_n, \tag{68}$$

$$\overline{\boldsymbol{U}}_{n+1,(0)} = \overline{\boldsymbol{U}}_n, \tag{69}$$

$$\overline{\boldsymbol{V}}_{n+1,(0)} = \frac{(\gamma-1)}{\gamma}\overline{\boldsymbol{V}}_n, \tag{70}$$

where the subscript 0 on the left-hand side is the nonlinear iteration index. Note that the predictions are based on Eqs. (66), (68) and (69), while Eqs. (67) and (70) are only a consequence of Eqs. (60) and (61).

2. Evaluate the control variables at intermediate time levels

$$\boldsymbol{V}_{n+\alpha_f,(0)} = \boldsymbol{V}_n + \alpha_f(\boldsymbol{V}_{n+1,(0)} - \boldsymbol{V}_n), \tag{71}$$

$$\boldsymbol{A}_{n+\alpha_m,(0)} = \boldsymbol{A}_n + \alpha_m(\boldsymbol{A}_{n+1,(0)} - \boldsymbol{A}_n), \tag{72}$$

$$\boldsymbol{P}_{n+1,(0)} = \boldsymbol{P}_{n+1,(0)}, \tag{73}$$

$$\overline{\boldsymbol{U}}_{n+\alpha_f,(0)} = \overline{\boldsymbol{U}}_n + \alpha_f(\overline{\boldsymbol{U}}_{n+1,(0)} - \overline{\boldsymbol{U}}_n), \tag{74}$$

$$\overline{\boldsymbol{V}}_{n+\alpha_m,(0)} = \overline{\boldsymbol{V}}_n + \alpha_m(\overline{\boldsymbol{V}}_{n+1,(0)} - \overline{\boldsymbol{V}}_n). \tag{75}$$

Multicorrector stage. Repeat the following steps for $i = 0, 1, 2, \ldots$ until convergence is achieved:

1. Use the intermediate time levels to assemble the residual vector $\boldsymbol{R}^K(\overline{\boldsymbol{V}}_{n+\alpha_m,(i)}, \boldsymbol{V}_{n+\alpha_f,(i)})$, assemble the tangent matrix $\boldsymbol{T}^K(\overline{\boldsymbol{V}}_{n+\alpha_m,(i)})$, and solve the linear algebraic system

$$\boldsymbol{T}^K \Delta\overline{\boldsymbol{V}}_{n+1,(i)} = -\boldsymbol{R}^K. \tag{76}$$

2. Update the Lagrangian control variables as follows

$$\overline{\boldsymbol{U}}_{n+1,(i+1)} = \overline{\boldsymbol{U}}_{n+1,(i)} + \gamma \Delta t \Delta \overline{\boldsymbol{V}}_{n+1,(i)}, \tag{77}$$

$$\overline{\boldsymbol{V}}_{n+1,(i+1)} = \overline{\boldsymbol{V}}_{n+1,(i)} + \Delta \overline{\boldsymbol{V}}_{n+1,(i)}. \tag{78}$$

   Note that the updates of $\overline{\boldsymbol{V}}_{n+1}$ are straightforward while $\overline{\boldsymbol{U}}_{n+1}$ is updated consistently with Eq. (61) of the generalized-α method.

3. Evaluate the Lagrangian control variables at intermediate time levels

$$\overline{\boldsymbol{U}}_{n+\alpha_f,(i+1)} = \overline{\boldsymbol{U}}_n + \alpha_f (\overline{\boldsymbol{U}}_{n+1,(i+1)} - \overline{\boldsymbol{U}}_n), \tag{79}$$

$$\overline{\boldsymbol{V}}_{n+\alpha_m,(i+1)} = \overline{\boldsymbol{V}}_n + \alpha_m (\overline{\boldsymbol{V}}_{n+1,(i+1)} - \overline{\boldsymbol{V}}_n). \tag{80}$$

4. Use the intermediate time levels to assemble the residual vectors $\boldsymbol{R}^M(\boldsymbol{V}_{n+\alpha_f,(i)}, \boldsymbol{A}_{n+\alpha_m,(i)}, \boldsymbol{P}_{n+1,(i)}, \overline{\boldsymbol{U}}_{n+\alpha_f,(i+1)})$ and $\boldsymbol{R}^I(\boldsymbol{V}_{n+\alpha_f,(i)})$, assemble the tangent submatrices $\boldsymbol{T}^{MI}_{11}(\boldsymbol{V}_{n+\alpha_f,(i)}, \boldsymbol{A}_{n+\alpha_m,(i)})$, $\boldsymbol{T}^{MI}_{12}(\boldsymbol{P}_{n+1,(i)})$, $\boldsymbol{T}^{MI}_{21}(\boldsymbol{V}_{n+\alpha_f,(i)})$, $\boldsymbol{T}^{MI}_{22} = \boldsymbol{0}$, and solve the linear algebraic system

$$\begin{pmatrix} \boldsymbol{T}^{MI}_{11} & \boldsymbol{T}^{MI}_{12} \\ \boldsymbol{T}^{MI}_{21} & \boldsymbol{T}^{MI}_{22} \end{pmatrix} \left\{ \begin{array}{c} \Delta \boldsymbol{A}_{n+1,(i)} \\ \Delta \boldsymbol{P}_{n+1,(i)} \end{array} \right\} = - \left\{ \begin{array}{c} \boldsymbol{R}^M \\ \boldsymbol{R}^I \end{array} \right\}. \tag{81}$$

5. Update the Eulerian control variables as follows

$$\boldsymbol{V}_{n+1,(i+1)} = \boldsymbol{V}_{n+1,(i)} + \gamma \Delta t \Delta \boldsymbol{A}_{n+1,(i)}, \tag{82}$$

$$\boldsymbol{A}_{n+1,(i+1)} = \boldsymbol{A}_{n+1,(i)} + \Delta \boldsymbol{A}_{n+1,(i)}, \tag{83}$$

$$\boldsymbol{P}_{n+1,(i+1)} = \boldsymbol{P}_{n+1,(i)} + \Delta \boldsymbol{P}_{n+1,(i)}. \tag{84}$$

   Note that the updates of $\boldsymbol{A}_{n+1}$ and $\boldsymbol{P}_{n+1}$ are straightforward while $\boldsymbol{V}_{n+1}$ is updated consistently with Eq. (60) of the generalized-α method.

6. Evaluate the Eulerian control variables at intermediate time levels

$$\boldsymbol{V}_{n+\alpha_f,(i+1)} = \boldsymbol{V}_n + \alpha_f (\boldsymbol{V}_{n+1,(i+1)} - \boldsymbol{V}_n), \tag{85}$$

$$\boldsymbol{A}_{n+\alpha_m,(i+1)} = \boldsymbol{A}_n + \alpha_m (\boldsymbol{A}_{n+1,(i+1)} - \boldsymbol{A}_n), \tag{86}$$

$$\boldsymbol{P}_{n+1,(i+1)} = \boldsymbol{P}_{n+1,(i+1)}. \tag{87}$$

Our parallel implementation is built on top of PetIGA [71] and PetIGA-MF [72–74], which gives us access to the preconditioners of the scientific library PETSc [75] and the ones in other libraries as the BoomerAMG package of *hypre* [76] and the ML library [77]. To linearize the residual $\boldsymbol{R}^K$, we use a Newton–Raphson method. To linearize the residuals $\boldsymbol{R}^M$ and $\boldsymbol{R}^I$, we use a Newton–Raphson method with a critical point line search [78]. In all the simulations performed in this paper, we stop the multicorrector stage when the L^2-norm of the residuals has decreased more than four orders of magnitude. As linear solver for the kinematic equation, we use the GMRES method [79] with an incomplete LU preconditioner. As linear solver for the momentum and mass conservation equations, we use the scalable block-preconditioning strategy defined in [73] for divergence-conforming B-splines.

5. Numerical examples

The simulations included in this section were computed using either the Bridges supercomputer, part of the Extreme Science and Engineering Discovery Environment (XSEDE) [80], or the MareNostrum supercomputer, part of the Barcelona Supercomputing Center.

In all the examples, the accuracy with which the incompressibility constraint is verified at the discrete level will be measured in both an Eulerian manner and a Lagrangian manner. The Eulerian manner consists of measuring the L^2-norm error of Eq. (2) (e_{DIV}), that is,

$$e_{DIV}(t) = \left(\int_{\Omega^{h^E}} (\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}^{h^E})^2 \, d\Omega \right)^{1/2}. \tag{88}$$

The Lagrangian manner consists of measuring the volume change of the solid (e_{VC}) as

$$e_{VC}(t) = \frac{\left| \int\limits_{\Omega_2^{t,h^L}} \mathrm{d}\Omega_t^2 - \int\limits_{\Omega_2^{0,h^L}} \mathrm{d}\Omega_0^2 \right|}{\int\limits_{\Omega_2^{0,h^L}} \mathrm{d}\Omega_0^2}. \tag{89}$$

Since we are using divergence-conforming B-splines, e_{DIV} is controlled by the accuracy with which the linear system of Eq. (81) is solved. As in [73], the iterative method used to solve the linear system is stopped when the L^2 norm of the unpreconditioned residual is less than a certain value a_{tol}, which is set to 10^{-10} in all our simulations unless when mentioned otherwise. e_{VC} inherits the error coming from e_{DIV} (which in our method is negligible) and adds the discretization error of solving Eq. (3). Therefore, we expect e_{VC} to be larger than e_{DIV}.

5.1. Soft disk in a lid-driven cavity flow

We first present a common benchmark problem [81,43,65,82,14] to evaluate the performance of immersed FSI methods when it comes to impose the incompressibility constraint at the discrete level. It consists of a highly deformable bulky disk entrained in a lid-driven cavity flow. As tested in [43], the softer the disk is, the more challenging it becomes to accurately impose the incompressibility constraint in an immersed method.

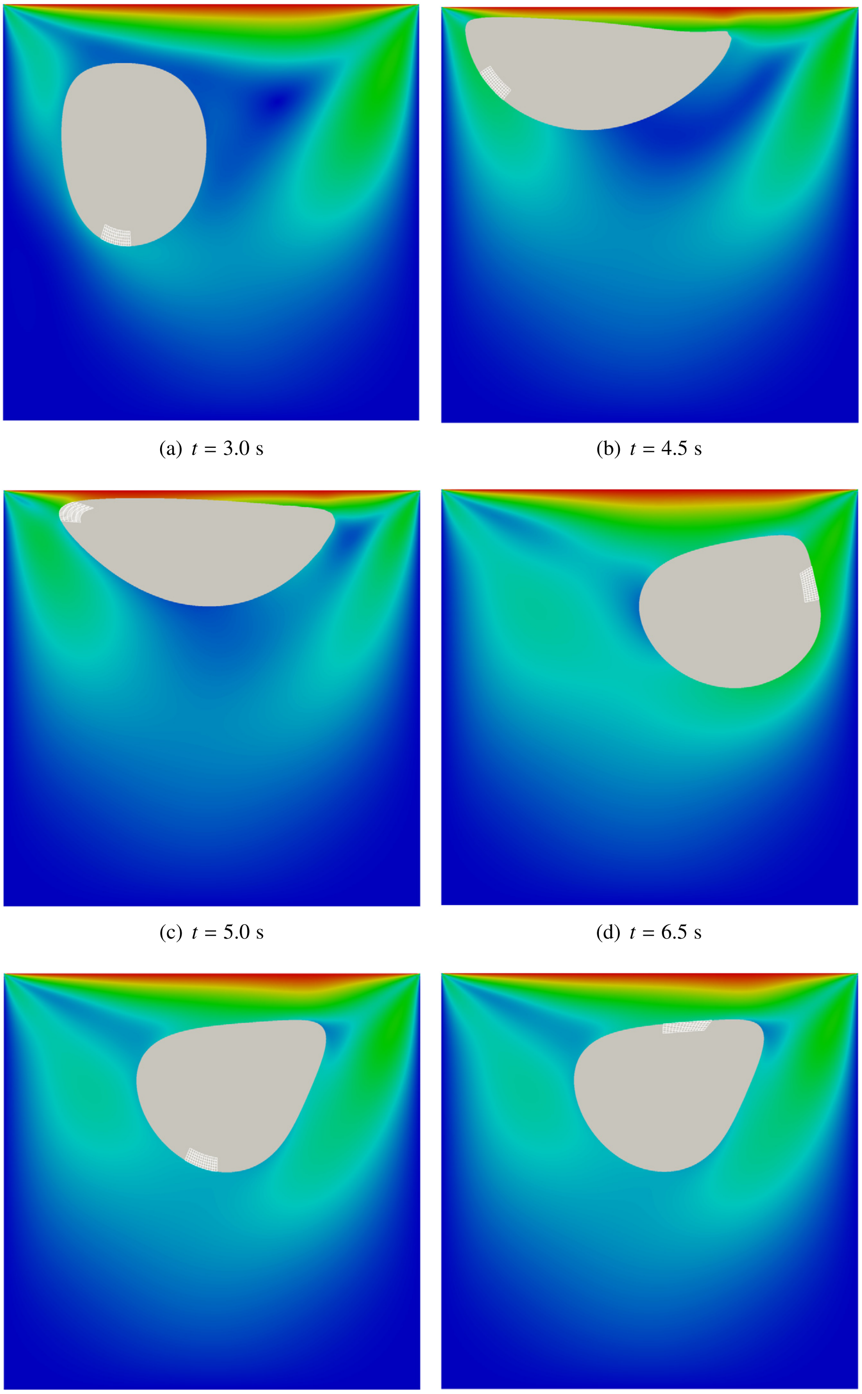
The physical domain Ω is a square with side $L = 1$ cm. The solid is a circle with diameter $b = 0.4$ cm whose center is initially located at $(0.6\ \text{cm}, 0.5\ \text{cm})$. At the top side of the square, the lid moves with a constant horizontal velocity $v_f = 1$ cm/s and its vertical velocity is zero. No-penetration and no-slip boundary conditions are imposed in the other three sides of the square. Both the fluid and the solid are initially at rest. The physical parameters that define this problem are the following: $\rho_1 = \rho_2 = 1.0\ \text{g/cm}^3$, $\mu_1 = \mu_2 = 0.01\ \text{g/(cm} \cdot \text{s)}$, $G = 0.1\ \text{g/(cm} \cdot \text{s}^2)$, $\mathbf{g}_M = (0.0\ \text{cm/s}^2, 0.0\ \text{cm/s}^2)$, and $\mathbf{g}_V = (0.0\ \text{g/(cm}^2 \cdot \text{s}^2), 0.0\ \text{g/(cm}^2 \cdot \text{s}^2))$. The channel Reynolds number, which describes the unperturbed channel flow, is $R_e^c = (\rho_1 v_f L)/\mu_1 = 100$ and the particle Reynolds number is $R_e^p = (R_e^c) b^2/L^2 = 16$. The shear modulus considered in this work is the lowest for which we have found volume-conservation studies in the literature [43,82].

The test established in the literature consists in measuring the volume-conservation error (e_{VC}) for the time interval $t \in [0.0\ \text{s}, 10.0\ \text{s}]$. During this time, the solid initially moves towards the top left corner of the square as shown in Fig. 1a). Then, it moves just underneath the lid experiencing large shear strains due to the flow boundary conditions, see Figs. 1b) and c). Finally, it moves downwards as shown in Fig. 1d). In [43], the IFEM method led to a 25% volume-conservation error before the algorithm failed to converged around $t = 5.5$ s with a relatively fine discretization (the precise mesh resolutions are not specified in the manuscript). The authors proposed in [43] a volume correction strategy to be carried out every ten time steps which reduces the volume-conservation error to 2.5%. In [82], the coarsest discretization used by the authors has 11,392 EDOF and 1,313 LDOF for which an 8.7% volume-conservation error is obtained. The finest discretization used in [82] has 182,280 EDOF and 5,252 LDOF for which a 1.9% volume-conservation error is obtained. Regarding the time step used in [82], it is chosen such that $\Delta t = 0.64 h^E$. In [14], this problem is solved using $0.02\ \text{g/(cm} \cdot \text{s)}$ as shear modulus instead of $0.01\ \text{g/(cm} \cdot \text{s)}$ as in [81,43,65,82]. The coarsest and finest discretizations used in [14] have 64 × 64 and 256 × 256 Eulerian elements leading to 0.15% and 0.029% volume-conservation errors, respectively. Regarding the time step used in [14], it is chosen such that $\Delta t = 0.125 h^E$. In this work, we start with a very coarse discretization, namely, 16 × 16 Eulerian elements (1,008 EDOF) with $k = 2$, 4×24 Lagrangian elements (336 LDOF) with $p = 2$, and time step $\Delta t = 0.08$ s. After that, we refine our discretization by performing uniform h-refinement six times in each mesh and dividing the time step by two each time a new level of refinement is introduced (see Table 1 for details). The following relation holds in our simulations $\Delta t = 1.28 h^E$. Therefore, we are using larger time steps than in [14,82]. Note that the use of quadratic NURBS for the Lagrangian mesh enables us to represent the initial geometry of the solid exactly in all discretizations considered. The value of e_{VC} for each discretization is included in Table 1 and plotted in Fig. 2a) together with data from the literature. For a given resolution and the same shear modulus, the use of divergence-conforming B-splines leads to an increased accuracy in volume conservation of various orders of magnitude in comparison with the aforementioned results from the literature. Regarding e_{DIV}, its value is lower than 5.0e−8 for all the discretizations considered. This value can be further decreased several orders of magnitude

Table 1

Discretizations considered for the soft disk in a lid-driven cavity flow along with the volume-conservation error for each discretization.

n_{el}^E	EDOF	k	n_{el}^L	LDOF	p	Δt (s)	e_{VC}
16 × 16	1,008	2	4 × 24	336	2	8.0e−2	2.2375e−3
32 × 32	3,536	2	8 × 48	520	2	4.0e−2	4.4891e−4
64 × 64	13,200	2	16 × 96	1,800	2	2.0e−2	1.3649e−4
128 × 128	50,960	2	32 × 192	6,664	2	1.0e−2	4.5708e−5
256 × 256	200,208	2	64 × 384	25,608	2	5.0e−3	1.0058e−5
512 × 512	793,616	2	128 × 768	100,360	2	2.5e−3	2.8072e−6
1024 × 1024	3,160,080	2	256 × 1536	397,320	2	1.25e−3	6.7843e−7



(a) $t = 3.0$ s

(b) $t = 4.5$ s

(c) $t = 5.0$ s

(d) $t = 6.5$ s

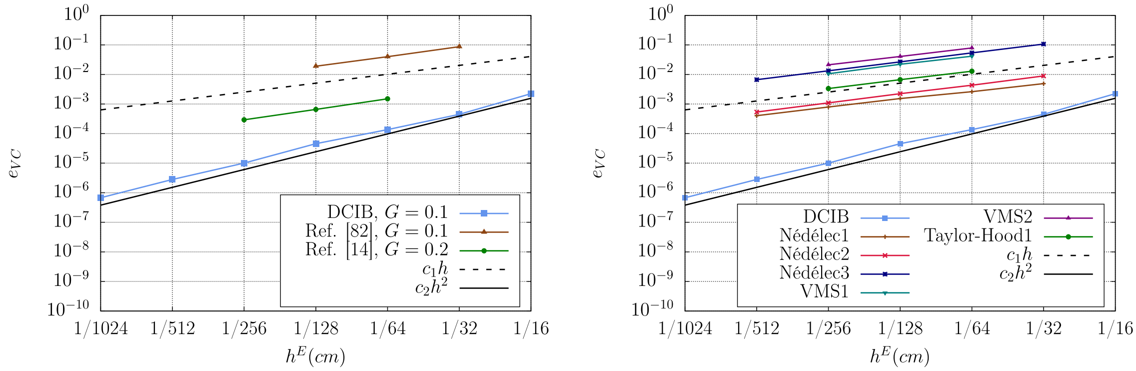
(e) $t = 42.5$ s

(f) $t = 45.0$ s

Fig. 1. Soft disk in a lid-driven cavity flow. The velocity magnitude and the deformed geometry of the solid are plotted at different times. Some Lagrangian elements are highlighted in white color so that the rotation motion undergone by the solid can be observed. In all pictures, the velocity magnitude varies from 0.0 cm/s (blue color) to 1.0 cm/s (red color). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

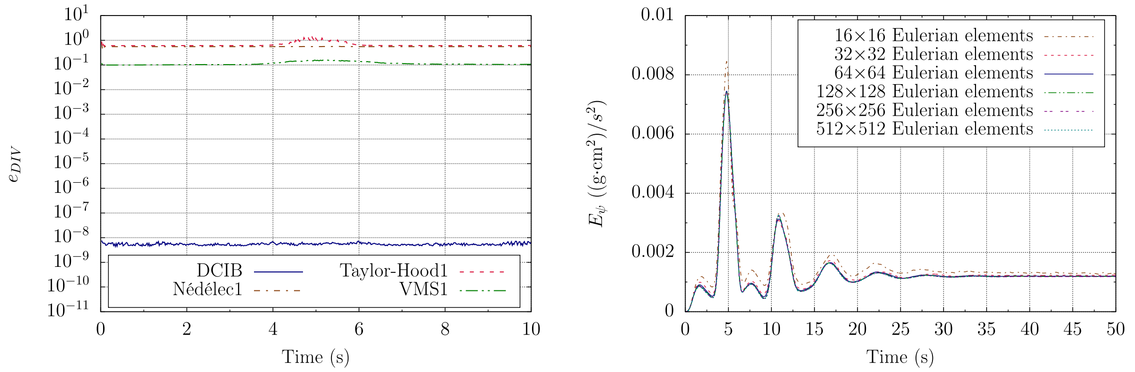
by decreasing the absolute tolerance ($a_{tol} = 10^{-10}$) with which the linear system of Eq. (81) is solved. However, doing so does not further decrease the value of e_{VC} since the discretization error of solving Eq. (3) is already the dominant error.

To show that the use of divergence-conforming B-splines is actually behind the very low values of e_{VC} and e_{DIV} reported in the aforegoing paragraph, we have solved this benchmark using other well-established IGA discretizations instead



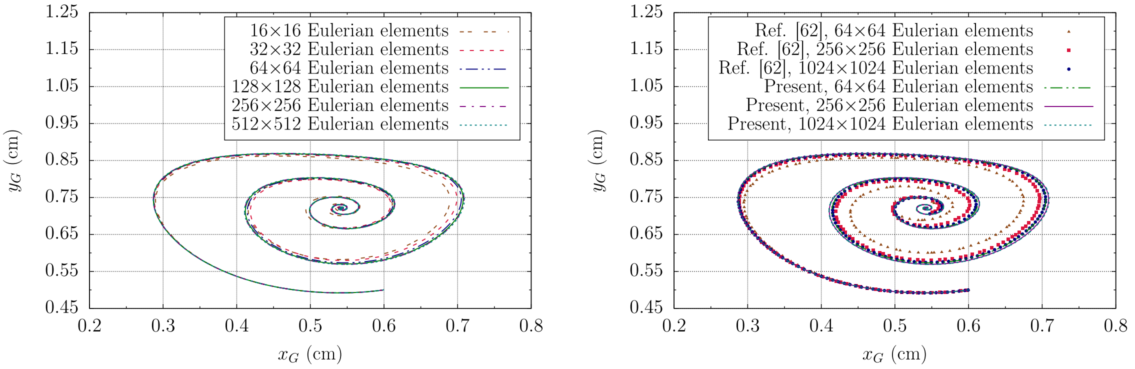
(a) e_{VC} for different immersed FSI methods

(b) e_{VC} for different IGA discretizations



(c) e_{DIV} for different IGA discretizations

(d) Time evolution of the elastic energy



(e) Evolution of (x_G,y_G)

(f) Comparison with the literature

Fig. 2. Convergence studies for the soft disk in a lid-driven cavity flow. (a) Incompressibility error at the Lagrangian level for the different discretizations detailed in Table 1 along with the data from [82,14]. Instead of having first-order convergence as in [82,14], our method leads to second-order convergence of the incompressibility error at the Lagrangian level. (b)–(c) Incompressibility error at the Lagrangian and Eulerian levels using different IGA discretizations. The use of divergence-conforming B-splines leads to extreme increases in accuracy. (d) Time evolution of the elastic energy stored by the disk. (e) Evolution of the center of gravity (x_G, y_G). In d) and e), the curves using 256 × 256 or more Eulerian elements are indistinguishable at the scale of the plot. (f) Our convergence results of (x_G, y_G) are compared with those in [65]. The simulations are stopped at time $t = 20$ s in [65] while we stopped our simulations at time $t = 50$ s, once the center of gravity has already reached a steady value. The curves of [65] seem to converge towards our fully-converged curve. Note that our mesh with 64 × 64 elements leads to the same level of accuracy than the mesh with 1024 × 1024 elements in [65], which is the finest mesh considered in that paper. In all of our discretizations, $\Delta t = 1.28h^E$.

of divergence-conforming B-splines while keeping all other aspects of our algorithm the same. In [47], besides of proposing divergence-conforming B-splines as a smooth spline generalization of Raviart–Thomas elements, smooth spline generalizations of Nédélec elements of second kind and Taylor–Hood elements are proposed. All these three smooth spline generalizations are *inf–sup* stable and H^1 conforming, but only divergence-conforming B-splines lead to pointwise divergence-free solutions. Using the incompressible Neo-Hookean material defined in Eq. (11), the spline generalization of Nédélec elements of second kind led to various orders of magnitude of reduced accuracy in e_{VC}, the results are plotted in Fig. 2b) under the tag Nédélec3. The spline generalization of Taylor–Hood elements led to even larger values of e_{VC} and failed to converge around $t = 5.5$ s as in [43]. In [6], it is mentioned that using a compressible material law instead of its incompressible analog leads to significantly lower values of e_{VC} in immersed FSI methods that impose the incompressibility constraint in both the fluid and the solid domains in an Eulerian manner through Eq. (2). The compressible analog of Eq. (11) is

$$\psi = \frac{1}{2}G\left(J^{-2/d}I_1 - d\right) + \frac{1}{2}\kappa\left(\frac{1}{2}(J^2-1) - \ln J\right). \tag{90}$$

Using Eq. (90) instead of Eq. (11) with $\kappa = 0$ g/(cm $\cdot$ s) significantly decreases the values of e_{VC} when using the spline generalization of Nédélec elements of second kind; the results are plotted in Fig. 2b) under the tag Nédélec2. The spline generalization of Taylor–Hood elements using Eq. (90) with $\kappa = 0$ g/(cm $\cdot$ s) still failed to converge around $t = 5.5$ s. As mentioned in [6], considering a value of κ different than zero, further decreases the value of e_{VC}, but it makes it harder to converge the final system of equations. The largest value of κ for which we are able to obtain convergence is $\kappa = 0.5$ g/(cm $\cdot$ s). With this value of κ, the spline generalization of Taylor–Hood elements are now able to complete the test. The results for the spline generalizations of both Nédélec and Taylor–Hood elements with $\kappa = 0.5$ g/(cm $\cdot$ s) are included in Fig. 2b) under the tags Nédélec1 and Taylor–Hood1, respectively. The most widespread stabilized IGA discretization is variational multiscale (VMS) [83]. The VMS discretization also fails to converge around $t = 5.5$ s when Eq. (11) is used. However, it is able to converge when Eq. (90) is used with either $\kappa = 0$ g/(cm $\cdot$ s) and $\kappa = 0.5$ g/(cm $\cdot$ s), the values of e_{VC} are plotted in Fig. 2b) under the tags VMS2 and VMS1, respectively. We have used Eq. (90) instead of Eq. (11) using divergence-conforming B-splines, unlike the other IGA discretizations, the values of e_{VC} are essentially the same. Therefore, we will use Eq. (11) in the next examples, which is the material law consistent with the fact that Eq. (2) is imposed in both the fluid and the solid domains.

Fig. 2c) plots the time evolution of e_{DIV} for the four IGA discretizations considered in this example using 64 × 64 Eulerian elements. Divergence-conforming B-splines lead to more than seven orders of magnitude of increased accuracy in comparison with the other IGA discretizations. Figs. 2c) and b) evince that the large error of immersed FSI methods in mass conservation is at the Eulerian level and then this error is propagated to the Lagrangian level when the Lagrangian displacement is computed from the Eulerian velocity using Eq. (3). The use of divergence-conforming B-splines eliminates the incompressibility error at the Eulerian level and this in turn leads to a large drop of the incompressibility error at the Lagrangian level. We believe that measuring e_{DIV} in this and other benchmarks for immersed FSI methods is at least as important as measuring e_{VC}.

As can be seen in Fig. 2a) the DCIB method leads to second-order convergence of e_{VC} as the discretization is refined. However, as shown in Figs. 2a) and b), e_{VC} converges at first order in both the immersed FSI methods from the literature [82,14] and the other IGA discretizations considered in this paper. This difference in the convergence of e_{VC} is probably due to the fact that when pointwise divergence-free Eulerian discretizations as divergence-conforming B-splines are used, the Lagrangian displacement does not inherit any incompressibility error coming from the Eulerian velocity and the only source of error in e_{VC} is the discretization error of solving the kinematic equation that relates the Lagrangian velocity with the Eulerian velocity (Eq. (3)).[3] However, for other IGA discretizations and in [82,14], the main source of error in e_{VC} is the incompressibility error coming from the Eulerian velocity which seems to decrease with only first-order convergence.

Our simulations are run until the disk is trapped in the main vortex, leading to a steady value of the solid's center of gravity around $t = 45$ s. Although the center of gravity reaches a steady value, the solid keeps rotating around itself as can be seen in Figs. 1e) and f) where the boundaries of some Lagrangian elements are plotted in white color so that the rotation motion can be observed. In Fig. 2b), the time evolution of the elastic energy stored in the solid ($E_\psi = \int_{\Omega_2^0} \psi \, \mathrm{d}\Omega_2^0$) is plotted for different mesh resolutions. Convergence is obtained as we refine our discretization, namely, the results with 256 × 256 Eulerian elements or more are indistinguishable at the scale of the plot. The evolution of the center of gravity (x_G, y_G) for different discretizations is plotted in Fig. 2c). Again, the results with 256 × 256 Eulerian elements or more are indistinguishable at the scale of the plot. In [65], the evolution of the center of gravity is studied for various discretizations using a full Eulerian finite difference approach for solving FSI problems. Note that in [65] the simulations are stopped at time $t = 20$ s. Both convergence studies are plotted in Fig. 2d) for comparison. Although resolutions of up to 1024 × 1024 elements are reached in [65], the trajectories plotted in Fig. 11 of [65] and reproduced here in Fig. 2d) do not reach a fully-converged result. Having said that, the trajectories in [65] clearly exhibit a convergent trend. As can be seen in Fig. 2d), this trend clearly moves towards the fully-converged curve obtained here with the DCIB method.

Finally, note that the DCIB method is able to accurately and automatically transfer data between meshes when the solid is on top of the Eulerian element row touching the domain boundary, which happens in this example when 32 × 32 elements

---

[3] In order to minimize this discretization error of Eq. (3) for a given mesh resolution, the DCIB method solves this equation in variational form using the Bubnov–Galerkin method. This is in contrast with our previous works [15,16,25] where Eq. (3) was solved in strong form using isogeometric collocation [84,85].

or less are used in the Eulerian mesh. As mentioned in [43,81], managing this kind of situations is not straightforward when discretized Dirac delta functions are used to transfer data between meshes, namely, a simplified interpolation rule is usually used for the points that are too close to the domain boundary.

5.2. Oscillating disk surrounded by fluid

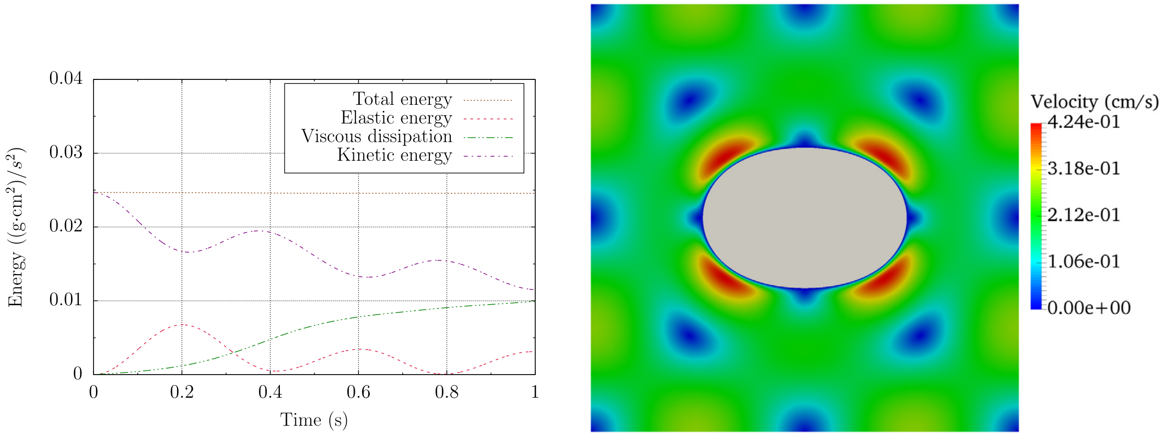
We next consider a benchmark problem [81,86] to evaluate the performance of immersed FSI methods with respect to their energy conservation at the discrete level. It consists of a deformable bulky disk embedded in a flow driven by the initial condition of the Eulerian velocity. The system behaves as a damped oscillator.

The physical domain Ω is a square with side $L = 1$ cm. The solid is a circle with diameter $b = 0.4$ cm whose center is initially located at (0.5 cm, 0.5 cm). Periodic boundary conditions are imposed in all sides of the square. The disk is initially stress free and the initial condition for the Eulerian velocity is defined through the stream function $\phi = \phi_0 \sin(k_x x)\sin(k_y y)$, where $\phi_0 = 0.05$ cm^2/s and $k_x = k_y = 2\pi$. The physical parameters proposed in [81] to solve this problem are the following: $\rho_1 = \rho_2 = 1.0$ g/cm^3, $\mu_1 = \mu_2 = 0.001$ g/(cm · s), $G = 1.0$ g/(cm · s^2), $\mathbf{g}_M = (0.0\ \text{cm/s}^2, 0.0\ \text{cm/s}^2)$, and $\mathbf{g}_V = (0.0\ \text{g/(cm}^2 \cdot \text{s}^2), 0.0\ \text{g/(cm}^2 \cdot \text{s}^2))$.

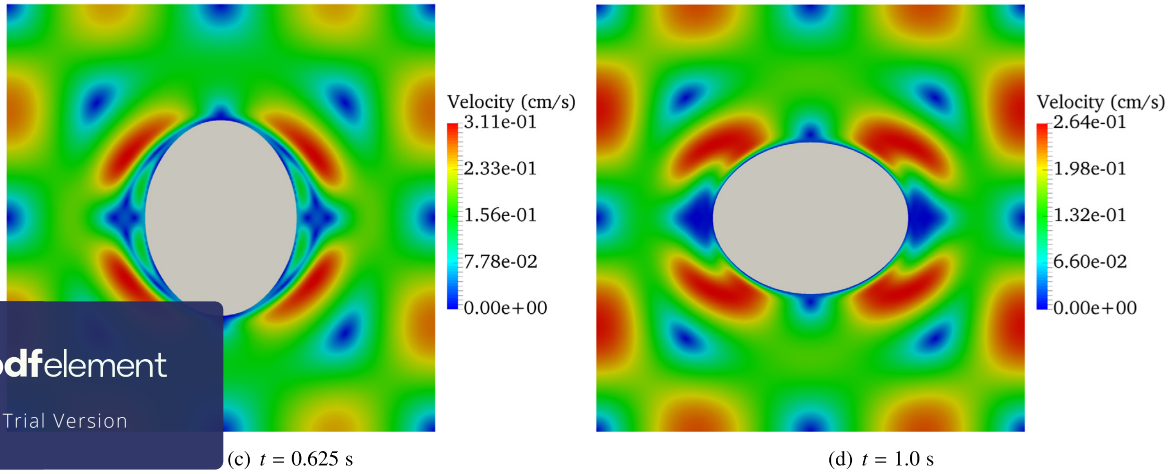
At the continuous level, the total energy (E), that is, the sum of the kinetic energy (E_k), the viscous dissipation (E_d), and elastic energy of the solid (E_ψ) is constant:

$$E = E_k + E_d + E_\psi, \tag{91}$$

with



(a) Energy balance (b) $t = 0.2$ s



(c) $t = 0.625$ s (d) $t = 1.0$ s

Fig. 3. Oscillating disk surrounded by fluid with an imposed initial condition for the velocity. (a) Evolution of the different types of energy for the time interval $t \in [0.0\ \text{s}, 1.0\ \text{s}]$. (b)–(d) The velocity magnitude and the deformed geometry of the solid are plotted at different times.

Table 2
Energy-conservation error for the oscillating disk surrounded by fluid.

Numerical method	ρ_2 (g/cm^3)	n_{el}^E	e_E
DCIB	1.0	80 × 80	8.5634e−3
DCIB	1.0	160 × 160	4.53130e−3
DCIB	1.0	320 × 320	2.4661e−3
Ref. [81]	1.0	128 × 128	1.2e−2
Ref. [86]	1.0	40 × 40 + 2 levels of local refinement	1.6e−2
DCIB	0.01	80 × 80	1.6001e−2
DCIB	0.1	80 × 80	1.2992e−2
DCIB	2.0	80 × 80	8.24688e−3
Ref. [86]	2.0	40 × 40 + 2 levels of local refinement	2.2e−2

$$E_k = \frac{\rho_1}{2}\int\limits_{\Omega} \boldsymbol{v}\cdot\boldsymbol{v}\,\mathrm{d}\Omega + \frac{\rho_2-\rho_1}{2}\int\limits_{\Omega_2^t} \boldsymbol{v}\cdot\boldsymbol{v}\,\mathrm{d}\Omega_2^t, \tag{92}$$

$$E_d = \int\limits_0^t\int\limits_{\Omega} \mu_1 \nabla_{\boldsymbol{x}}^{\mathrm{sym}}\boldsymbol{v} : \nabla_{\boldsymbol{x}}\boldsymbol{v}\,\mathrm{d}\Omega\,\mathrm{d}s, \tag{93}$$

$$E_\psi = \int\limits_{\Omega_2^0} \frac{G}{2}(I_1 - d)\,\mathrm{d}\Omega_2^0. \tag{94}$$

The test established in the literature consists in measuring the energy-conservation error ($e_E = |E(t) - E(0)|/E(0)$) for the time interval $t \in [0.0\ \mathrm{s}, 1.0\ \mathrm{s}]$. Fig. 3a) plots the evolution of the different types of energy in this interval. Figs. 3b)–d) shows the velocity magnitude together with the solid deformation at three different times.

We start with a coarse discretization, namely, 80×80 Eulerian elements with $k = 2$, 24×144 Lagrangian elements with $p = 2$, and time step $\Delta t = 0.001$ s. After that, the discretization is refined by performing uniform h-refinement two times in each mesh and dividing the time step by two each time a new level of refinement is introduced. In Table 2, we include the value of e_E for the three discretizations considered in this work as well as the energy-conservation error obtained in [81,86]. For a given resolution, the DCIB method leads to a lower energy-conservation error.

As discussed in Section 4.1.6, using $(p+1)^d$ quadrature points per Lagrangian element to compute the integrals that are posed in Ω_2^0 is suboptimal. Using our coarsest discretization, we run simulations using $(3)^2$, $(4)^2$, $(5)^2$, ..., $(10)^2$ quadrature points per Lagrangian element. The variations in the total energy depending on the number of quadrature points being used per Lagrangian element are completely negligible, namely, smaller than 0.02% with respect to the value obtained using $(p+1)^d$ quadrature points per Lagrangian element. Analogously, the variations in the area of the solid are smaller than 0.0002% with respect to the value obtained using $(p+1)^d$ quadrature points per Lagrangian element. We have done this test in other examples of this manuscript and the variations were negligible in all cases. Therefore, the accuracy with which these integrals are computed is not compromising the overall accuracy of our algorithm.

We now consider the following values for the solid density: $\rho_2 = 0.01$ g/cm^3, $\rho_2 = 0.1$ g/cm^3, and $\rho_2 = 2.0$ g/cm^3. The value of e_E for each of these density values is included in Table 2 using our coarsest discretization. The value of e_{DIV} is always lower than 9.0e−8 for all discretizations considered. Using our coarsest discretization, the value of e_{VC} is 1.04e−4, 1.18e−4, 5.19e−5, and 7.57e−5 for $\rho_2 = 0.01$ g/cm^3, $\rho_2 = 0.1$ g/cm^3, $\rho_2 = 1.0$ g/cm^3, and $\rho_2 = 2.0$ g/cm^3, respectively. With respect to e_{VC} as the discretization is refined, when $\rho_2 = 2.0$ g/cm^3, its value goes from 7.5e−5 for the coarsest discretization to 4.9e−6 for the finest discretization. These errors are various orders of magnitude lower than those obtained in [86], where e_{VC} goes from 1.1e−2 for the coarsest discretization to 2.4e−3 for the finest discretization.

5.3. Hollow disk in a shear flow

As third example, we consider a highly deformable hollow disk embedded in a shear flow. This example is used in order to numerically measure the convergence rates of the proposed method. As explained in [4,87,88,14], immersed FSI formulations often lead to exact solutions with low regularity at the fluid–solid interface. More specifically, the velocity has C^0 continuity and the pressure is discontinuous at the interface for co-dimension one solids [4,88]. This is also the case for co-dimension zero solids as long as the elastic traction of the solid at the interface is nonzero [4]. The elastic traction at the interface is nonzero for most solid materials (materials composed by fibers parallel to the interface are notable exceptions). Therefore, when the elastic traction at the interface is nonzero, $\boldsymbol{v}$ belongs to H^1, but not to H^2 and p belongs to L^2, but not to H^1. As a consequence, the convergence rates of immersed FSI methods are usually bounded by the regularity of the exact solution instead of being bounded by the approximation properties of the discretization employed.

In this example, the physical domain Ω is a rectangle with sides $L_x = 0.0032$ cm and $L_y = 0.0016$ cm. The solid is a hollow circle whose inner diameter is $a = 0.0009$ cm, its outer diameter is $b = 0.0015$ cm, and its center is initially located at $(0.0016\ \mathrm{cm}, 0.0008\ \mathrm{cm})$. A horizontal velocity $v_f = 0.08$ cm/s is applied at the top and bottom sides of the rectangle

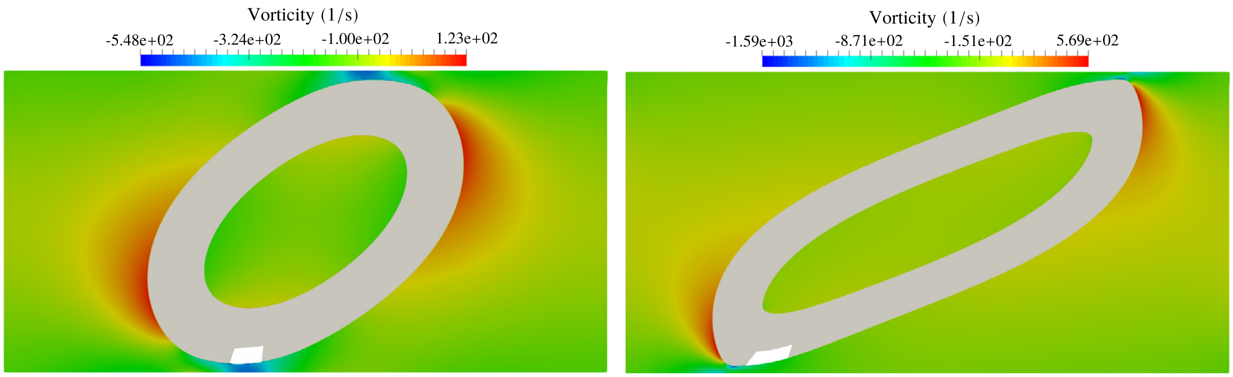
Table 3
Discretizations considered for the hollow disk in a shear flow.

n_{el}^E	EDOF	k	n_{el}^L	LDOF	p	Δt (s)
32 × 16	5,002	2	3 × 48	440	2	4.0e−4
64 × 32	12,002	2	6 × 96	1,024	2	2.0e−4
128 × 64	25,472	2	12 × 192	2,744	2	1.0e−4
256 × 128	100,096	2	24 × 384	10,088	2	5.0e−5
512 × 256	396,800	2	48 × 768	36,864	2	2.5e−5
1024 × 512	1,580,032	2	96 × 1536	147,456	2	1.25e−5

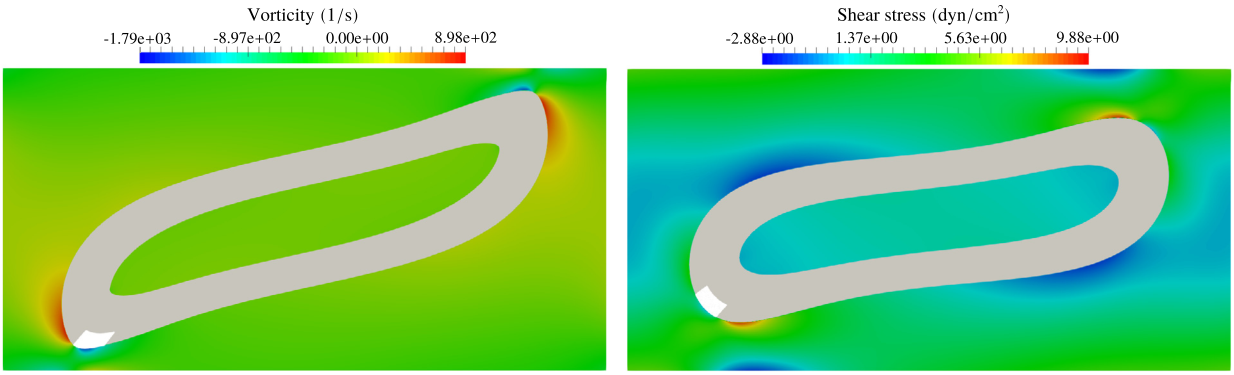
in opposite directions. No-penetration boundary condition is applied at the top and bottom sides of the rectangle. Periodic boundary conditions are applied in the left and right sides of the rectangle. If there was no solid, the applied boundary conditions would lead to a pure shear flow with shear rate $\dot{\gamma} = 100\,\text{s}^{-1}$. Both the fluid and the solid are initially at rest. The physical parameters defining this problem are the following: $\rho_1 = \rho_2 = 1.0$ g/cm^3, $\mu_1 = \mu_2 = 0.012$ g/(cm · s), $G = 20.0$ g/(cm · s^2), $\mathbf{g}_M = (0.0\text{ cm/s}^2, 0.0\text{ cm/s}^2)$, and $\mathbf{g}_V = (0.0\text{ g/(cm}^2 \cdot \text{s}^2), 0.0\text{ g/(cm}^2 \cdot \text{s}^2))$. The channel Reynolds number, which describes the unperturbed channel flow, is $R_e^c = (\rho_1 v_f L_y)/\mu_1 = 1.066\text{e}{-2}$ and the particle Reynolds number is $R_e^p = (R_e^c) b^2 / L_y^2 = 9.375\text{e}{-3}$.

We start with a very coarse discretization, namely, 32 × 16 Eulerian elements with $k = 2$, 3 × 48 Lagrangian elements with $p = 2$, and time step $\Delta t = 0.0004$ s. After that, the discretization is refined by performing uniform h-refinement five times in each mesh and dividing the time step by two each time a new level of refinement is introduced (see Table 3 for details). As in Sections 5.1 and 5.2, using quadratic NURBS for the Lagrangian mesh enables us to represent the initial geometry of the solid exactly in all discretizations considered. Figs. 4a)–f) represent the solid deformation at different times. In addition, Figs. 4a)–c) show the flow vorticity and Figs. 4d)–f) show the shear stress of the flow. Thanks to the higher inter-element continuity of divergence-conforming B-splines, we have plotted both the vorticity and the shear stress from a direct evaluation of our numerical solution, that is, without using any projection or smoothing technique. As can be seen in Fig. 5a), the inclination angle of the solid, measured as the angle between the minor principal axis of inertia of the solid and the horizontal direction, reaches a constant value around $t = 0.15$ s. Once this constant value is reached, the solid is not in a motionless configuration, instead it is rotating together with the inner fluid as can be seen in Figs. 4e) and f) by the Lagrangian elements drawn in white color. Regarding e_{DIV}, its value until the inclination angle reaches a constant value is always lower than 1.0e−10 for all discretizations considered. With respect to e_{VC}, its value until the inclination angle reaches a constant value goes from 2.7e−3 for the coarsest discretization to 1.4e−6 for the finest discretization.

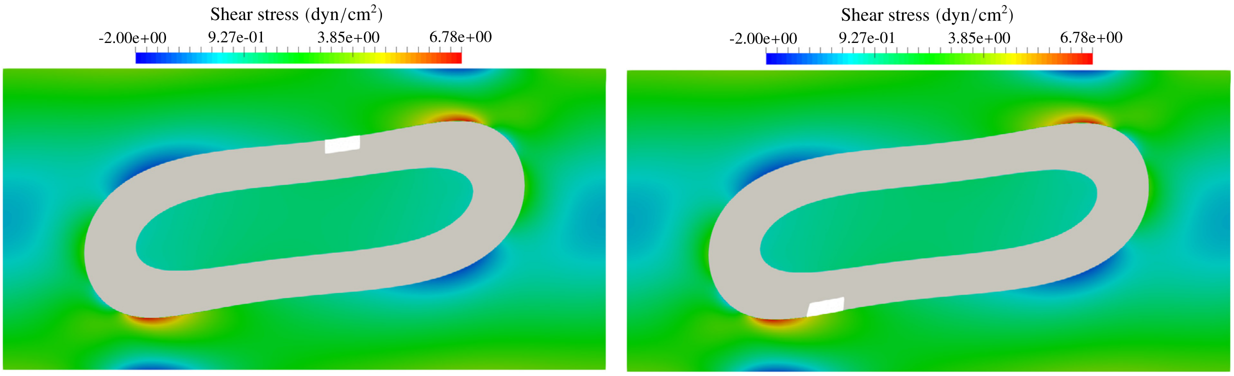
Since the exact solution of this problem is unknown, the finest discretization is used as the reference solution for computing convergence rates. The convergence rates are computed at time $t = 0.15$ s. According to Figs. 5b), c), d), e), and f) the convergence rates for the Eulerian velocity in L^2 norm, the Eulerian velocity in H^1 norm, the Eulerian pressure in L^2 norm, the Lagrangian displacement in L^2 norm, and the Lagrangian displacement in H^1 norm seem to be 1.5, 0.5, 0.75, 1.5, and 1.0 respectively. When discretized Dirac delta functions are used for the data transfer between meshes, IB methods, applied to problems with co-dimension zero solids and nonzero elastic traction at the interface, are consistently reported [14,4] to lead to 1.0 as convergence rate of both the Eulerian velocity and the Lagrangian displacement in L^2 norm. However, in [4], a variational formulation of the IB method led to a convergence rate of 1.5 for the Eulerian velocity in L^2 norm. The convergent rate of the Eulerian velocity in L^2 norm obtained in [4] is consistent with our results and it suggests that variational formulations lead to a 0.5 increase in the convergence rate of the Eulerian velocity in L^2 norm in comparison with using discretized Dirac deltas. In [4], the convergent rate of the Lagrangian displacement in L^2 norm was not measured, but our results suggest that it also increases 0.5 for variational formulations. The convergence rate of the Eulerian velocity in H^1 norm is 0.5 regardless whether or not discretized Dirac delta functions are used. With respect to the convergence rate of the Lagrangian displacement in H^1 norm, we have not found any IB numerical method that computes this norm in the literature to compare performance. It stands out that the convergence rate in H^1 norm for the Lagrangian displacement is 0.5 higher than for the Eulerian velocity. This difference may be related with the fact that the fluid–solid interface, which is the reduced-regularity region of our problem, is located in the interior of the domain in which the Eulerian velocity is defined, but it is located at the boundary of the domain in which the Lagrangian displacement is defined. A detailed study of this difference is beyond the scope of our current paper, but this difference clearly warrants additional research in the future. Regarding the convergence rate for the Eulerian pressure in L^2 norm, IB methods that use either variational formulations or discretized Dirac delta functions lead to 0.5. In this example, we seem to have mild increase of the convergence rate of the Eulerian pressure in L^2 norm, which is closer to 0.75 than 0.5. In any case, the main conclusion is that the convergence rates of immersed FSI methods under uniform h-refinement are generally bounded by the regularity of the exact solution. Therefore, it is particularly important to perform mesh-independence studies for the main physical quantities of interest when immersed FSI methods are applied to challenging applications. The accuracy per degree of freedom of immersed FSI methods can be increased through the use of adaptive local h-refinement capabilities in the Eulerian mesh, which is an appealing direction of future work for the authors.



(a) $t = 0.005$ s (b) $t = 0.015$ s



(c) $t = 0.025$ s (d) $t = 0.035$ s

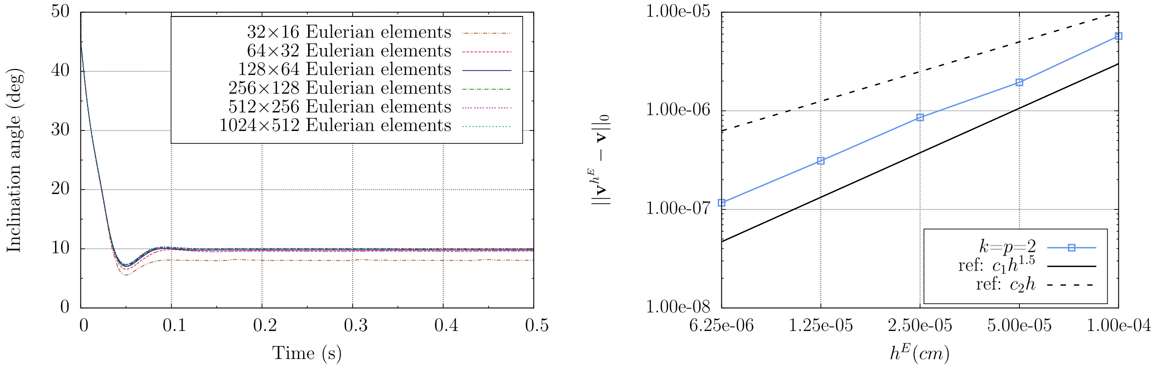


(e) $t = 0.125$ s (f) $t = 0.275$ s

Fig. 4. Hollow disk in a shear flow. (a)–(f) The deformed geometry of the solid is plotted at different times. Some Lagrangian elements are highlighted in white color so that the rotation motion undergone by the solid can be observed. (a)–(c) show the fluid vorticity and (d)–(f) show the shear stress of the fluid.

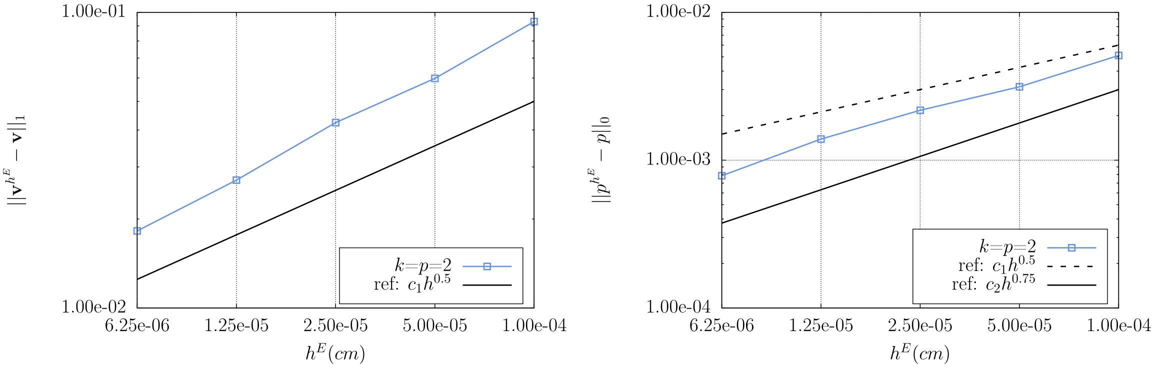
5.4. *Eight hollow spheres in a gravity-driven flow*

As fourth example, we study eight hollow spheres in a gravity-driven flow. Along the simulation, the solids will undergo nearly-contact configurations. These configurations are challenging to handle in immersed FSI methods. The reason is that immersed FSI methods, unlike body-fitted FSI methods, are not able to exactly impose the no-penetration and no-slip conditions at the fluid–solid interface since the background mesh and the Lagrangian mesh on top of it are not conforming at the fluid–solid interface. As a result, two solids in a nearly-contact configuration may potentially undergo small unphysical overlaps when the kinematic interface conditions are not imposed with enough accuracy. To avoid unphysical overlaps, additional terms based on contact theory or collision theory are often added to immersed FSI methods as in [21,38,46,89,35]. We use this three-dimensional example to check whether or not the DCIB method, without adding any additional term, leads to small unphysical overlaps between solids.



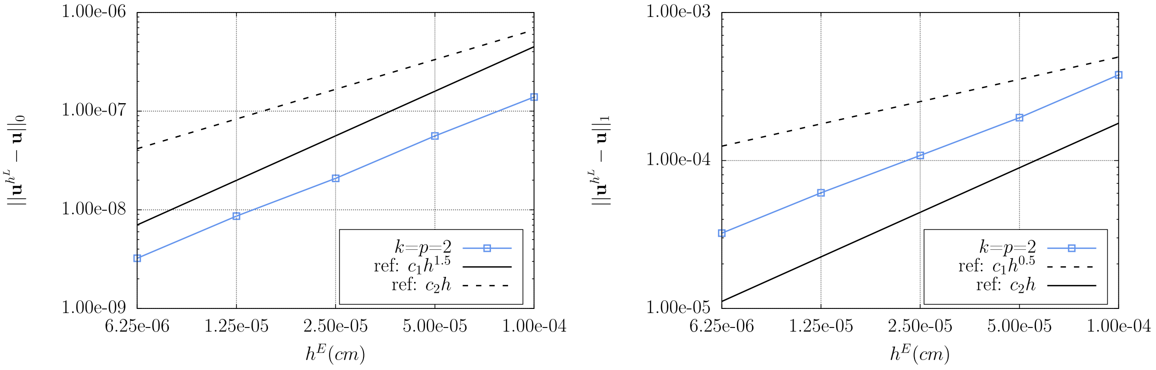
(a) Inclination angle

(b) L^2-norm convergence of $\boldsymbol{v}^{h^E}$



(c) H^1-norm convergence of $\boldsymbol{v}^{h^E}$

(d) L^2-norm convergence of p^{h^E}

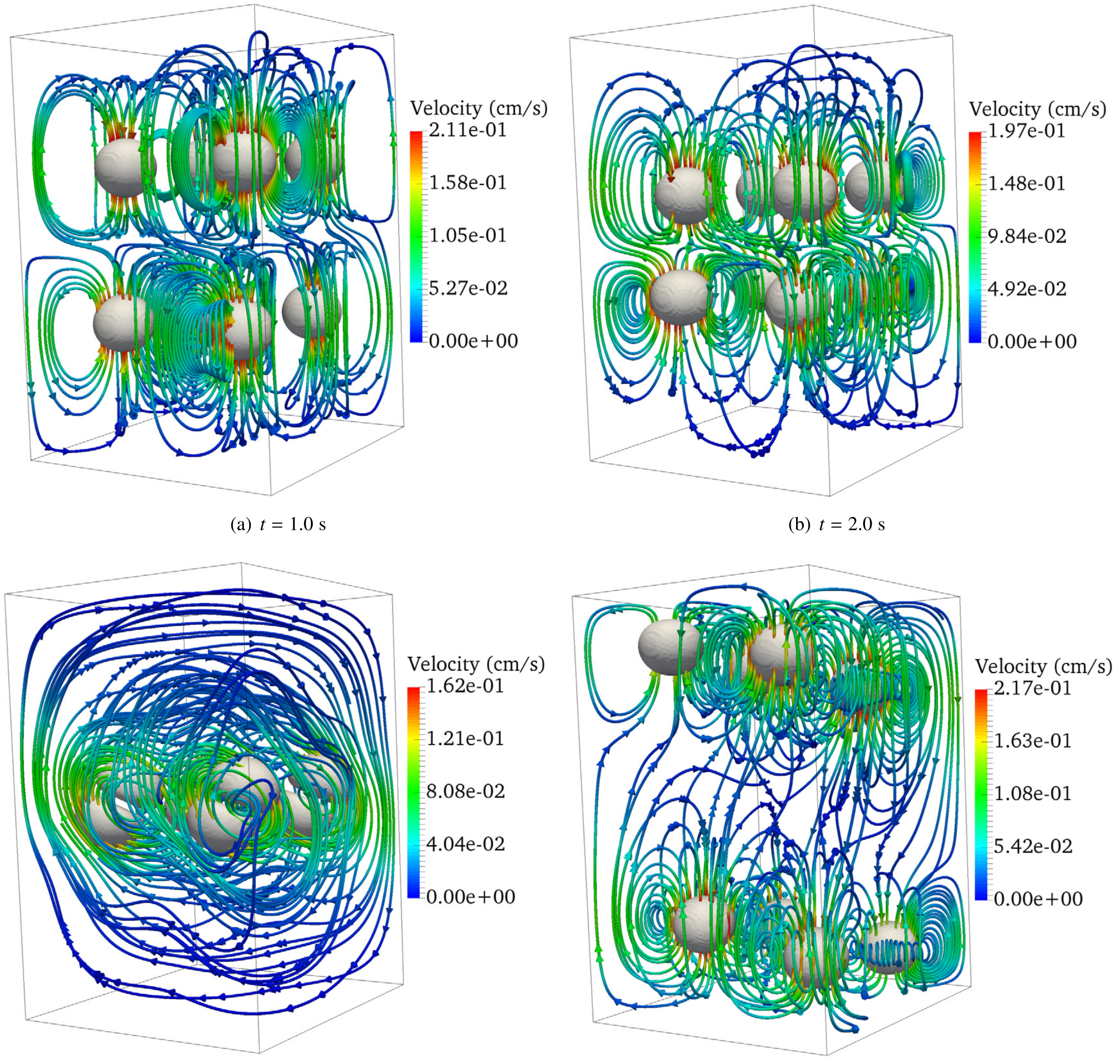


(e) L^2-norm convergence of $\boldsymbol{u}^{h^L}$

(f) H^1-norm convergence of $\boldsymbol{u}^{h^L}$

Fig. 5. Convergence studies for the hollow disk in a shear flow. (a) Inclination angle for the different discretizations detailed in Table 3. The curves using 256 × 128 or more Eulerian elements are indistinguishable at the scale of the plot. (b) The convergence rate of the Eulerian velocity in L^2 norm is 1.5. (c) The convergence rate of the Eulerian velocity in H^1 norm is 0.5. (d) The convergence rate of the Eulerian pressure in L^2 norm is close to 0.75. (e) The convergence rate of the Lagrangian displacement in L^2 norm is 1.5. (f) The convergence rate of the Lagrangian displacement in H^1 norm is 1.0.

In this example, the physical domain Ω is a parallelogram with sides $L_x = L_y = 2.0$ cm and $L_z = 3.0$ cm. The solids are hollow spheres with inner and outer diameters equal to $a = 0.12$ cm and $b = 0.44$ cm, respectively. The centers of the eight hollow spheres are $C^{S1} = (0.5\ \text{cm}, 0.5\ \text{cm}, 0.75\ \text{cm})$, $C^{S2} = (1.5\ \text{cm}, 0.5\ \text{cm}, 0.75\ \text{cm})$, $C^{S3} = (0.5\ \text{cm}, 1.5\ \text{cm}, 0.75\ \text{cm})$, $C^{S4} = (1.5\ \text{cm}, 1.5\ \text{cm}, 0.75\ \text{cm})$, $C^{S5} = (0.54\ \text{cm}, 0.54\ \text{cm}, 1.5\ \text{cm})$, $C^{S6} = (1.54\ \text{cm}, 0.54\ \text{cm}, 1.5\ \text{cm})$, $C^{S7} = (0.54\ \text{cm}, 1.54\ \text{cm}, 1.5\ \text{cm})$, and $C^{S8} = (1.54\ \text{cm}, 1.54\ \text{cm}, 1.5\ \text{cm})$. No-penetration and no-slip boundary conditions are applied at the top and bottom sides of the parallelogram. No-penetration boundary condition together with homogeneous Neumann boundary conditions in the other two directions are applied to the four lateral boundaries of the parallelogram. Both the fluid and the solid are initially at rest. The four solids at the top have density $\rho_2^{s1} = \rho_2^{s2} = \rho_2^{s3} = \rho_2^{s4} = 1.5$ g/cm^3 and the four solids at the bottom have density $\rho_2^{s5} = \rho_2^{s6} = \rho_2^{s7} = \rho_2^{s8} = 0.5$ g/cm^3 while the fluid density is $\rho_1 = 1.0$ g/cm^3. The



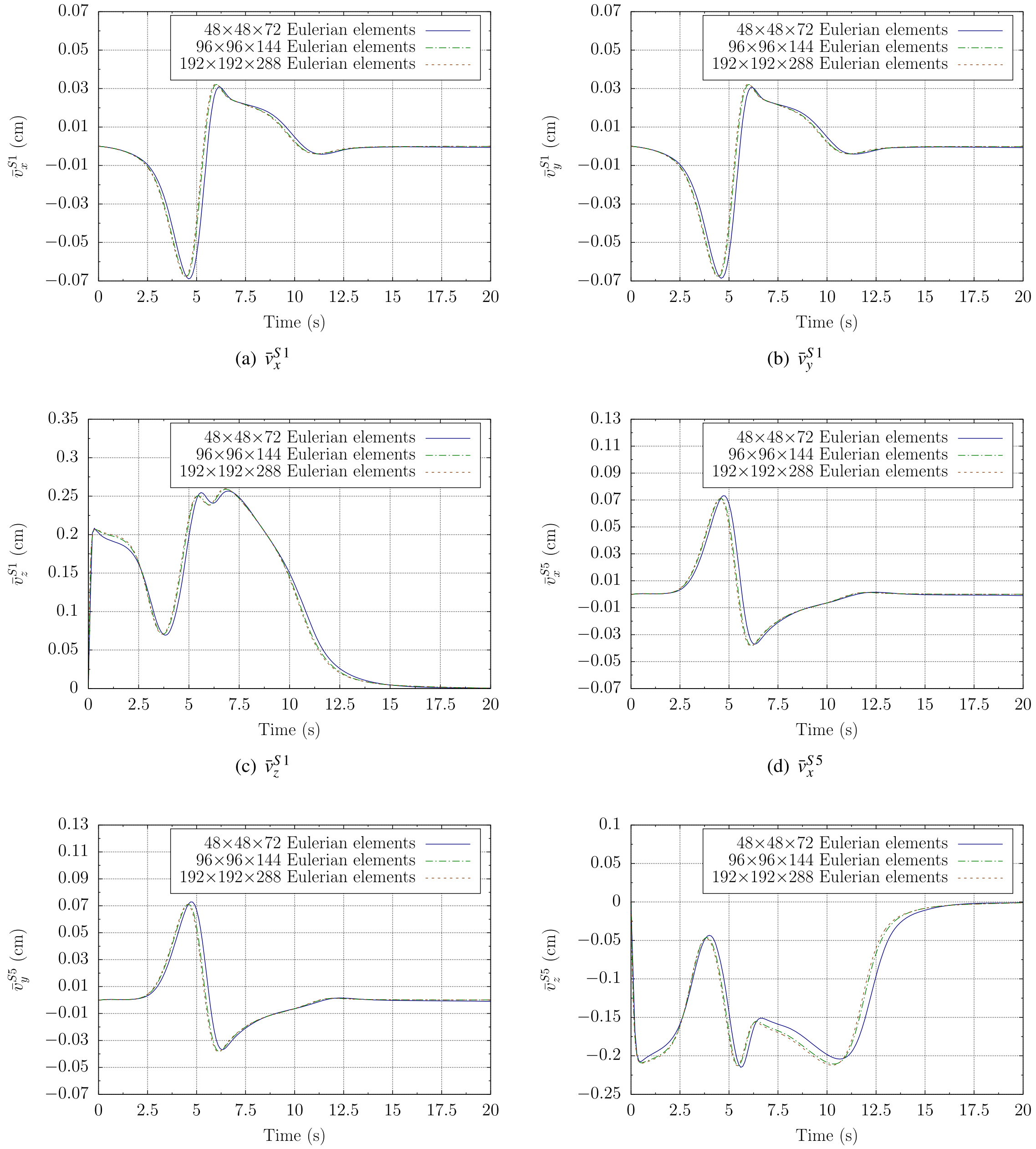
(a) $t = 1.0$ s (b) $t = 2.0$ s

(c) $t = 4.0$ s (d) $t = 10.0$ s

Fig. 6. Hollow spheres in a gravity-driven flow. The flow streamlines and the deformed geometry of the solids are plotted at different times. The gravity direction is the vertical direction and downwards. The flow patterns have a high complexity. The DCIB method, which is fully-implicit in time, is able to accurately impose the no-penetration and no-slip conditions at the fluid–solid interface. This prevents the solids from overlapping with each other without the need of adding *ad hoc* terms or contact theory to the FSI formulation.

remaining physical parameters that define this problem are the following: $\mu_1 = \mu_2 = 0.1$ g/(cm · s), $G = 1.0$ g/(cm · s^2), $\mathbf{g}_M = (0.0\ \text{cm/s}^2, 0.0\ \text{cm/s}^2, -9.81\ \text{cm/s}^2)$, and $\mathbf{g}_V = (0.0\ \text{g/(cm}^2 \cdot \text{s}^2), 0.0\ \text{g/(cm}^2 \cdot \text{s}^2), 0.0\ \text{g/(cm}^2 \cdot \text{s}^2))$. The channel Reynolds number is $R_e^c = (\rho_1 v_f L_x)/\mu_1 = 5$ and the particle Reynolds number is $R_e^p = (R_e^c) b^2/L_x^2 = 0.242$, where the characteristic velocity chosen is $v_f = 0.25$ cm/s.

The following discretization is considered to solve this problem: $48 \times 48 \times 72$ Eulerian elements with $k = 2$, $4 \times 18 \times 36$ Lagrangian elements with $p = 2$, and time step $\Delta t = 0.1$ s. After that, the discretization is refined by performing uniform h-refinement two times in each mesh and dividing the time step by two each time a new level of refinement is introduced. The final discretization has 43,807,916 EDOF and 199,800 LDOF for each solid. The use of quadratic NURBS for the Lagrangian mesh enables us to represent the hollow spheres exactly for the three discretizations considered. Figs. 6a)–d) represent some flow streamlines colored by the velocity magnitude together with the deformed geometry of the solids at different times. As shown in Fig. 6d), the solids get very close to each other, but they never contact each other or undergo small overlaps, which would have violated the kinematic interface conditions of the mathematical model detailed in Section 3. This is the case for the three discretizations considered in this example and for other two-dimensional simulations involving nearly-contact configurations that we have performed (data not shown for shortness). The enhanced accuracy with which the kinematic interface conditions are imposed in the DCIB method may be related with its fully-implicit time discretization instead of treating some of the terms in charge of imposing the kinematic interface conditions in an explicit manner as it is usually done in immersed FSI methods including the aforementioned papers [21,38,46,89,35]. This topic warrants further research.



(a) $\bar{v}_x^{S1}$ (b) $\bar{v}_y^{S1}$

(c) $\bar{v}_z^{S1}$ (d) $\bar{v}_x^{S5}$

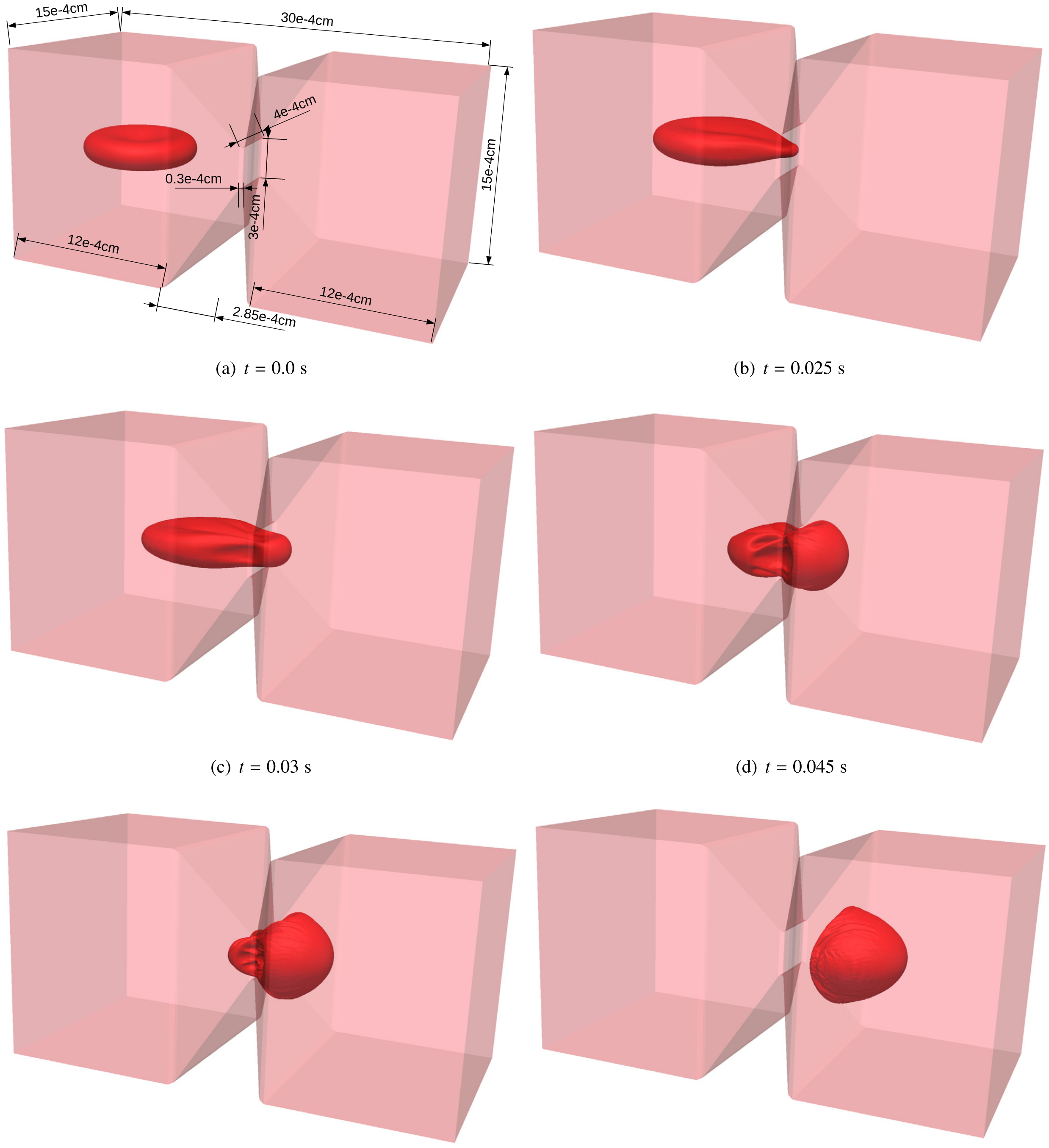
(e) $\bar{v}_y^{S5}$ (f) $\bar{v}_z^{S5}$

Fig. 7. Convergence studies for the hollow spheres in a gravity-driven flow. (a), (b), and (c) plot the time evolution of the x, y, and z components of the mean Lagrangian velocity for the S1 sphere ($\bar{\mathbf{v}}^{S1}$), respectively. (d), (e), and (f) plot the time evolution of the x, y, and z components of the mean Lagrangian velocity for the S5 sphere ($\bar{\mathbf{v}}^{S5}$), respectively.

In Figs. 7a)–f), the convergence of the mean Lagrangian velocity in each direction is plotted for solids S1 and S5 showing that the obtained results are nearly mesh independent. Regarding e_{DIV}, its value is always lower than 5.0e−7 for all discretizations considered. With respect to e_{VC}, its value goes from 4.3e−3 for the coarsest discretization to 2.8e−4 for the finest discretization.

5.5. Red blood cell passing through a constriction

The simulation of different kinds of biological cells going through small narrowings has attracted notable attention in recent years [90,25,91,92]. This kind of problem is a good way of testing how robust an immersed FSI method is when it



(a) $t = 0.0$ s (b) $t = 0.025$ s

(c) $t = 0.03$ s (d) $t = 0.045$ s

(e) $t = 0.0525$ s (f) $t = 0.0725$ s

Fig. 8. Red blood cell traversing a constriction, mimicking an interendothelial slit in the human spleen. (a) Initial position of the red blood cell and details of the constriction geometry. (b)–(f) The deformed geometry of the red blood cell is plotted at different times. The large deformations of the red blood cell are evident, which are a challenge to handle in mesh-based numerical methods.

comes to deal with large distortions of the Lagrangian mesh. Here, we consider a red blood cell in a pressure-driven flow traversing a constriction. This problem setting mimics an interendothelial slit in the human spleen, which is the smallest narrowing that red blood cells go through in the human body. As material model, we use the following incompressible hyperelastic material that has been experimentally adjusted to represent the deformations of red blood cells through the use of optical tweezers in [93]

$$\psi = \frac{G}{2}(I_1 - d) + \frac{G}{30}(I_1 - d)^3 \,, \tag{95}$$

$$\boldsymbol{S} = G\boldsymbol{I} + \frac{G}{20}(\mathrm{tr}\boldsymbol{C} - d)(\mathrm{tr}\boldsymbol{C} - d)\boldsymbol{I} \,, \tag{96}$$

where $\mathrm{tr}\boldsymbol{C}$ is the trace of $\boldsymbol{C}$.

The geometry of the physical domain Ω is described in Fig. 8a). The initial geometry of the red blood cell is chosen to be the biconcave shape that red blood cells develop at rest. This shape is defined in [94] and we use a thickness of 5.0e−7 cm. The center of gravity of the red blood cell is initially located at $C^S = (0.0006\ \mathrm{cm}, 0.00075\ \mathrm{cm}, 0.00075\ \mathrm{cm})$. Periodic boundary conditions are applied in the streamwise direction and no-penetration and no-slip boundary conditions are applied at the remaining boundaries. Both the fluid and the solid are initially at rest. The physical parameters defining this problem are the following: $\rho_1 = \rho_2 = 1.0\ \mathrm{g/cm^3}$, $\mu_1 = 0.012\ \mathrm{g/(cm \cdot s)}$, $\mu_2 = 0.0\ \mathrm{g/(cm \cdot s)}$, $G = 11000.0\ \mathrm{g/(cm \cdot s^2)}$, $\boldsymbol{g}_M = (0.0\ \mathrm{cm/s^2}, 0.0\ \mathrm{cm/s^2}, 0.0\ \mathrm{cm/s^2})$, and $\boldsymbol{g}_V = (55000.0\ \mathrm{g/(cm^2 \cdot s^2)}, 0.0\ \mathrm{g/(cm^2 \cdot s^2)}, 0.0\ \mathrm{g/(cm^2 \cdot s^2)})$.

The following discretization is considered to solve this problem: $144 \times 48 \times 48$ Eulerian elements with $k = 2$, $162 \times 1 \times 384$ Lagrangian elements with $p = 2$, which leads to 1,454,400 EDOF and 190,896 LDOF, respectively. The time step is $\Delta t = 0.000025$ s. Figs. 8b)–f) show the large deformations undergone by the red blood cell at different times as it goes through the constriction. As shown in Figs. 8b)–f), the higher inter-element continuity of NURBS gives us sufficient robustness [62] to handle this kind of simulations even for relatively coarse meshes as the ones used in this example. As in Section 5.1, the solid goes on top of the Eulerian elements in contact with the domain boundary along the simulation, which is automatically managed in our numerical method. Regarding e_{DIV} and e_{VC}, their values along the simulation are always lower than 5.0e−11 and 2.8e−3, respectively.

6. Conclusions

The DCIB method is applied to solve a widespread benchmark problem for evaluating the volume conservation of immersed FSI methods. The DCIB method leads to completely negligible incompressibility errors at the Eulerian level and various orders of magnitude of increased accuracy at the Lagrangian level compared to other immersed FSI methods. These results evince how divergence-conforming B-splines can effectively tackle a main criticism of immersed FSI methods, namely, the difficulty to accurately impose incompressibility at both the Eulerian and Lagrangian levels. We also compared the evolution of our physical quantities with the convergence studies done in [65]. Good agreement is found between the two numerical methods. The DCIB method with 4,096 Eulerian elements results in the same level of accuracy than a mesh with 1,048,576 elements in [65]. The performance of the DCIB method regarding its energy conservation is evaluated in the second example of this manuscript. For a given discretization, the DCIB method is able to conserve energy with more accuracy than the other immersed FSI methods that have solved this benchmark before. With respect to the data transfer between the Eulerian and Lagrangian discretizations, the examples show the various advantages of performing the data transfer directly at the quadrature level without using discretized Dirac delta functions, namely, the solids can get as close to the boundary of the domain as the physics of the problem dictates without any special treatment and we obtain a 0.5 increase in the convergence rates of the Eulerian velocity and the Lagrangian displacement in L^2-norm. The DCIB method is a fully-implicit method in time. As shown in the fourth example, the DCIB method is able to accurately impose the no-penetration and no-slip conditions at the interface thus avoiding solids from overlapping with each other for coarse and fine spatial discretizations and without adding any additional terms to the formulation. The robustness of the method is further enhanced by discretizing the Lagrangian displacement using NURBS, which enables to handle large mesh distortions as those encountered in the last example of this work. Finally, we would like to mention that the DCIB method is a general-purpose method for FSI problems. However, the authors believe that the DCIB method, as any other immersed FSI method, should be mainly used for problems where body-fitted FSI methods [70,95–97] cannot be applied in an automatic way or would require to remesh the fluid domain every few time steps with the associated accuracy losses. This is due to the fact that body-fitted methods, applied to FSI problems that can be solved without remeshing and without getting highly distorted meshes in the fluid domain, generally lead to higher accuracy per degree of freedom than immersed methods.

7. Future work

The following directions of future research are appealing to the authors:

- Using the DCIB method to solve FSI problems that involve solid formulations posed on co-dimension one manifolds such as, e.g., capsules [98], vesicles [99,100], and Kirchhoff–Love shells [101].
- Extending the DCIB method to FSI problems that require multi-patch spline meshes. Here, we explained how to construct single-patch NURBS and single-patch divergence-conforming B-spline discretizations, which enables to represent geometries that can be generated by mapping a square (a cube) in a two-dimensional (three-dimensional) setting. This is usually enough for most benchmark and academic problems, but many problems of scientific and engineering interest require multi-patch spline meshes. The construction of multi-patch NURBS discretizations with C^0 continuity at patch boundaries is conceptually straightforward. Constructing multi-patch divergence-conforming B-spline discretizations requires the use of a discontinuous Galerkin framework at patch boundaries to enforce tangential continuity in this

region while maintaining the stability and conservation properties of the single-patch divergence-conforming B-spline discretization as explained and tested in [48,49].

- Deriving the consistent tangent matrix of the DCIB method. The nonlinear convergence of our block-iterative approach, which is the standard approach in immersed FSI methods, will end up failing to converge when either very large density ratios between the fluid and the solid or highly stiff solids are considered. For the FSI applications where the aforementioned situations apply, the convergence of the system can be achieved by assembling the consistent tangent matrix of the three equations involved in IB methods, that is, performing a monolithic/fully-coupled linearization.
- Applying the DCIB method to problems with high Reynolds number. The problems solved in this work have low and moderate Reynolds numbers. In order to apply the DCIB method to high Reynolds number problems, one can either still use the Galerkin method in conjunction with sufficiently refined Eulerian meshes or stabilize the convective term to use coarser Eulerian meshes. In [102], it was shown how, even for highly coarse meshes, divergence-conforming B-splines combined with either the Galerkin method or the variational multiscale (VMS) formulation give essentially the same results when applied to fluid mechanics problems with low and moderate Reynolds numbers.
- Including adaptive h-refinement capabilities at the Eulerian level to have increased resolution near the fluid–solid interface while maintaining *inf–sup* stable and *pointwise* divergence-free solutions. Potential candidates to perform this task are analysis-suitable T-splines [103] and hierarchical B-splines [104]. Analysis-suitable T-splines can be used directly to obtain stable divergence-conforming discretizations [105]. In the case of hierarchical B-splines, a subset of hierarchical B-splines needs to be defined to make sure that a stable divergence-conforming discretization is obtained. In [106], a set of local, easy-to-compute, and sufficient conditions are defined that lead to two-dimensional stable divergence-conforming hierarchical B-spline spaces. As mentioned by the authors in [106], appealing directions of future research are to generalize these local conditions to the three-dimensional setting and construct adaptive h-refinement algorithms which yield hierarchical B-spline spaces satisfying these conditions.

Acknowledgements

H. Casquero and Y. Zhang were supported in part by the PECASE Award N00014-16-1-2254 and NSF CAREER Award OCI-1149591. H. Gomez was partially supported by the European Research Council through the FP7 Ideas Starting Grant (project # 307201). C. Bona-Casas was supported by the Spanish Ministry of Economy and Competitiveness (MINECO/AEI/FEDER, UE) through project DPI2017-86610-P. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575. Specifically, it used the Bridges system, which is supported by NSF award number ACI-1445606, at the Pittsburgh Supercomputing Center (PSC). This work also used the computer resources at MareNostrum and the technical support provided by Barcelona Supercomputing Center (RES-FI-2018-1-0026). Finally, we would like to thank Dr. Sugiyama for facilitating us the data that we have reproduced in this article in order to make comparisons.

References

[1] C. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.
[2] C. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (3) (1977) 220–252.
[3] C. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[4] D. Boffi, L. Gastaldi, L. Heltai, C.S. Peskin, On the hyper-elastic formulation of the immersed boundary method, Comput. Methods Appl. Mech. Eng. 197 (25) (2008) 2210–2231.
[5] D. Boffi, L. Gastaldi, A finite element approach for the immersed boundary method, Comput. Struct. 81 (8) (2003) 491–501.
[6] L. Zhang, A. Gerstenberger, X. Wang, W.K. Liu, Immersed finite element method, Comput. Methods Appl. Mech. Eng. 193 (2004) 2051–2067.
[7] W.K. Liu, Y. Liu, D. Farrell, L. Zhang, X. Wang, Y. Fukui, N. Patankar, Y. Zhang, C. Bajaj, J. Lee, J. Hong, X. Chen, H. Hsu, Immersed finite element method and its applications to biological systems, Comput. Methods Appl. Mech. Eng. 195 (2006) 1722–1749.
[8] C. Wang, J.D. Eldredge, Strongly coupled dynamics of fluids and rigid-body systems with the immersed boundary projection method, J. Comput. Phys. 295 (2015) 87–113.
[9] B. Kallemov, A. Bhalla, B. Griffith, A. Donev, An immersed boundary method for rigid bodies, Commun. Appl. Math. Comput. Sci. 11 (1) (2016) 79–141.
[10] D. Boffi, N. Cavallini, L. Gastaldi, Finite element approach to immersed boundary method with different fluid and solid densities, Math. Models Methods Appl. Sci. 21 (12) (2011) 2523–2550.
[11] T.G. Fai, B.E. Griffith, Y. Mori, C.S. Peskin, Immersed boundary method for variable viscosity and variable density problems using fast constant-coefficient linear solvers I: numerical method and results, SIAM J. Sci. Comput. 35 (5) (2013) B1132–B1161.
[12] J. Du, R.D. Guy, A.L. Fogelson, An immersed boundary method for two-fluid mixtures, J. Comput. Phys. 262 (2014) 231–243.
[13] Y. Guo, C.-Y. Wu, C. Thornton, Modeling gas–particle two-phase flows with complex and moving boundaries using DEM–CFD with an immersed boundary method, AIChE J. 59 (4) (2013) 1075–1087.
[14] B.E. Griffith, X. Luo, Hybrid finite difference/finite element immersed boundary method, Int. J. Numer. Methods Biomed. Eng. 33 (12) (2017) 1–31.
[15] H. Casquero, C. Bona-Casas, H. Gomez, A NURBS-based immersed methodology for fluid–structure interaction, Comput. Methods Appl. Mech. Eng. 284 (2015) 943–970.
[16] H. Casquero, L. Liu, C. Bona-Casas, Y. Zhang, H. Gomez, A hybrid variational-collocation immersed method for fluid–structure interaction using unstructured T-splines, Int. J. Numer. Methods Eng. 105 (11) (2016) 855–880.
[17] J. Sigüenza, S. Mendez, D. Ambard, F. Dubois, F. Jourdan, R. Mozul, F. Nicoud, Validation of an immersed thick boundary method for simulating fluid–structure interactions of deformable membranes, J. Comput. Phys. 322 (2016) 723–746.
[18] F.-B. Tian, H. Luo, L. Zhu, J.C. Liao, X.-Y. Lu, An efficient immersed boundary-lattice Boltzmann method for the hydrodynamic interaction of elastic filaments, J. Comput. Phys. 230 (19) (2011) 7266–7283.
[19] Z. Li, J. Favier, U. D'Ortona, S. Poncet, An immersed boundary-lattice Boltzmann method for single- and multi-component fluid flows, J. Comput. Phys. 304 (2016) 424–440.

[20] B.E. Griffith, Immersed boundary model of aortic heart valve dynamics with physiological driving and loading conditions, Int. J. Numer. Methods Biomed. Eng. 28 (3) (2012) 317–345.

[21] I. Borazjani, Fluid–structure interaction, immersed boundary-finite element method simulations of bio-prosthetic heart valves, Comput. Methods Appl. Mech. Eng. 257 (2013) 103–116.

[22] A. Gilmanov, T.B. Le, F. Sotiropoulos, A numerical approach for simulating fluid structure interaction of flexible thin shells undergoing arbitrarily large deformations in complex domains, J. Comput. Phys. 300 (2015) 814–843.

[23] H. Gao, L. Feng, N. Qi, C. Berry, B. Griffith, X. Luo, A coupled mitral valve-left ventricle model with fluid–structure interaction, arXiv preprint, arXiv:1704.01960.

[24] D. Cordasco, P. Bagchi, Dynamics of red blood cells in oscillating shear flow, J. Fluid Mech. 800 (2016) 484–516.

[25] H. Casquero, C. Bona-Casas, H. Gomez, NURBS-based numerical proxies for red blood cells and circulating tumor cells in microscale blood flow, Comput. Methods Appl. Mech. Eng. 316 (2017) 646–667.

[26] P. Balogh, P. Bagchi, A computational approach to modeling cellular-scale blood flow in complex geometry, J. Comput. Phys. 334 (2017) 280–307.

[27] E. Lushi, C.S. Peskin, Modeling and simulation of active suspensions containing large numbers of interacting micro-swimmers, Comput. Struct. 122 (2013) 239–248.

[28] A.P. Hoover, B.E. Griffith, L.A. Miller, Quantifying performance in the medusan mechanospace with an actively swimming three-dimensional jellyfish model, J. Fluid Mech. 813 (2017) 1112–1155.

[29] M. Ge, K. Chua, C. Shu, W. Yang, Analytical and numerical study of tissue cryofreezing via the immersed boundary method, Int. J. Heat Mass Transf. 83 (2015) 1–10.

[30] W.-X. Huang, C.B. Chang, H.J. Sung, Three-dimensional simulation of elastic capsules in shear flow by the penalty immersed boundary method, J. Comput. Phys. 231 (8) (2012) 3340–3364.

[31] W.-F. Hu, Y. Kim, M.-C. Lai, An immersed boundary method for simulating the dynamics of three-dimensional axisymmetric vesicles in Navier–Stokes flows, J. Comput. Phys. 257 (2014) 670–686.

[32] T. Kempe, J. Fröhlich, An improved immersed boundary method with direct forcing for the simulation of particle laden flows, J. Comput. Phys. 231 (9) (2012) 3663–3684.

[33] A. Calderer, S. Kang, F. Sotiropoulos, Level set immersed boundary method for coupled simulation of air/water interaction with complex floating structures, J. Comput. Phys. 277 (2014) 201–227.

[34] K.J. Galvin, A. Linke, L.G. Rebholz, N.E. Wilson, Stabilizing poor mass conservation in incompressible flow problems with large irrotational forcing and application to thermal convection, Comput. Methods Appl. Mech. Eng. 237 (2012) 166–176.

[35] R. Glowinski, T. Pan, T. Hesla, D. Joseph, J. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow, J. Comput. Phys. 169 (2) (2001) 363–426.

[36] F.P. Baaijens, A fictitious domain/mortar element method for fluid–structure interaction, Int. J. Numer. Methods Fluids 35 (7) (2001) 743–761.

[37] R. Van Loon, P.D. Anderson, J. De Hart, F.P. Baaijens, A combined fictitious domain/adaptive meshing method for fluid–structure interaction in heart valves, Int. J. Numer. Methods Fluids 46 (5) (2004) 533–544.

[38] D. Kamensky, M.-C. Hsu, D. Schillinger, J.A. Evans, A. Aggarwal, Y. Bazilevs, M.S. Sacks, T.J.R. Hughes, An immersogeometric variational framework for fluid–structure interaction: application to bioprosthetic heart valves, Comput. Methods Appl. Mech. Eng. 284 (2015) 1005–1053.

[39] D. Kamensky, J.A. Evans, M.-C. Hsu, Y. Bazilevs, Projection-based stabilization of interface Lagrange multipliers in immersogeometric fluid–thin structure interaction analysis, with application to heart valve modeling, Comput. Math. Appl. 74 (9) (2017) 2068–2088.

[40] C.S. Peskin, B.F. Printz, Improved volume conservation in the computation of flows with immersed elastic boundaries, J. Comput. Phys. 105 (1) (1993) 33–46.

[41] B.E. Griffith, On the volume conservation of the immersed boundary method, Commun. Comput. Phys. 12 (2) (2012) 401–432.

[42] L. Boilevin-Kayl, M.A. Fernández, J.-F. Gerbeau, Numerical methods for immersed FSI with thin-walled structures, Comput. Fluids (2018), https://doi.org/10.1016/j.compfluid.2018.05.024.

[43] X. Wang, L.T. Zhang, Interpolation functions in the immersed boundary and finite element methods, Comput. Mech. 45 (4) (2010) 321–334.

[44] F. Alauzet, B. Fabrèges, M.A. Fernández, M. Landajuela, Nitsche-XFEM for the coupling of an incompressible fluid with immersed thin-walled structures, Comput. Methods Appl. Mech. Eng. 301 (2016) 300–335.

[45] Y. Bao, A. Donev, B.E. Griffith, D.M. McQueen, C.S. Peskin, An immersed boundary method with divergence-free velocity interpolation and force spreading, J. Comput. Phys. 347 (2017) 183–206.

[46] D. Kamensky, M.-C. Hsu, Y. Yu, J.A. Evans, M.S. Sacks, T.J. Hughes, Immersogeometric cardiovascular fluid–structure interaction analysis with divergence-conforming B-splines, Comput. Methods Appl. Mech. Eng. 314 (2017) 408–472.

[47] A. Buffa, J. Rivas, G. Sangalli, R. Vázquez, Isogeometric discrete differential forms in three dimensions, SIAM J. Numer. Anal. 49 (2) (2011) 818–844.

[48] J.A. Evans, T.J. Hughes, Isogeometric divergence-conforming B-splines for the Darcy–Stokes–Brinkman equations, Math. Models Methods Appl. Sci. 23 (04) (2013) 671–741.

[49] J.A. Evans, T.J. Hughes, Isogeometric divergence-conforming B-splines for the steady Navier–Stokes equations, Math. Models Methods Appl. Sci. 23 (08) (2013) 1421–1478.

[50] J.A. Evans, T.J. Hughes, Isogeometric divergence-conforming B-splines for the unsteady Navier–Stokes equations, J. Comput. Phys. 241 (2013) 141–167.

[51] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Eng. 194 (2005) 4135–4195.

[52] L. Piegl, W. Tiller, The NURBS Book, Springer Science & Business Media, 2012.

[53] D.N. Arnold, R.S. Falk, R. Winther, Finite element exterior calculus, homological techniques, and applications, Acta Numer. 15 (2006) 1–155.

[54] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, Isogeometric Analysis Toward Integration of CAD and FEA, Wiley, 2009.

[55] M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M. Wu, J. Mineroff, A. Reali, Y. Bazilevs, M. Sacks, Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models, Comput. Mech. 55 (6) (2015) 1211–1225.

[56] L. Heltai, J. Kiendl, A. DeSimone, A. Reali, A natural framework for isogeometric fluid–structure interaction based on BEM–shell coupling, Comput. Methods Appl. Mech. Eng. 316 (2017) 522–546.

[57] J. Maestre, J. Pallares, I. Cuesta, M.A. Scott, A 3D isogeometric BE–FE analysis with dynamic remeshing for the simulation of a deformable particle in shear flows, Comput. Methods Appl. Mech. Eng. 326 (2017) 70–101.

[58] Y. Bazilevs, G. Moutsanidis, J. Bueno, K. Kamran, D. Kamensky, M. Hillman, H. Gomez, J. Chen, A new formulation for air-blast fluid–structure interaction using an immersed approach: part II – coupling of IGA and meshfree discretizations, Comput. Mech. (2017) 1–16.

[59] C. Kadapa, W. Dettmer, D. Perić, A fictitious domain/distributed Lagrange multiplier based fluid–structure interaction scheme with hierarchical B-spline grids, Comput. Methods Appl. Mech. Eng. 301 (2016) 1–27.

[60] C. Kadapa, W. Dettmer, D. Perić, A stabilised immersed framework on hierarchical b-spline grids for fluid–flexible structure interaction with solid–solid contact, Comput. Methods Appl. Mech. Eng. 335 (2018) 472–489.

[61] I. Akkerman, Y. Bazilevs, V.M. Calo, T.J.R. Hughes, S. Hulshoff, The role of continuity in residual-based variational multiscale modeling of turbulence, Comput. Mech. 41 (2008) 371–378.

[62] S. Lipton, J. Evans, Y. Bazilevs, T. Elguedj, T.J.R. Hughes, Robustness of isogeometric structural discretizations under severe mesh distortion, Comput. Methods Appl. Mech. Eng. 199 (2010) 357–373.

[63] D. Boffi, L. Gastaldi, L. Heltai, Numerical stability of the finite element immersed boundary method, Math. Models Methods Appl. Sci. 17 (10) (2007) 1479–1505.

[64] D. Boffi, L. Gastaldi, L. Heltai, On the CFL condition for the finite element immersed boundary method, Comput. Struct. 85 (11) (2007) 775–783.

[65] K. Sugiyama, S. Ii, S. Takeuchi, S. Takagi, Y. Matsumoto, A full Eulerian finite difference approach for solving fluid–structure coupling problems, J. Comput. Phys. 230 (3) (2011) 596–627.

[66] D. Boffi, F. Brezzi, M. Fortin, et al., Mixed Finite Element Methods and Applications, vol. 44, Springer, 2013.

[67] F. Auricchio, D. Boffi, L. Gastaldi, A. Lefieux, A. Reali, A study on unfitted 1D finite element methods, Comput. Math. Appl. 68 (12) (2014) 2080–2102.

[68] J. Chung, G. Hulbert, A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method, J. Appl. Mech. 60 (2) (1993) 371–375.

[69] K. Jansen, C. Whiting, G. Hulbert, Generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method, Comput. Methods Appl. Mech. Eng. 190 (2000) 305–319.

[70] Y. Bazilevs, K. Takizawa, T.E. Tezduyar, Computational Fluid–Structure Interaction: Methods and Applications, John Wiley & Sons, 2012.

[71] L. Dalcin, N. Collier, P. Vignal, A. Côrtes, V. Calo, PetIGA: a framework for high-performance isogeometric analysis, Comput. Methods Appl. Mech. Eng. 308 (2016) 151–181.

[72] A.F. Sarmiento, A.M. Côrtes, D. Garcia, L. Dalcin, N. Collier, V.M. Calo, PetIGA-MF: a multi-field high-performance toolbox for structure-preserving B-splines spaces, J. Comput. Sci. 18 (2017) 117–131.

[73] A. Côrtes, L. Dalcin, A. Sarmiento, N. Collier, V. Calo, A scalable block-preconditioning strategy for divergence-conforming B-spline discretizations of the Stokes problem, Comput. Methods Appl. Mech. Eng. 316 (2017) 839–858.

[74] L. Espath, A. Sarmiento, P. Vignal, B. Varga, A. Cortes, L. Dalcin, V. Calo, Energy exchange analysis in droplet dynamics via the Navier–Stokes–Cahn–Hilliard model, J. Fluid Mech. 797 (2016) 389–430.

[75] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, D.A. May, L.C. McInnes, K. Rupp, B.F. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc Web page, http://www.mcs.anl.gov/petsc, 2017.

[76] U.M. Yang, et al., BoomerAMG: a parallel algebraic multigrid solver and preconditioner, Appl. Numer. Math. 41 (1) (2002) 155–177.

[77] M.W. Gee, C.M. Siefert, J.J. Hu, R.S. Tuminaro, M.G. Sala, ML 5.0 Smoothed Aggregation User's Guide, Tech. rep., SAND2006-2649, Sandia National Laboratories, 2006.

[78] P.R. Brune, M.G. Knepley, B.F. Smith, X. Tu, Composing scalable nonlinear algebraic solvers, SIAM Rev. 57 (4) (2015) 535–565.

[79] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.

[80] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G.D. Peterson, et al., XSEDE: accelerating scientific discovery, Comput. Sci. Eng. 16 (5) (2014) 62–74.

[81] H. Zhao, J.B. Freund, R.D. Moser, A fixed-mesh method for incompressible flow–structure systems with finite solid deformations, J. Comput. Phys. 227 (6) (2008) 3114–3140.

[82] S. Roy, L. Heltai, F. Costanzo, Benchmarking the immersed finite element method for fluid–structure interaction problems, Comput. Math. Appl. 69 (10) (2015) 1167–1188.

[83] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Comput. Methods Appl. Mech. Eng. 197 (2007) 173–201.

[84] F. Auricchio, L. Beirao Da Veiga, T.J.R. Hughes, A. Reali, G. Sangalli, Isogeometric collocation methods, Math. Models Methods Appl. Sci. 20 (2010) 2075–2107.

[85] H. Casquero, L. Liu, Y. Zhang, A. Reali, H. Gomez, Isogeometric collocation using analysis-suitable T-splines of arbitrary degree, Comput. Methods Appl. Mech. Eng. 301 (2016) 164–186.

[86] Y. Wang, P.K. Jimack, M.A. Walkley, A one-field monolithic fictitious domain method for fluid–structure interactions, Comput. Methods Appl. Mech. Eng. 317 (2017) 1146–1168.

[87] B.E. Griffith, C.S. Peskin, On the order of accuracy of the immersed boundary method: higher order convergence rates for sufficiently smooth problems, J. Comput. Phys. 208 (1) (2005) 75–105.

[88] Y. Yu, D. Kamensky, M.-C. Hsu, X.Y. Lu, Y. Bazilevs, T.J. Hughes, Error Estimates for Dynamic Augmented Lagrangian Boundary Condition Enforcement, with Application to Immersogeometric Fluid–Structure Interaction, ICES REPORT 17-21, The University of Texas at Austin, 2017.

[89] C.-C. Liao, W.-W. Hsiao, T.-Y. Lin, C.-A. Lin, Simulations of two sedimenting–interacting spheres with different sizes and initial configurations using immersed boundary method, Comput. Mech. 55 (6) (2015) 1191–1200.

[90] I.V. Pivkin, Z. Peng, G.E. Karniadakis, P.A. Buffet, M. Dao, S. Suresh, Biomechanics of red blood cells in human spleen and consequences for physiology and disease, Proc. Natl. Acad. Sci. USA (2016) 201606751.

[91] J. Gounley, E.W. Draeger, A. Randles, Numerical simulation of a compound capsule in a constricted microchannel, Proc. Comput. Sci. 108 (2017) 175–184.

[92] F. Serrano-Alcalde, J.M. García-Aznar, M.J. Gómez-Benito, The role of nuclear mechanics in cell deformation under creeping flows, J. Theor. Biol. 432 (2017) 25–32.

[93] J. Mills, L. Qie, M. Dao, C. Lim, S. Suresh, et al., Nonlinear elastic and viscoelastic deformation of the human red blood cell with optical tweezers, MCB 1 (2004) 169–180.

[94] R. Skalak, A. Tozeren, R. Zarda, S. Chien, Strain energy function of red blood cell membranes, Biophys. J. 13 (3) (1973) 245–264.

[95] Y. Bazilevs, V.M. Calo, T.J.R. Hughes, Y. Zhang, Isogeometric fluid–structure interaction: theory, algorithms, and computations, Comput. Mech. 43 (2008) 3–37.

[96] J. Bueno, H. Casquero, Y. Bazilevs, H. Gomez, Three-dimensional dynamic simulation of elastocapillarity, Meccanica 53 (6) (2018) 1221–1237.

[97] K. Takizawa, T.E. Tezduyar, T. Terahara, Ram-air parachute structural and fluid mechanics computations with the space–time isogeometric analysis (ST-IGA), Comput. Fluids 141 (2016) 191–200.

[98] C. Pozrikidis, Modeling and Simulation of Capsules and Biological Cells, CRC Press, 2003.

[99] H. Zhao, A.P. Spann, E.S. Shaqfeh, The dynamics of a vesicle in a wall-bound shear flow, Phys. Fluids 23 (12) (2011) 121901.

[100] R.A. Sauer, T.X. Duong, K.K. Mandadapu, D.J. Steigmann, A stabilized finite element formulation for liquid shells and its application to lipid bilayers, J. Comput. Phys. 330 (2017) 436–466.

[101] H. Casquero, L. Liu, Y. Zhang, A. Reali, J. Kiendl, H. Gomez, Arbitrary-degree T-splines for isogeometric analysis of fully nonlinear Kirchhoff–Love shells, Comput. Aided Des. 82 (2017) 140–153.

[102] T.M. van Opstal, J. Yan, C. Coley, J.A. Evans, T. Kvamsdal, Y. Bazilevs, Isogeometric divergence-conforming variational multiscale formulation of incompressible turbulent flows, Comput. Methods Appl. Mech. Eng. 316 (2017) 859–879.

[103] L. Beirao da Veiga, A. Buffa, G. Sangalli, R. Vazquez, Analysis suitable T-splines of arbitrary degree: definition, linear independence, and approximation properties, Math. Models Methods Appl. Sci. 23 (11) (2013) 1979–2003.

[104] G. Lorenzo, M. Scott, K. Tew, T. Hughes, H. Gomez, Hierarchically refined and coarsened splines for moving interface problems, with particular application to phase-field models of prostate tumor growth, Comput. Methods Appl. Mech. Eng. 319 (2017) 515–548.

[105] A. Buffa, G. Sangalli, R. Vázquez, Isogeometric methods for computational electromagnetics: B-spline and T-spline discretizations, J. Comput. Phys. 257 (2014) 1291–1320.

[106] J.A. Evans, M.A. Scott, K. Shepherd, D. Thomas, R. Vazquez, Hierarchical B-spline complexes of discrete differential forms, arXiv preprint, arXiv:1708.04195.