A review of finite element methods for fluid-structure interactions

Yongxing Wang[1*]

[1*]School of Computer Science, University of Leeds, Leeds, UK.

Corresponding author(s). E-mail(s): scsywan@leeds.ac.uk;

Abstract

We present a comprehensive review of finite element methods (FEM) for solving fluid–structure interaction (FSI) problems. Our aim is to systematically categorise and compare these methods based on three key aspects: mesh type, coupling strategy, and the choice of solved variables. For each method, we highlight its respective advantages and limitations. In addition, we formulate and compare the finite element weak forms of the various approaches, employing Newton's method to linearise nonlinear terms at the continuous functional level. We implement six FSI methods in the open-source software package `FreeFEM++`, and compare these methods based on three benchmarks.

Keywords: FSI, Monolithic, Partitioned, One-velocity, Interface problem, ALE, FEM

1 Introduction

Numerical methods for Fluid-Structure Interaction (FSI) problems have rapidly developed over the past decades, with numerous model methods designed to address their efficiency, stability, and accuracy. Most of these methods share similar features and formulations; however, there is a lack of a comprehensive summary, review, or comparison of these approaches. Existing review articles are either outdated [1] or are focused on specific method types, such as the immersed boundary method [2, 3] or the full Eulerian method [4]. There are also review articles that focus on specific application areas, such as FSI in biology and medicine [5, 6] or sub and supersonic airflow [7]. Lecture notes [8] cover a range of FSI topics, including classical partitioned methods [9], space-time FEM methods, adaptive meshing techniques, and monolithic methods. There are many FSI books concentrating on various approaches: monolithic

approaches applied in hemodynamics and particulate flow [10]; adaptive methods [11]; space-time FEM approaches applied in diverse fields such as blood flow, parachutes, and wind turbines [12]; fully Eulerian FSI formulations addressing error estimation, adaptivity, and optimisation [13]; analysis of inverse FSI problems through shape analysis [14]; Arbitrary Lagrangian-Eulerian (ALE)-partitioned methods with stability analysis and applications in multiphase problems [15]; FSI in low-Reynolds flow with applications in studying the deformability of biological cells [16]; and two-phase FSI utilising partitioned, interface-fitted methods [17].

A comprehensive categorisation of existing FSI methods may be based on three fundamental questions: (1) What kind of mesh is used? This can be one interface-fitted mesh, one mesh without interface fitting, or two meshes. (2) Which variables are solved for? This can encompass the fluid velocity v, pressure p, and solid displacement d; or a single velocity for both fluid and solid by expressing the solid equations in terms of velocity (methods 4, 8 and 11 in Table 1); only fluid velocity and pressure, in which case the solid equations are not explicitly solved (methods 3, 7 and 10 in Table 1). (3) What type of coupling strategies are employed? This can be monolithic/fully-coupled, or partitioned/segregated. As a result, a combination of the answers to these three questions would lead to distinct types of numerical methods, as summarised in Table 1, with corresponding literature citations below.

Variables to solve:	v for fluid and d for solid		v for fluid	v for fluid and solid
Coupling strategy:	Partitioned	Monolithic	Partitioned	Monolithic
One mesh (interfaced fitted)	1. Two-way coupling	2.1 Monolithic ALE 2.2 Locally modified FEM	3. One-way coupling	4. One-velocity monolithic
Two meshes	5.1 Modified IFEM 5.2 CutFEM	6.1 FDM/DLM 6.2 CutFEM	7. IFEM	8. One-velocity FDM
One mesh (interface unfitted)	×	9.1 Fully Eulerian 9.2 CutFEM	10. classical IBM	11. Phase-field

Table 1: Various numerical methods categorised by addressing the three key questions above. ×: This combination hasn't been identified in the author's literature review. ALE: Arbitrary Lagrangian-Eulerian; IFEM: Immersed Finite Element Method; FDM: Fictitious Domain Method; DLM: Distributed Lagrange Multiplier; IBM: Immersed Boundary Method.

1. Partitioned/Segregated two-way coupling methods [12, 18–37].
2. (2.1) Monolithic/Fully-coupled ALE methods [38–42].
   (2.2) Locally modified finite element method [43–46].
3. One-way coupling FSI methods [47–49].
4. One-velocity Monolithic Eulerian methods [50–53].
5. (5.1) Modified immersed finite element methods [54].
   (5.2) CutFEM–based methods [55].
6. (6.1) Fictitious domain methods with distributed Lagrange multiplier [56–71].
   (6.2) CutFEM–based methods [72, 73].
7. Immersed finite element methods [1, 54, 62, 74–79].

8. One-velocity fictitious domain method [80–83].
9. (9.1) Fully Eulerian formulation [84–90].
   (9.2) CutFEM–based methods[91].
10. Immersed boundary methods [74, 92–111].
11. Phase-field models for fluid–structure interactions [112–114].

Remark 1. *Based on our literature review, certain methods have very limited publications. Specifically, we found four papers discussing cutFEM-based FSI methods, three of which originate from a single research group: one of these papers explores the partitioned method (Method 5.2), while the other two examine the monolithic approach (Method 6.2). The remaining cutFEM paper presents a very new variant of cutFEM (Method 9.2) based on a fully Eulerian formulation. Only one paper covers the modified IFEM (Method 5.1). Additionally, three papers are dedicated to phase-field-based approaches (Method 11).*

The classical partitioned or segregated methods (Method 1), as referenced in [12, 18–37], utilise a single interface-fitted mesh to solve for both fluid velocity and solid displacement. These methods are among the earliest numerical approaches developed to address FSI problems and are still under development, particularly in terms of parallelisation [31, 35], efficiency [32, 34], stability, and accuracy [36], as well as adaptivity [33, 36, 37]. For two-way coupling (or strong coupling) FSI problems, partitioned methods iteratively solve for the fluid and solid problems separately. They exchange information across the fluid-solid interface until a balance of variables is achieved at the interface. The advantage of classical partitioned methods is their relative ease of implementation, often based on existing fluid and solid code. However, their drawback lies in the difficulty of guaranteeing convergence during iteration, especially when there is rapid energy exchange between the fluid and solid at the interface [25, 26]. A simplified version of these two-way partitioned methods is the one-way FSI method (Method 3). In this method, only the fluid problem is solved explicitly, while the solid deforms passively by following the fluid fields. It provides an updated boundary condition for the fluid problem in every time frame [47–49].

Monolithic methods were developed to overcome the limitations of partitioned methods, which solve the fluid and solid equations within a single equation system. This approach has gained widespread acceptance as a more robust numerical method compared to classical partitioned methods. Initially, monolithic methods were based on an ALE interface-fitted mesh (Method 2.1). They solve for fluid velocity, pressure, solid displacement, and use a Lagrange multiplier to enforce continuity at the fluid-solid interface [38, 39, 41, 42].

Recent advancements in monolithic methods have introduced the use of a background Cartesian grid with locally cut triangles to accommodate the fluid-solid interface, named as the locally modified FEM (Method 2.2) [43–46]. This method solves for the variables of fluid velocity and pressure, as well as solid displacement and velocity. In comparison to traditional ALE methods, it circumvents the need for a moving mesh and is proficient in handling arbitrarily large solid deformations. However, a drawback lies in its management of the local anisotropic FEM mesh – Edge stabilisation techniques [115] are introduced to enhance the stability.

Monolithic methods have also evolved to use two meshes (Method 6.1) [40, 62, 63]. This approach addresses fluid velocity, pressure, and solid displacement while using a distributed Lagrange multiplier to ensure velocity consistency between the meshes. Another variation is the cutFEM-based FSI method (Method 6.2) [72, 73], which extends the solid foreground mesh to envelop a surrounding artificial fluid mesh with a fitted interface. This method involves cutting through local elements around the interface for numerical quadrature without modifying local degrees of freedom, unlike the locally modified method (Method 2.2). However, stability issues arise due to possible irregularities in local meshes, leading to the adoption of Nitsche-type interface coupling strategies with ghost penalty terms [116] to guarantee stability. The initial version of this cutFEM-based approach was presented in [55] as a partitioned approach (Method 5.2). A new version of the cutFEM-based FSI method (Method 9.2) is presented in [91], which uses a fully-Eulerian formulation on one mesh without interface fitting, solving for fluid pressure and velocity, solid displacement and velocity.

Alternatively, monolithic methods have been adapted to employ a single mesh without interface fitting (Method 9.1), known as the fully Eulerian formulation [84–90]. These methods use an Initial Points (IP) set to capture the fluid-structure interface, extending the unknown variables across the entire domain with a characteristic function. The FSI system is solved in a monolithic manner.

Recently, we developed a one-velocity method that solves for a single velocity field across the entire FSI domain, using either a single interface-fitted mesh (Method 4) [50–53] or two meshes (Method 8) [80–83]. In this one-velocity method, the solid equations are first expressed in terms of velocity, automatically ensuring velocity consistency at the fluid-solid interface with one interface-fitted mesh. When using two meshes, finite element isoparametric interpolation is employed to maintain velocity consistency [53, 80]. This interpolation is applied on both sides of the equations, which leads to a modification of both the matrix and the right-hand-side vector of the final discretised system derived from the background fluid problem.

The immersed boundary method (Method 10) [74, 92–94, 96, 98, 100–106, 117] was originally designed to address complex geometric configurations without encountering the challenges of interface-fitted meshes. It effectively solves a fluid equation on a static background mesh using finite difference, finite element methods, or, more recently, the Lattice Boltzmann method. This involves incorporating a singular boundary forcing term at the right-hand side of the fluid equation to model the influence of an elastic or rigid solid. The force is typically evaluated based on the curvature of the solid boundary, which is then distributed to the background mesh using smooth bell-shaped functions to alleviate singularity. The classical IBM started as an explicit partitioned method and gradually evolved to include implicit schemes [118, 119] as well as monolithic schemes [117, 120, 121]. IBM has also been extended to immersed interfaces to consider sharp interfaces [105, 122–131], level-set IBM [107–111], finite element IBM [95, 97], and isogeometric IBM [132–134]. Additionally, there is the immersed boundary-Lattice Boltzmann method [119, 135–143].

The immersed finite element methods (Method 7) [1, 54, 62, 75–79, 144–147] were developed from the IBM to enable the modeling of various properties of a volumetric solid. They share the same concept of modeling the solid as a forcing term while

effectively solving a fluid equation. Instead of evaluating the force from the fluid boundary, as IBMs do, IFEMs calculate the force from a separate solid mesh. The classical IFEMs are explicit, meaning the solid force is computed based on the previous solid state of the FSI system. There is also a modified IFEM (Method 5.1) [54] that solves the solid equations and provides feedback in a loop. However, accurately prescribing a boundary condition for the solid problem is challenging.

The phase-field models were originally developed to address the free boundary between two-phase flows [148, 149], introducing a phase variable to smear out parameter jumps across the interface between two fluids. Effectively, a smooth interface is created between the two phases through the use of this phase variable. When applying this concept to FSI problems (Method 11), the phase variable not only needs to smear out parameter jumps, such as densities of the fluid and solid, but also needs to smear out the constitutive equations. The related publications are limited; in [112, 113], the Cauchy stress tensor of the solid is expressed in terms of velocity, combined with the fluid viscous stress tensor by the phase variable.

Remark 2. *For all two-mesh methods, it is reasonable to assume that both the fluid and solid are either incompressible or compressible to ensure the consistency of velocity on meshes. For example, if the fluid velocity is divergence-free, the velocity interpolated from the fluid to the solid mesh must also be divergence-free.*

Remark 3. *Monolithic methods traditionally use Lagrange multipliers to enforce the consistency of variables at the fluid-solid interface. When applying monolithic methods to two meshes, it becomes natural to use a distribution Lagrange multiplier to ensure the consistency between the two meshes when solving for both velocity and displacement [63]. However, interpolation becomes a natural choice when solving for only a single velocity field [80].*

The paper is structured as follows: Section 2 introduces the fundamental partial differential equations (PDEs) governing the FSI problems, along with a presentation of two hyperelastic solid models. Subsequently, a comprehensive overview of numerical methods, categorised in Table 1, is provided, encompassing interface-fitted methods (Section 3), two-mesh methods (Section 4), and interface-unfitted methods (Section 5). Three numerical examples are presented in Section 6 to compare six FSI methods. Applications are discussed in Section 7, with conclusions drawn in Section 8.

2 Governing partial differential equations

We consider a general fluid-structure interaction problem, where the fluid occupies the domain $\Omega_t^f \subset \mathbb{R}^d$, and the solid occupies the domain $\Omega_t^s \subset \mathbb{R}^d$, as shown in Figure 1 at time t, with $d = 2, 3$ being the dimension. Here, superscripts f and s denote the fluid and solid respectively, while the subscript t indicates domain is time-dependent. The fixed domain $\Omega = \Omega_t^f \cup \Omega_t^s$ comprises both the fluid and solid regions, having an outer boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$, where Dirichlet and Neumann boundary conditions are imposed on Γ_D and Γ_N respectively. The boundary $\Gamma_t = \partial\Omega_t^f \cap \partial\Omega_t^s$ represents the moving interface between the fluid and solid. In consideration of large solid deformations in this paper, we denote the reference (material) coordinates of the solid as $\mathbf{X}$ and the current coordinates as $\mathbf{x}$.

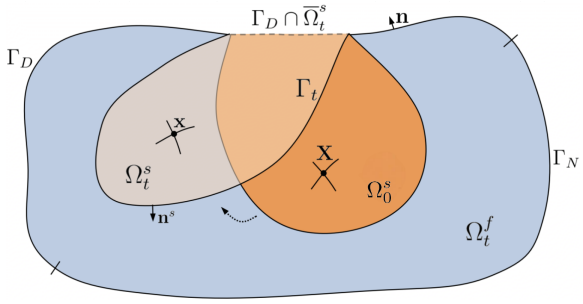

Fig. 1: Schematic diagram of FSI, where $\Omega = \Omega_t^f \cup \Omega_t^s$ and $\partial\Omega = \Gamma_D \cup \Gamma_N$.

By introducing an indicator function $1_\omega(\mathbf{x}) = 1$ if $\mathbf{x} \in \omega$ and $1_\omega(\mathbf{x}) = 0$ otherwise, we express $\rho = \rho^f 1_{\Omega_t^f} + \rho^s 1_{\Omega_t^s}$, $\mathbf{u} = \mathbf{u}^f 1_{\Omega_t^f} + \mathbf{u}^s 1_{\Omega_t^s}$, and $\boldsymbol{\sigma} = \boldsymbol{\sigma}^f 1_{\Omega_t^f} + \boldsymbol{\sigma}^s 1_{\Omega_t^s}$, representing the density, velocity vector, and Cauchy stress tensor, respectively. Additionally, let $\mathbf{g}$ denotes the acceleration due to gravity, the conservation of momentum takes the same form for both the fluid and solid, differing only in the specific expression of the constitutive equations:

$$\rho \mathcal{D}_t \mathbf{u} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g}, \tag{1}$$

where $\mathcal{D}_t(\cdot)$ represents the total derivative. In our paper, we consider two solid models: incompressible and compressible. (i) for an incompressible solid, the following continuity equation is satisfied for both the fluid and solid velocity:

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in} \quad \Omega. \tag{2}$$

(ii) For a compressible solid model, the continuity equation is only required in the fluid domain:

$$\nabla \cdot \mathbf{u}^f = 0 \quad \text{in} \quad \Omega_t^f. \tag{3}$$

The constitutive equation for a Newtonian fluid is described as follows:

$$\boldsymbol{\sigma}^f = \mu^f \mathrm{D}\mathbf{u}^f - p^f \mathbf{I}, \tag{4}$$

where operator $\mathrm{D}(\cdot) = \nabla(\cdot) + \nabla^\top(\cdot)$. Here, μ^f is the dynamic viscosity of the fluid, and p^f is the fluid pressure. Before introducing the constitutive equations for the solid in the subsequent sections, we complement the FSI system with the following boundary and initial conditions.

Interface boundary conditions:

$$\mathbf{u}^f = \mathbf{u}^s \quad \text{on} \quad \Gamma_t, \tag{5}$$

$$\boldsymbol{\sigma}^f \cdot \mathbf{n}^s = \boldsymbol{\sigma}^s \cdot \mathbf{n}^s \quad \text{on} \quad \Gamma_t. \tag{6}$$

Dirichlet and Neumann boundary conditions for the fluid:

$$\mathbf{u}^f = \bar{\mathbf{u}} \quad \text{on} \quad \Gamma_D, \tag{7}$$

$$\boldsymbol{\sigma}^f \cdot \mathbf{n} = \bar{\mathbf{h}} \quad \text{on} \quad \Gamma_N. \tag{8}$$

Finally, the initial conditions:

$$\mathbf{u}^f\big|_{t=0} = \mathbf{u}_0^f \quad \text{in} \quad \Omega_0^f, \tag{9}$$

$$\mathbf{u}^s|_{t=0} = \mathbf{u}_0^s \quad \text{in} \quad \Omega_0^s. \tag{10}$$

.

Remark 4. *Classically, the displacement vector* $\mathbf{d}$ *is used to describe the solid's motion instead of the velocity vector* $\mathbf{u}^s$*. In this paper, both of them are used depending on the numerical method. The relation between* $\mathbf{d}$ *and* $\mathbf{u}^s$ *is generally expressed as:*

$$\mathbf{u}^s = \mathcal{D}_t\mathbf{d} = \partial_t\mathbf{d} + ((\mathbf{u}^s - \mathbf{w}) \cdot \nabla)\,\mathbf{d}, \tag{11}$$

where $\mathbf{w}$ *is the velocity of the moving frame used to describe the solid. For the Lagrangian frame,* $\mathbf{w} = \mathbf{u}^s$*, thus* $\mathcal{D}_t\mathbf{d} = \partial_t\mathbf{d}$*, as adopted in Sections 3.1 (Method 1) and 3.2 (Method 2.1), and in all the two-mesh methods in Section 4, where the solid is moved by the its own velocity. For the Eulerian frame,* $\mathbf{w} = 0$*, and therefore* $\mathcal{D}_t\mathbf{d} = \partial_t\mathbf{d} + (\mathbf{u}^s \cdot \nabla)\,\mathbf{d}$*, as adopted in the methods using one mesh without interface fitting in Section 5. The solid is moved by an arbitrary velocity* $\mathbf{w}$ *in the one-velocity monolithic method in Section 3.3. Note that the total derivative of velocity in equation (1) is generally expressed as:*

$$\mathcal{D}_t\mathbf{u} = \partial_t\mathbf{u} + ((\mathbf{u} - \mathbf{w}) \cdot \nabla)\,\mathbf{u}. \tag{12}$$

2.1 Compressible Saint-Venant-Kirchhoff solid model

The energy function for the Venant-Kirchhoff solid model is expressed as follows [150, 151]:

$$\Psi(\mathbf{F}) = \mu^s \mathrm{tr}(\mathbf{E}^2) + \frac{\lambda^s}{2}\mathrm{tr}^2(\mathbf{E}), \tag{13}$$

with μ^s and λ^s being the Lamé constants, and

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{F}^\top\mathbf{F} - \mathbf{I}\right) \tag{14}$$

represents the Lagrangian Green strain, where $\mathbf{F} = \nabla_{\mathbf{X}}\mathbf{d} + \mathbf{I}$ denotes the deformation tensor and $\mathbf{I}$ the identity tensor. Therefore, $\mathbf{E}$ can also be expressed as a function of the displacement:

$$\mathbf{E}(\mathbf{d}) = \frac{1}{2}\left(\mathrm{D}_{\mathbf{X}}\mathbf{d} + \nabla_{\mathbf{X}}^\top\mathbf{d}\nabla_{\mathbf{X}}\mathbf{d}\right), \tag{15}$$

where operator $\mathrm{D}_{\mathbf{X}}(\cdot) = \nabla_{\mathbf{X}}^\top(\cdot) + \nabla_{\mathbf{X}}(\cdot)$ acting on the reference configuration, as compared to $\mathrm{D}(\cdot)$, which operates on the current configuration. The first Piola-Kirchhoff stress tensor can then be expressed as [81]:

$$\mathbf{P} = \partial_{\mathbf{F}} \Psi(\mathbf{F}) = \mathbf{F}\left(2\mu^s \mathbf{E} + \lambda^s \mathrm{tr}(\mathbf{E})\mathbf{I}\right). \tag{16}$$

Notice that

$$\mathbf{S} = 2\mu^s \mathbf{E} + \lambda^s \mathrm{tr}(\mathbf{E})\mathbf{I} \tag{17}$$

is the second Piola-Kirchhoff stress. The relationships between the first Piola-Kirchhoff stress $\mathbf{P}$, the second Piola-Kirchhoff stress $\mathbf{S}$, and the Cauchy stress $\boldsymbol{\sigma}$ are as follows: $\mathbf{P} = \mathbf{FS}$, $\boldsymbol{\sigma}^s = J^{-1}\mathbf{FSF}^\top$, where $J = \det(\mathbf{F})$ denotes the determinant of $\mathbf{F}$ [12].

2.2 Incompressible neo-Hookean solid model

Starting from the energy function [59, 152]:

$$\Psi(\mathbf{F}) = \frac{\mu^s}{2}\left(\mathrm{tr}(\mathbf{FF}^\top) - d\right) - \mu^s \ln(J) + \frac{\lambda^s}{2}\ln^2(J), \tag{18}$$

for a general compressible neo-Hookean solid model, with $d = 2, 3$ being the dimension, we obtain [81]:

$$\boldsymbol{\sigma}^s = J^{-1}\partial_{\mathbf{F}}\Psi(\mathbf{F})\,\mathbf{F}^\top = \mu^s J^{-1}\left(\mathbf{FF}^\top - \mathbf{I}\right) + J^{-1}\lambda^s \ln(J)\mathbf{I}. \tag{19}$$

For the incompressible case, $J^{-1}\lambda^s \ln(J)$ is replaced by a pressure term $-p^s$ and use $J = 1$ or $\nabla \cdot \mathbf{u}^s = 0$ (equivalent based on Jacobi's formula [153]) to determine p^s [57]. Therefore, the constitutive equation for an incompressible neo-Hookean model can be expressed as the sum of the deviatoric part $\boldsymbol{\tau}^s$ and the pressure:

$$\boldsymbol{\sigma}^s = \boldsymbol{\tau}^s - p^s\mathbf{I} = \mu^s J^{-1}\left(\mathbf{FF}^\top - \mathbf{I}\right) - p^s\mathbf{I}. \tag{20}$$

Drawing upon the governing partial differential equations and the aforementioned two solid models, existing numerical methods for FSI problems are presented within the finite element method framework in subsequent sections. We concentrate on the finite element weak formulation, linearisation, and the solution algorithm, rather than delving into specifics of finite element discretisation and techniques for solving the ultimate linear algebraic system. These procedures constitute standard finite element methodologies readily available in literature. However, we shall briefly touch upon time discretisation to facilitate discussions about the solution algorithm and to outline the final linear equation systems. This involves discretising the time domain $[0, T]$ into intervals: $[0, T] = \bigcup_{n=0}^{N-1} [t_n, t_{n+1}]$ with $t_0 = 0$, $t_N = T$ and time step $\Delta t = t_{n+1} - t_n$ ($t_n = n\Delta t$).

3 Methods using one interface-fitted mesh

For interface-fitted methods, the fluid and solid share discretised interface mesh points. Both the velocity variable from the fluid side and the displacement (or velocity) variable from the solid side are defined on these points, as depicted in Figure 2. Whether solving the FSI problem sequentially (as in Section 3.1) or simultaneously (as in Sections 3.2 and 3.3), the primary concern is ensuring alignment between these variables from both sides.

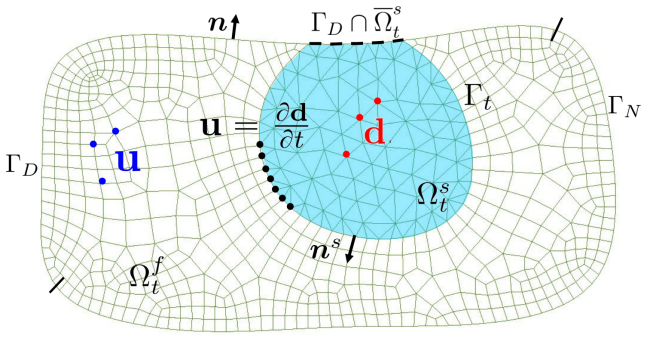

Fig. 2: Schematic diagram of an interface-fitted mesh, where $\Omega = \Omega_t^f \cup \Omega_t^s$ and $\partial\Omega = \Gamma_D \cup \Gamma_N$. Solving for the fluid velocity $\mathbf{u}^f$ in Ω_t^f and the solid displacement $\mathbf{d}$ in Ω_t^s.

For a large deformation problem, as considered in this paper, it is necessary to perform remeshing and/or use a moving mesh strategy to ensure the mesh is interface-fitted at every time step. When remeshing, the interface nodes are fixed. When moving the mesh, we adopt two strategies depending on the underlying method. The first strategy is to move all the solid nodes using their own velocity (no convection term in the solid domain; see Remark 4) and to solve the mesh equation (22) only in the fluid domain (with the interface being moved by its own velocity, corresponding to boundary condition (24)) to obtain the mesh velocity field for the fluid. The second strategy is to solve the mesh equation (22) in the whole domain (with the interface being moved by its own velocity, corresponding to boundary condition (59)) to obtain the mesh velocity for both the fluid and the solid. In this second case, a convection term appears in the solid equation, as noted in Remark 4. In this section, we will employ an ALE mesh to demonstrate the methodology. In this scenario, the fluid momentum equation (1) (in Ω_t^f) can be expressed as:

$$\rho^f \partial_t \mathbf{u}^f + \rho^f \left(\left(\mathbf{u}^f - \mathbf{w} \right) \cdot \nabla \right) \mathbf{u}^f = \nabla \cdot \boldsymbol{\sigma}^f + \rho^f \mathbf{g}. \tag{21}$$

We solve the following linear elastic equation within Ω_t^f at each time step to calculate $\mathbf{w}$.

$$\nabla \cdot \left(\mu_m \mathrm{D}\mathbf{w} - \lambda_m \left(\nabla \cdot \mathbf{w} \right) \mathbf{I} \right) = 0, \tag{22}$$

where μ_m and λ_m represent the Lamé constants selected based on the solid parameters (although alternative values could be chosen). The boundary conditions are established as follows.

$$\mathbf{w} \cdot \mathbf{n} = 0 \qquad \text{on} \quad \partial\Omega_t^f, \tag{23}$$

and

$$\mathbf{w} = \Delta\mathbf{d}/\Delta t \qquad \text{on} \quad \Gamma_t, \tag{24}$$

where $\Delta\mathbf{d} = \mathbf{d}_{n+1} - \mathbf{d}_n$ represents the solid displacement at the current step from time t_n to t_{n+1}.

3.1 Two-way partitioned/segregated methods (Method 1)

In classical partitioned or segregated methods in the literature, the solid displacement (rather than solid velocity) is typically the solved dependent variable. Consequently, it becomes convenient to express the solid momentum equation (1) in terms of displacement within Ω_t^s as:

$$\rho^s \partial_{tt}\mathbf{d} = \nabla \cdot \boldsymbol{\sigma}^s + \rho^s \mathbf{g}. \tag{25}$$

Partitioned methods solve the fluid equation (21), mesh equation (22), and the solid equation (25) sequentially, as described in **Algorithm** 1. Boundary conditions on Γ_t can be considered in two ways: one approach involves solving the fluid equation (21) using the Dirichlet boundary condition $\mathbf{u}^f = \mathbf{u}^s$, then computing the reaction force $\mathbf{h}^f$ on Γ_t after solving (21). This is followed by using the Neumann boundary condition $\boldsymbol{\sigma}^s \cdot \mathbf{n}^s = \mathbf{h}^f$ to solve the solid equation (25). Alternatively, the process can be reversed: starting by solving the solid equation (25) with the Dirichlet boundary condition $\mathbf{d} = \mathbf{u}^f \Delta t + \mathbf{d}_n$, then computing the reaction force $\mathbf{h}^s$ on Γ_t after solving (25). This is followed by using the Neumann boundary condition $\boldsymbol{\sigma}^f \cdot \mathbf{n}^s = \mathbf{h}^s$ to solve the fluid equation (21). For illustration purposes, **Algorithm** 1 adopts the former approach.

Utilising the finite element method, the weak form of the fluid equations (21) and (3) can be expressed as follows given test functions $\delta\mathbf{u}$ and δp:

$$\begin{aligned} &\int_{\Omega_t^f} \rho^f \partial_t \mathbf{u}^f \cdot \delta\mathbf{u} + \int_{\Omega_t^f} \rho^f \left(\left(\mathbf{u}^f - \mathbf{w} \right) \cdot \nabla \right) \mathbf{u}^f \cdot \delta\mathbf{u} \\ &= \int_{\partial\Omega_t^f} \left(\boldsymbol{\sigma}^f \cdot \mathbf{n} \right) \cdot \delta\mathbf{u} - \int_{\Omega_t^f} \boldsymbol{\sigma}^f : \nabla\delta\mathbf{u} + \int_{\Omega_t^f} \rho^f \mathbf{g} \cdot \delta\mathbf{u}, \end{aligned} \tag{26}$$

and

$$\int_{\Omega_t^f} \delta p \nabla \mathbf{u} = 0. \tag{27}$$

In equation (26), the fluid boundary comprises three parts, $\partial\Omega_t^f = \Gamma_N \cup \Gamma_D \cup \Gamma_t$. When solving the fluid problem using the Dirichlet boundary condition at the interface Γ_t, as adopted in **Algorithm** 1, the boundary integral in (26) becomes:

$$\int_{\partial\Omega_t^f} \left(\boldsymbol{\sigma}^f \cdot \mathbf{n} \right) \cdot \delta\mathbf{u} = \int_{\Gamma_N} \bar{\mathbf{h}} \cdot \delta\mathbf{u} \, . \tag{28}$$

By solving the aforementioned equation with Dirichlet boundary conditions (5) and (7) to obtain $\mathbf{u}^f$, the reaction force $\mathbf{h}^f = \boldsymbol{\sigma}^f \cdot \mathbf{n}^s \big|_{\partial\Omega_t^f}$ is then obtained as a post-processing step:

$$
\begin{aligned}
&\int_{\partial\Omega_t^f} \mathbf{h}^f \cdot \delta\mathbf{u} = \int_{\partial\Omega_t^f} \left(\boldsymbol{\sigma}^f \cdot \mathbf{n}^s\right) \cdot \delta\mathbf{u} \\
&= \int_{\Omega_t^f} \rho^f \partial_t \mathbf{u}^f \cdot \delta\mathbf{u} + \int_{\Omega_t^f} \rho^f \left(\left(\mathbf{u}^f - \mathbf{w}\right) \cdot \nabla\right) \mathbf{u}^f \cdot \delta\mathbf{u} \\
&+ \int_{\Omega_t^f} \boldsymbol{\sigma}^f : \nabla\delta\mathbf{u} - \int_{\Omega_t^f} \rho^f \mathbf{g} \cdot \delta\mathbf{u},
\end{aligned} \tag{29}
$$

which is a rearrangement of equation (26). Notice that the Dirichlet boundary conditions should be omitted when computing the reaction force using (29).

Algorithm 1 Partitioned/Segregated methods

Given the solid displacement $\mathbf{d}_n$ and the fluid velocity $\mathbf{u}_n^f$ at time t_n, the following iterative process computes $\mathbf{d}_{n+1}$ and $\mathbf{u}_{n+1}^f$ at time t_{n+1}:

1. Initiate with a reasonable estimate of $\Delta\mathbf{d}$ (such as 0) on the interface Γ_t. Solve the mesh equation (22) with boundary conditions (23) and (24) to determine the mesh velocity $\mathbf{w}$.
2. Solve the fluid equations (26) and (27) with boundary conditions (5) ($\mathbf{u}^s|_{\Gamma_t} = \mathbf{w}|_{\Gamma_t}$), (7), and (8).
3. Compute the reaction force $\mathbf{h}^f$ on Γ_t using equation (29).
4. Solve the solid equation (25), corresponding weak form (31) or the linearised weak form (44)) with Neumann boundary condition $\boldsymbol{\sigma} \cdot \mathbf{n}^s = \mathbf{h}^f$ on Γ_t and Dirichlet boundary condition $\mathbf{d} = \bar{\mathbf{u}}\Delta t + \mathbf{d}_n$ on $\Gamma_D \cap \Omega_t^s$.
5. Let $\Delta\mathbf{d}_t = \mathbf{d} - \mathbf{d}_n$. If $\|\Delta\mathbf{d}_t - \Delta\mathbf{d}\|_{\Gamma_t} \geq tol$, then let $\Delta\mathbf{d} = (1-\omega)\,\Delta\mathbf{d} + \omega\Delta\mathbf{d}_t$ and proceed to Step 2 for the next iteration cycle. Here, tol represents a tolerance and ω is a relaxation factor.

Let's bypass the detailed explanation of solving the fluid equations (Step 2 in **Algorithm 1**) and concentrate on solving the solid equation (25) (step 4 in **Algorithm 1**). This is because the former can be found in standard literature (refer to the list of bibliographies after Table 1). However, the methodologies for solving the latter problem introduced here are not widely available. We refer to [12, 154, 155] and bring some components together in the following context. Let us take the Saint Venant-Kirchhoff solid model as an example, and consider a method of linearisation in order to solve the solid equation (25).

Given a test function $\delta\mathbf{d} = \delta\mathbf{d}\left(\mathbf{x}\right) = \delta\mathbf{d}\left(\mathbf{x}(\mathbf{X})\right)$, the weak form of the solid equation (25) can be expressed as:

$$
\int_{\Omega_t^s} \rho^s \partial_{tt}\mathbf{d} \cdot \delta\mathbf{d} + \int_{\Omega_t^s} \boldsymbol{\sigma}^s : \nabla\delta\mathbf{d} = \int_{\Gamma_t} \left(\boldsymbol{\sigma}^s \cdot \mathbf{n}^s\right) \cdot \delta\mathbf{d} + \int_{\Omega_t^s} \rho^s \mathbf{g} \cdot \delta\mathbf{d}, \tag{30}
$$

or (with $\rho_0^s = J\rho^s$ denoting the solid's initial density):

$$\int_{\Omega_0^s} \rho_0^s \partial_{tt}\mathbf{d} \cdot \delta\mathbf{d} + \int_{\Omega_t^s} \boldsymbol{\sigma}^s : \nabla\delta\mathbf{d} = \int_{\Gamma_t} (\boldsymbol{\sigma}^s \cdot \mathbf{n}^s) \cdot \delta\mathbf{d} + \int_{\Omega_0^s} \rho_0^s \mathbf{g} \cdot \delta\mathbf{d}. \tag{31}$$

We rewrite the integral of the Cauchy stress tensor in the current configuration in (31) as an integral of the second Piola-Kirchhoff stress in the reference configuration:

$$\int_{\Omega_t^s} \boldsymbol{\sigma}^s : \nabla\delta\mathbf{d} = \int_{\Omega_t^s} J^{-1}\mathbf{F}\mathbf{S}\mathbf{F}^\top : \nabla\delta\mathbf{d} = \int_{\Omega_0^s} \mathbf{S} : \mathbf{F}^\top \nabla_{\mathbf{X}}\delta\mathbf{d}. \tag{32}$$

The symmetry of the second Piola-Kirchhoff stress $\mathbf{S}$ leads to:

$$\mathbf{S} : \mathbf{F}^\top (\nabla_{\mathbf{X}}\delta\mathbf{d}) = \mathbf{S}^\top : \mathbf{F}^\top (\nabla_{\mathbf{X}}\delta\mathbf{d}) = \mathbf{S} : \left(\nabla_{\mathbf{X}}^\top \delta\mathbf{d}\right) \mathbf{F}, \tag{33}$$

which further yields:

$$\mathbf{S} : \mathbf{F}^\top (\nabla_{\mathbf{X}}\delta\mathbf{d}) = \frac{1}{2}\mathbf{S} : \left[\mathbf{F}^\top (\nabla_{\mathbf{X}}\delta\mathbf{d}) + \left(\nabla_{\mathbf{X}}^\top \delta\mathbf{d}\right) \mathbf{F}\right] = \mathbf{S} : \delta\mathbf{E}, \tag{34}$$

where $\mathbf{E}$ is the Lagrangian Green strain tensor defined in (14), and

$$\delta\mathbf{E} = \frac{1}{2}\left[\mathbf{F}^\top (\nabla_{\mathbf{X}}\delta\mathbf{d}) + \left(\nabla_{\mathbf{X}}^\top \delta\mathbf{d}\right) \mathbf{F}\right]. \tag{35}$$

Finally, combining equations (32), (34) and (35), we obtain:

$$\int_{\Omega_t^s} \boldsymbol{\sigma}^s : \nabla\delta\mathbf{d} = \int_{\Omega_0^s} \mathbf{S} : \delta\mathbf{E}. \tag{36}$$

Remark 5. *The notations $\delta\mathbf{u}$, $\delta\mathbf{d}$, or $\delta\mathbf{E}$ can be interpreted in three ways: as a test function within a finite element space, as an infinitesimal virtual velocity (displacement or strain), or as a variation operator defined in A.*

Remark 6. *If $\delta\mathbf{E}$ is interpreted as a virtual strain. Consequently, $\int_{\Omega_0^s} \mathbf{S} : \delta\mathbf{E}$ represents the work done by the second Piola-Kirchhoff stress $\mathbf{S}$ on the virtual strain $\delta\mathbf{E}$. Equation (31) can therefore be interpreted as a work balance. Moreover, $\delta\mathbf{E}$ can be viewed as the variation of the Lagrangian Green strain $\mathbf{E}$ based on the Gateaux variation (see A).*

Let us now deduce the variation of the Lagrangian Green strain $\mathbf{E}$ and linearise expression (36). Taking the variation of $\mathbf{E}$ based on (14) gives:

$$\delta\mathbf{E}(\mathbf{d}; \delta\mathbf{d}) = \frac{1}{2}\left(\mathbf{F}^\top \delta\mathbf{F} + \delta\mathbf{F}^\top \mathbf{F}\right) = \frac{1}{2}\left[\mathbf{F}^\top (\nabla_{\mathbf{X}}\delta\mathbf{d}) + \left(\nabla_{\mathbf{X}}^\top \delta\mathbf{d}\right) \mathbf{F}\right], \tag{37}$$

which explains that (35) is not just a notation (see Remark 6). This can be equivalently expressed, based on (15), as:

$$\delta\mathbf{E}(\mathbf{d}; \delta\mathbf{d}) = \frac{1}{2}\left(\mathrm{D}_{\mathbf{X}}\delta\mathbf{d} + \nabla_{\mathbf{X}}^\top \mathbf{d} \nabla_{\mathbf{X}}\delta\mathbf{d} + \nabla_{\mathbf{X}}^\top \delta\mathbf{d} \nabla_{\mathbf{X}}\mathbf{d}\right). \tag{38}$$

The second order variation of $\mathbf{E}$, along a direction $\mathbf{h}$, is:

$$\begin{aligned}\delta^2\mathbf{E}(\mathbf{d};\delta\mathbf{d},\mathbf{h}) &= \partial_\epsilon \delta\mathbf{E}(\mathbf{d}+\epsilon\mathbf{h};\delta\mathbf{d})|_{\epsilon=0}\\ &= \frac{1}{2}\left(\nabla_{\mathbf{X}}^\top \mathbf{h}\nabla_{\mathbf{X}}\delta\mathbf{d} + \nabla_{\mathbf{X}}^\top \delta\mathbf{d}\nabla_{\mathbf{X}}\mathbf{h}\right).\end{aligned} \tag{39}$$

We are now ready to linearise term (36). Let $\mathcal{F}(\mathbf{d}) = \mathbf{S} : \delta\mathbf{E} = \mathbf{S}\circ\mathbf{E}(\mathbf{d}) : \delta\mathbf{E}(\mathbf{d};\delta\mathbf{d})$, the first order Taylor expression of $\mathcal{F}(\mathbf{d})$ at a reference point $\tilde{\mathbf{d}}$ gives

$$\mathcal{F}(\mathbf{d}) \approx \mathcal{F}(\tilde{\mathbf{d}}) + \delta\mathcal{F}(\tilde{\mathbf{d}};\mathbf{h}), \quad \mathbf{h} = \mathbf{d} - \tilde{\mathbf{d}}. \tag{40}$$

The first term in (40) can be expressed as:

$$\mathcal{F}(\tilde{\mathbf{d}}) = \mathbf{S}\circ\mathbf{E}(\tilde{\mathbf{d}}) : \delta\mathbf{E}(\tilde{\mathbf{d}};\delta\mathbf{d}), \tag{41}$$

and the second term in (40) can be expressed as:

$$\delta\mathcal{F}(\tilde{\mathbf{d}};\mathbf{h}) = \mathbf{S}\circ\delta\mathbf{E}(\tilde{\mathbf{d}};\mathbf{h}) : \delta\mathbf{E}(\tilde{\mathbf{d}};\delta\mathbf{d}) + \mathbf{S}\circ\mathbf{E}(\tilde{\mathbf{d}}) : \delta^2\mathbf{E}(\tilde{\mathbf{d}};\delta\mathbf{d},\mathbf{h}). \tag{42}$$

Combing equations (41) and (42), together with $\mathbf{h} = \mathbf{d} - \tilde{\mathbf{d}}$, yields

$$\begin{aligned}\mathcal{F}(\mathbf{d}) &\approx \mathbf{S}\circ\delta\mathbf{E}(\tilde{\mathbf{d}};\mathbf{d}-\tilde{\mathbf{d}}) : \delta\mathbf{E}(\tilde{\mathbf{d}};\delta\mathbf{d}) + \mathbf{S}\circ\mathbf{E}(\tilde{\mathbf{d}}) : \left[\delta\mathbf{E}(\tilde{\mathbf{d}};\delta\mathbf{d}) + \delta^2\mathbf{E}(\tilde{\mathbf{d}};\delta\mathbf{d},\mathbf{d}-\tilde{\mathbf{d}})\right]\\ &= \mathbf{S}\circ\delta\mathbf{E}(\tilde{\mathbf{d}};\mathbf{d}-\tilde{\mathbf{d}}) : \delta\mathbf{E}(\tilde{\mathbf{d}};\delta\mathbf{d}) + \mathbf{S}\circ\mathbf{E}(\tilde{\mathbf{d}}) : \delta\mathbf{E}(\mathbf{d};\delta\mathbf{d})\\ &= \mathbf{S}\circ\delta\mathbf{E}(\tilde{\mathbf{d}};\mathbf{d}) : \delta\mathbf{E}(\tilde{\mathbf{d}};\delta\mathbf{d}) + \mathbf{S}\circ\mathbf{E}(\tilde{\mathbf{d}}) : \delta\mathbf{E}(\mathbf{d};\delta\mathbf{d}) - \mathbf{S}\circ\delta\mathbf{E}(\tilde{\mathbf{d}};\tilde{\mathbf{d}}) : \delta\mathbf{E}(\tilde{\mathbf{d}};\delta\mathbf{d}).\end{aligned} \tag{43}$$

In the above, $\delta\mathbf{E}(\tilde{\mathbf{d}};\delta\mathbf{d}) + \delta^2\mathbf{E}(\tilde{\mathbf{d}};\delta\mathbf{d},\mathbf{d}-\tilde{\mathbf{d}}) = \delta\mathbf{E}(\mathbf{d};\delta\mathbf{d})$ may be validated using (38) and (39). Alternatively, since $\delta\mathbf{E}(\mathbf{d};\delta\mathbf{d})$ is linear in terms of $\mathbf{d}$, this can also be justified by Taylor expansion. Finally, the linearised (31) becomes:

$$\begin{aligned}&\int_{\Omega_0^s}\rho_0^s\partial_{tt}\mathbf{d}\cdot\delta\mathbf{d} + \int_{\Omega_0^s}\mathbf{S}\circ\delta\mathbf{E}(\tilde{\mathbf{d}};\mathbf{d}) : \delta\mathbf{E}(\tilde{\mathbf{d}};\delta\mathbf{d}) + \mathbf{S}\circ\mathbf{E}(\tilde{\mathbf{d}}) : \delta\mathbf{E}(\mathbf{d};\delta\mathbf{d})\\ &= \int_{\Omega_0^s}\mathbf{S}\circ\delta\mathbf{E}(\tilde{\mathbf{d}};\tilde{\mathbf{d}}) : \delta\mathbf{E}(\tilde{\mathbf{d}};\delta\mathbf{d}) + \int_{\Gamma_t}(\boldsymbol{\sigma}^s\cdot\mathbf{n}^s)\cdot\delta\mathbf{d} + \int_{\Omega_0^s}\rho_0^s\mathbf{g}\cdot\delta\mathbf{d}.\end{aligned} \tag{44}$$

It can be seen from **Algorithm** 1 that the partitioned methods have to iterate at every time step until the velocity on the interface Γ_t does not change. The convergence is problem dependent and cannot be guaranteed even if relaxation is adopted, especially in the case where the solid and fluid have a large energy exchange, for example, when the solid and fluid have similar inertia.

3.2 Monolithic ALE methods (Method 2.1)

The monolithic methods are considered more robust, addressing the weaknesses inherent in the partitioned methods. However, this approach shifts the challenge to

solving the discretised linear equation system. The monolithic methods solve the fluid equations (21) and (3) in Ω_t^f, the solid equation (25) in Ω_t^s and the mesh equation (22) in Ω_t^f, together with the consistency equation on Γ_t:

$$\partial_t \mathbf{d} = \mathbf{u}^f. \tag{45}$$

We use a Lagrange multiplier $\mathbf{L}$ to enforce this condition on Γ_t, and express the weak form of these equations with corresponding boundary conditions which are the same as discussed in Section 3.1. The unknown variables include fluid velocity $\mathbf{u}^f$ and pressure p, solid displacement $\mathbf{d}$, Lagrange multiplier $\mathbf{L}$ and mesh velocity $\mathbf{w}$, with the corresponding test functions being denoted by $\delta\mathbf{u}$, δp, $\delta\mathbf{d}$, $\delta\mathbf{L}$ and $\delta\mathbf{w}$, respectively. Then the weak form of the solid equation (25) is almost the same as (44), but replacing the stress $\boldsymbol{\sigma}^s \cdot \mathbf{n}^s$ on Γ_t by a Lagrange multiplier $\mathbf{L}$, which could be expressed as

$$\begin{aligned} &\int_{\Omega_0^s} \rho_0^s \partial_{tt}\mathbf{d} \cdot \delta\mathbf{d} + \int_{\Omega_0^s} \mathbf{S} \circ \delta\mathbf{E}(\tilde{\mathbf{d}}; \mathbf{d}) : \delta\mathbf{E}(\tilde{\mathbf{d}}; \delta\mathbf{d}) + \mathbf{S} \circ \mathbf{E}(\tilde{\mathbf{d}}) : \delta\mathbf{E}(\mathbf{d}; \delta\mathbf{d}) \\ &- \int_{\Gamma_t} \mathbf{L} \cdot \delta\mathbf{d} = \int_{\Omega_0^s} \mathbf{S} \circ \delta\mathbf{E}(\tilde{\mathbf{d}}; \tilde{\mathbf{d}}) : \delta\mathbf{E}(\tilde{\mathbf{d}}; \delta\mathbf{d}) + \int_{\Omega_0^s} \rho_0^s \mathbf{g} \cdot \delta\mathbf{d}. \end{aligned} \tag{46}$$

The weak form of the fluid equations (26) and (27), using the constitutive equation (4) and considering the Lagrange multiplier $\mathbf{L}$ on Γ_t (replacing the stress $\boldsymbol{\sigma}^f \cdot \mathbf{n}^s = \boldsymbol{\sigma}^s \cdot \mathbf{n}^s$ on Γ_t by a Lagrange multiplier $\mathbf{L}$ based on the interface boundary condition (6)), can be expressed as:

$$\begin{aligned} &\int_{\Omega_t^f} \rho^f \partial_t \mathbf{u}^f \partial t \cdot \delta\mathbf{u} + \int_{\Omega_t^f} \rho^f \left(\left(\mathbf{u}^f - \mathbf{w} \right) \cdot \nabla \right) \mathbf{u}^f \cdot \delta\mathbf{u} \\ &+ \int_{\Omega_t^f} \mu^f \mathrm{D}\mathbf{u}^f : \nabla\delta\mathbf{u} - \int_{\Omega_t^f} p \nabla \cdot \delta\mathbf{u} + \int_{\Gamma_t} \mathbf{L} \cdot \delta\mathbf{u} \\ &= \int_{\Gamma_N} \bar{\mathbf{h}} \cdot \delta\mathbf{u} + \int_{\Omega_t^f} \rho^f \mathbf{g} \cdot \delta\mathbf{u}, \end{aligned} \tag{47}$$

and

$$- \int_{\Omega_t^f} \delta p \nabla \cdot \mathbf{u}^f = 0. \tag{48}$$

The weak form of the consistency equation (45) can be expressed as

$$\int_{\Gamma_t} \left(\partial_t \mathbf{d} - \mathbf{u}^f \right) \cdot \delta\mathbf{L} = 0. \tag{49}$$

Finally, the weak form of the mesh equation (22) may be expressed as follows.

$$\frac{\mu_m}{2} \int_{\Omega_t^f} \mathrm{D}\mathbf{w} : \mathrm{D}\delta\mathbf{w} - \lambda_m \int_{\Omega_t^f} \left(\nabla \cdot \mathbf{w} \right) \left(\nabla \cdot \delta\mathbf{w} \right) d\mathbf{x} = 0. \tag{50}$$

We use the Characteristic–Galerkin method [156] to discretise the total derivative of the fluid velocity:

$$\partial_t \mathbf{u}^f + \left(\left(\mathbf{u}^f - \mathbf{w}\right) \cdot \nabla\right) \mathbf{u}^f \approx \frac{\mathbf{u}^f_{n+1} - \mathbf{u}^f_n \circ \mathbb{Y}_n}{\Delta t}, \tag{51}$$

where $\mathbb{Y}_n(\mathbf{x}) = \mathbf{x} - \mathbf{w}\Delta t$ and $\mathbf{u}^f_n \circ \mathbb{Y}_n(\mathbf{x}) = \mathbf{u}_n(\mathbf{x} - \mathbf{w}\Delta t)$.

We discretise the time derivatives of the solid as follows:

$$\mathbf{u}^s_{n+1} = \partial_t \mathbf{d}_{n+1} \approx \frac{\mathbf{d}_{n+1} - \mathbf{d}_n}{\Delta t}, \tag{52}$$

and

$$\partial_{tt} \mathbf{d}_{n+1} \approx \frac{\mathbf{d}_{n+1} - 2\mathbf{d}_n + \mathbf{d}_{n-1}}{\Delta t^2}. \tag{53}$$

For simplicity, we omit the subscript $n+1$ for $\mathbf{u}_{n+1}$ and $\mathbf{d}_{n+1}$. Let Ω^f_{n+1} and Γ_{n+1} denote the fluid domain and the fluid–solid interface, respectively, at time t_{n+1}. The monolithic weak form is obtained by substituting the time discretisations in (51), (52), and (53) into (47), (49), and (46), respectively.

$$\begin{aligned}
&\int_{\Omega^f_{n+1}} \rho^f \frac{\mathbf{u}^f - \mathbf{u}^f_n \circ \mathbb{Y}_n}{\Delta t} \cdot \delta\mathbf{u} + \frac{1}{2} \int_{\Omega^f_{n+1}} \mu^f \mathrm{D}\mathbf{u}^f : \mathrm{D}\delta\mathbf{u} \\
&- \int_{\Omega^f_{n+1}} p \nabla \cdot \delta\mathbf{u} - \int_{\Omega^f_{n+1}} \delta p \nabla \cdot \mathbf{u}^f \\
&+ \int_{\Omega^s_0} \rho^s_0 \frac{\mathbf{d} - 2\mathbf{d}_n + \mathbf{d}_{n-1}}{\Delta t^2} \cdot \frac{\delta\mathbf{d}}{\Delta t} \\
&+ \frac{1}{\Delta t} \int_{\Omega^s_0} \mathbf{S} \circ \delta\mathbf{E}(\tilde{\mathbf{d}}; \mathbf{d}) : \delta\mathbf{E}(\tilde{\mathbf{d}}; \delta\mathbf{d}) + \mathbf{S} \circ \mathbf{E}(\tilde{\mathbf{d}}) : \delta\mathbf{E}(\mathbf{d}; \delta\mathbf{d}) \\
&- \int_{\Gamma_{n+1}} \mathbf{L} \cdot \left(\frac{\delta\mathbf{d}}{\Delta t} - \delta\mathbf{u}\right) - \int_{\Gamma_{n+1}} \left(\frac{\mathbf{d} - \mathbf{d}_n}{\Delta t} - \mathbf{u}^f\right) \cdot \delta\mathbf{L} \\
&+ \frac{\mu_m}{2} \int_{\Omega^f_{n+1}} \mathbf{D}\mathbf{w} : \mathbf{D}\delta\mathbf{w} - \lambda_m \int_{\Omega^f_{n+1}} (\nabla \cdot \mathbf{w}) (\nabla \cdot \delta\mathbf{w}) \\
&= \int_{\Gamma_N} \bar{\mathbf{h}} \cdot \delta\mathbf{u} + \int_{\Omega^f_t} \rho^f \mathbf{g} \cdot \delta\mathbf{u} + \int_{\Omega^s_0} \rho^s_0 \mathbf{g} \cdot \frac{\delta\mathbf{d}}{\Delta t} \\
&+ \frac{1}{\Delta t} \int_{\Omega^s_0} \mathbf{S} \circ \delta\mathbf{E}(\tilde{\mathbf{d}}; \tilde{\mathbf{d}}) : \delta\mathbf{E}(\tilde{\mathbf{d}}; \delta\mathbf{d}),
\end{aligned} \tag{54}$$

with initial conditions (9) and (10), boundary conditions (7) and (8) for the fluid velocity $\mathbf{u}^f$, and boundary conditions (23) and (24) for the mesh velocity $\mathbf{w}$. It's important to note that we divide by Δt on both sides of equation (46) before summing up in order to produce the symmetric linear system below. Equation (54) yields the

following linear algebraic system after spatial discretisation:

$$\begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} & \mathbf{B}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{12}^\top & \mathbf{F}_{22} & \mathbf{B}_2 & \mathbf{0} & \mathbf{0} & \mathbf{C}_1 & \mathbf{0} \\ \mathbf{B}_1^\top & \mathbf{B}_2^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{12}^\top & \mathbf{S}_{22} & \mathbf{C}_2 & \mathbf{C}_m \\ \mathbf{0} & \mathbf{C}_1^\top & \mathbf{0} & \mathbf{0} & \mathbf{C}_2^\top & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C}_m^\top & \mathbf{0} & \mathbf{K} \end{bmatrix} \begin{pmatrix} \mathbf{u}_{\text{in}} \\ \mathbf{u}_{\text{on}} \\ \mathbf{p} \\ \mathbf{d}_{\text{in}} \\ \mathbf{d}_{\text{on}} \\ \mathbf{L} \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{0} \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}. \tag{55}$$

In the above, $\begin{pmatrix} \mathbf{u}_{\text{in}} \\ \mathbf{u}_{\text{on}} \end{pmatrix} = \mathbf{u}$, with $\mathbf{u}_{\text{on}}$ being the velocity components on boundary Γ_t, and $\mathbf{u}_{\text{in}}$ being the velocity components in $\overline{\Omega}_t^f \setminus \Gamma_t$. Similarly $\begin{pmatrix} \mathbf{d}_{\text{in}} \\ \mathbf{d}_{\text{on}} \end{pmatrix} = \mathbf{d}$, where $\mathbf{d}_{\text{on}}$ are the displacement components on boundary Γ_t, and $\mathbf{d}_{\text{in}}$ the displacement components in $\overline{\Omega}_t^s \setminus \Gamma_t$. Matrices $\begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \mathbf{F}_{12}^\top & \mathbf{F}_{22} \end{bmatrix}$ and $\begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{12}^\top & \mathbf{S}_{22} \end{bmatrix}$ are discretisations of the fluid velocity integrals and the solid displacement integrals, respectively (including mass matrices and stiffness matrices). Matrix $\begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{pmatrix}$ indicates the coupling between fluid velocity and pressure. Matrix $\mathbf{C}_1$ represents the coupling between fluid velocity $\mathbf{u}_{\text{on}}$ and the Lagrange multiplier $\mathbf{L}$, while $\mathbf{C}_2$ denotes the coupling between solid displacement $\mathbf{d}_{\text{on}}$ and the Lagrange multiplier $\mathbf{L}$. Matrix $\mathbf{K}$ arises from the discretisation of the mesh equation, and matrix $\mathbf{C}_m$ indicates the coupling between the mesh velocity $\mathbf{w}$ and solid displacement $\mathbf{d}_{\text{on}}$ due to boundary condition (24).

As mentioned at the beginning of this section, the major advantage of the monolithic approach is its general stability and robustness owing to the fully-coupled system. However, this also presents a disadvantage: solving the large linear system (55) requires a very powerful linear algebraic solver.

Remark 7. *Throughout this paper, we use the Characteristic–Galerkin method for time discretisation. One reason is that it facilitates the presentation, as the convection term is naturally absorbed into the time discretisation. Another reason is its convenient implementation in* `FreeFEM++` *that we use. However, other time discretisation schemes may also be adopted, such as the backward Euler method for the partial time derivative, combined with either a splitting strategy or a Newton method to treat the convection term.*

3.3 One-velocity monolithic method (Method 4)

The monolithic Eulerian method [50–53] reduces the size of the linear system (55) by solving for only one velocity field across the entire fluid-structure domain $\Omega_t^f \cup \Omega_t^s$. However, it still maintains much of its robustness due to the system remaining fully-coupled and being solved simultaneously. The use of the Lagrange multiplier becomes unnecessary with this method because there is only one velocity variable defined on the fluid-solid interface, as depicted in Figure 3.

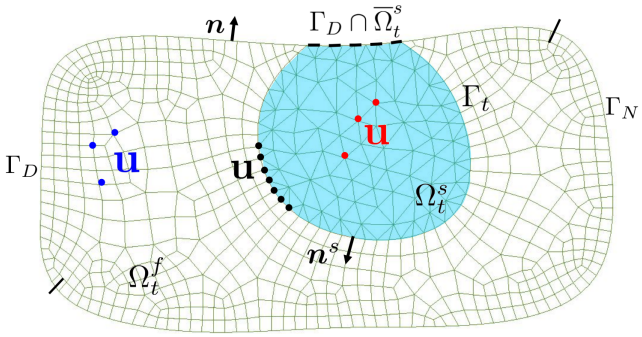

Fig. 3: Schematic diagram illustrating an interface-fitted mesh, where $\Omega = \Omega_t^f \cup \Omega_t^s$, and $\partial\Omega = \Gamma_D \cup \Gamma_N$. Solving for a single velocity $\mathbf{u}$ within Ω.

References [50–52] employ a remeshing technique to maintain a mesh fitted on the fluid-structure interface. Here, we adopt an ALE mesh as adopted in [53]. We assume both the fluid and solid are incompressible, as established in these references, to introduce the one-velocity monolithic formulation. By utilising the same test function $\delta\mathbf{u}$ for both fluid and solid equations, the weak form of the momentum equation (1) and continuity equation (2) can be expressed as:

$$\begin{aligned}&\int_\Omega \rho \mathcal{D}_t \mathbf{u} \cdot \delta\mathbf{u} + \frac{\mu^f}{2}\int_{\Omega_t^f} \mathrm{D}\mathbf{u} : \mathrm{D}\delta\mathbf{u} - \int_\Omega p\nabla\cdot\delta\mathbf{u} - \int_\Omega \delta p \nabla\cdot\mathbf{u} \\ &+ \int_{\Omega_t^s} \boldsymbol{\tau}^s : \nabla\delta\mathbf{u} = \int_{\Gamma_N} \bar{\mathbf{h}}\cdot\delta\mathbf{u} + \int_\Omega \rho\mathbf{g}\cdot\delta\mathbf{u},\end{aligned} \tag{56}$$

where $\boldsymbol{\tau}^s$ is implicitly viewed as a function of $\boldsymbol{u}$ and t, that is, $\boldsymbol{\tau}^s(\boldsymbol{u}, t)$. Using the Characteristic-Galerkin method as previously employed in (51), we once again apply it to discretise the total time derivative in (56). Consequently, we obtain:

$$\begin{aligned}&\int_\Omega \rho \frac{\mathbf{u} - \mathbf{u}_n \circ \mathbb{Y}_n}{\Delta t}\cdot\delta\mathbf{u} + \frac{\mu^f}{2}\int_{\Omega_{n+1}^f} \mathrm{D}\mathbf{u} : \mathrm{D}\delta\mathbf{u} - \int_\Omega p\nabla\cdot\delta\mathbf{u} - \int_\Omega \delta p\nabla\cdot\mathbf{u} \\ &+ \int_{\Omega_{n+1}^s} \boldsymbol{\tau}^s : \nabla\delta\mathbf{u} = \int_{\Gamma_N} \bar{\mathbf{h}}\cdot\delta\mathbf{u} + \int_\Omega \rho\mathbf{g}\cdot\delta\mathbf{u},\end{aligned} \tag{57}$$

with the fluid boundary conditions (7) and (8).

The above equation is coupled with the mesh equation (22), which is solved over the whole domain:

$$\frac{\mu_m}{2}\int_\Omega \mathrm{D}\mathbf{w} : \mathrm{D}\delta\mathbf{w} - \lambda_m \int_\Omega (\nabla\cdot\mathbf{w})(\nabla\cdot\delta\mathbf{w})\, d\mathbf{x} = 0, \tag{58}$$

together with its boundary conditions (23) and

$$\mathbf{w} = \mathbf{u} \quad \text{on} \quad \Gamma_t. \tag{59}$$

Notice that it is also possible to solve the mesh equation only in the fluid domain Ω_t^f, as adopted in the previous two sections for Method 1 and Method 2.1, and to move the solid mesh using its own velocity. By doing so, the convection term in the solid domain in (57) must be omitted. However, because of the one-velocity feature, it is more convenient to solve the mesh equation over the entire domain Ω for this method. See the implementation in Section 6.

To implement (57), it is necessary to express $\boldsymbol{\tau}^s$ as velocity. This can be accomplished using the following three different approaches based on the variable being updated after time discretisation.

F-scheme: utilising the constitutive relation (20), the stress integral in (57) can be expressed as:

$$\int_{\Omega_{n+1}^s} \boldsymbol{\tau}^s : \nabla \delta\mathbf{u} = \mu^s \int_{\Omega_{\mathbf{X}}} \mathbf{F} : \nabla_{\mathbf{X}} \delta\mathbf{u} - \mu^s \int_{\Omega_{n+1}^s} J^{-1} \nabla \cdot \delta\mathbf{u}. \tag{60}$$

Since $\partial_t \mathbf{F} = \nabla_{\mathbf{X}} \mathbf{u}$, $\mathbf{F}$ can be updated as: $\mathbf{F}_{n+1} = \mathbf{F}_n + \Delta t \mathbf{u}_{n+1}$.

$\boldsymbol{\tau}$-scheme: as introduced in [80], this scheme represents a higher-order discretisation in time:

$$\begin{aligned} \boldsymbol{\tau}_{n+1}^s = \boldsymbol{\tau}_n^s + \mu^s \Delta t \left(\mathrm{D}_n \mathbf{u} + \nabla_n \mathbf{u} \boldsymbol{\tau}_n^s + \boldsymbol{\tau}_n^s \nabla_n \mathbf{u} \right) \\ + \Delta t^2 \left(\nabla_n \mathbf{u} \nabla_n^\top \mathbf{u} + \nabla_n \mathbf{u} \boldsymbol{\tau}_n^s \nabla_n \mathbf{u} \right). \end{aligned} \tag{61}$$

In the above, the operators ∇_n and D_n act on the previous configuration $\mathbf{x}_n$. Again, we have omitted the superscript $n+1$ for $\mathbf{u}_{n+1}$, indicating the current time step.

d-scheme: the deviatoric stress, in 2D as presented in [51], can be expressed as:

$$\boldsymbol{\tau}^s = \mu^s \left(\mathrm{D}\mathbf{d} - \nabla \mathbf{d} \nabla^\top \mathbf{d} \right), \tag{62}$$

and in 3D as in [52]:

$$\boldsymbol{\tau}^s = \mu^s \left(\mathrm{D}\mathbf{d} - \nabla \mathbf{d} \nabla^\top \mathbf{d} \right)^2 + c(\mathbf{x}, \mathbf{t}) \left(\mathrm{D}\mathbf{d} - \nabla \mathbf{d} \nabla^\top \mathbf{d} \right), \tag{63}$$

where $c(\mathbf{x}, t)$ is a parameter dependent on the configuration (for further details, refer to [52]). The displacement can be updated as $\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t \mathbf{u}_{n+1}$.

These three schemes each come with their own advantages and disadvantages: both the **F**-scheme and $\boldsymbol{\tau}$-scheme share the same formulations in both 2D and 3D. The former is formulated in the reference configuration, while the latter is in the current configuration. The **F**-scheme is linear and straightforward to implement, whereas the $\boldsymbol{\tau}$-scheme facilitates remeshing if needed due to its formulation in the current configuration. The advantage of the **d**-scheme lies in updating a smaller tensor (a vector), in comparison to the other two schemes that update a larger tensor. However, a significant drawback of the **d**-scheme is its different formulations in 2D and 3D. Specifically, the 3D formulation is intricate, involving a nonlinear coefficient that depends on the material points.

Taking the **d**-scheme as an example, even in 2D, its nonlinearity leads to a complex update of the stress tensor, as illustrated below:

$$\begin{aligned}\boldsymbol{\tau}^s = &\Delta t\mu^s\left(\mathrm{D}\mathbf{u} - \Delta t\nabla\mathbf{u}\nabla^\top\mathbf{u}\right) - \Delta t\mu^s\left(\nabla\mathbf{d}_n\nabla^\top\mathbf{u} + \nabla\mathbf{u}\nabla^\top\mathbf{d}_n\right)\\ &+ \mu^s\mathrm{D}\mathbf{d}_n - \mu^s\nabla\mathbf{d}_n\nabla^\top\mathbf{d}_n.\end{aligned} \tag{64}$$

Substituting the above expression (64) into (57) and neglecting the second order term of Δt yields:

$$\begin{aligned}&\int_\Omega \rho\frac{\mathbf{u}-\mathbf{u}_n\circ\mathbb{Y}_n}{\Delta t}\cdot\delta\mathbf{u} + \frac{\mu}{2}\int_\Omega \mathrm{D}\mathbf{u}:\mathrm{D}\delta\mathbf{u} - \int_\Omega p\nabla\cdot\delta\mathbf{u} - \int_\Omega \delta p\nabla\cdot\mathbf{u}\\ &- \Delta t\mu^s\int_{\Omega^s_{n+1}}\left(\nabla\mathbf{d}_n\nabla^\top\mathbf{u} + \nabla\mathbf{u}\nabla^\top\mathbf{d}_n\right):\nabla\delta\mathbf{u} = \int_{\Gamma_N}\bar{\mathbf{h}}\cdot\delta\mathbf{u} + \int_\Omega\rho\mathbf{g}\cdot\delta\mathbf{u}\\ &- \frac{\mu^s}{2}\int_{\Omega^s_{n+1}}\mathrm{D}\mathbf{d}_n:\mathrm{D}\delta\mathbf{u} + \mu^s\int_{\Omega^s_{n+1}}\nabla\mathbf{d}_n\nabla^\top\mathbf{d}_n:\nabla\delta\mathbf{u},\end{aligned} \tag{65}$$

where $\mu = \mu^f 1_{\Omega^f_{n+1}} + \Delta t\mu^s 1_{\Omega^s_{n+1}}$.

Remark 8. *In both equations (54) and (65), there are integrals over the current domain at $t = t_{n+1}$, namely Ω^f_{n+1}, Ω^s_{n+1} or $\Omega = \Omega^f_{n+1}\cup\Omega^s_{n+1}$. Generally speaking, Ω_{n+1} needs be constructed iteratively from $\Omega^{(k)}_{n+1}$ with $k = 0, 1, 2, \ldots$ and $\Omega^{(0)}_{n+1} = \Omega_n$. See [51, 82] for further discussion of this procedure. In practice, one may instead integrate directly over the domain at $t = t_n$, using a small time step, without loss of accuracy or stability.*

3.4 Summary of the advantages and disadvantages

For all interface-fitted methods, the principal advantage lies in the use of a mesh that conforms to the interface, which generally ensures high accuracy in the numerical solution. However, this also constitutes a significant disadvantage: generating interface-fitted meshes for FSI problems is challenging, particularly in cases involving large solid deformations. Furthermore, additional errors are inevitably introduced when transferring solutions between meshes during frequent remeshing.

Two-way partitioned/segregated methods (Method 1):

- Advantages: Convenient to implement on the basis of existing fluid and solid codes.
- Disadvantages: Convergence cannot be guaranteed, particularly when there is frequent energy exchange between the fluid and the solid.

Monolithic ALE methods (Method 2.1):

- Advantages: Widely regarded as more robust FSI methods compared with their partitioned counterparts.
- Disadvantages: The resulting large discretised linear system is challenging to solve: An additional Lagrange multiplier is usually required to enforce consistency between the fluid velocity and the solid displacement at the interface. From the perspective

of the discretised linear system, the Lagrange multiplier introduces extra unstable directions (in addition to pressure) to the saddle-point linear system.

One-velocity monolithic method (Method 4):

- Advantages: Naturally monolithic because there is only one velocity at the interface, which automatically satisfies the velocity-consistency condition; involves fewer degrees of freedom compared with monolithic methods based on Lagrange multipliers that solve for the solid displacements.
- Disadvantages: It is challenging to express solid constitutive models in terms of velocity and displacement in the current configuration, especially in three dimensions. Even simple constitutive models formulated in the reference configuration often lead to complex non-linear formulations in the current configuration [157].

4 Methods using two meshes

The two-mesh methods generally utilise one Eulerian mesh to describe the entire domain $\Omega = \Omega_t^f \cup \Omega_t^s$ and a moving Lagrangian mesh to describe the solid domain Ω_t^s, as shown in Figure 4. The crucial advantage of the two-mesh methods is their ability to avoid the need for mesh adjustment to fit the fluid-structure interface. However, a disadvantage arises from the unfitted interface, causing material properties to be smeared across the interface. Therefore, managing the fluid-structure interface becomes the primary concern for two-mesh methods: this interface must be treated carefully to maintain stability and accuracy.

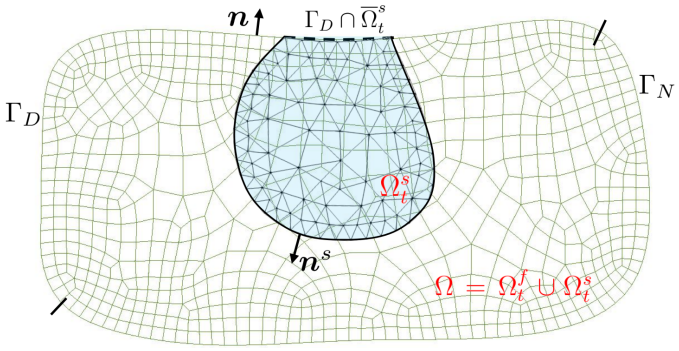

Fig. 4: Schematic diagram of FSI methods using two meshes: the background mesh covering the entire domain Ω remains fixed, while a foreground mesh discretises the solid domain Ω_t^s. In this scenario, the interface Γ_t between the fluid and solid, equivalent to the solid boundary, is not captured by the background mesh.

In two-mesh methods, it is often reasonable to assume an incompressible solid, given that the fluid is incompressible. This is because $\nabla \cdot \mathbf{u} = 0$ holds in the entire domain Ω, including the fictitious fluid region covered by the solid domain Ω_t^s. A projection $\mathbb{P}$ from Ω to Ω_t^s,

$$\mathbf{u}^s = \mathbb{P}(\mathbf{u}), \tag{66}$$

is naturally expected to satisfy $\nabla \cdot \mathbf{u}^s = 0$, implying an incompressible solid. However, constructing an exactly divergence-free projection is not straightforward; see, for example, a recent work [158]. In this section, we use the incompressible neo-Hookean solid model introduced in Section 2.2 to demonstrate the two-mesh methods.

Remark 9. *The background mesh employed for the partitioned [55] (Method 5.2) and monolithic [72, 73] (Method 6.2) CutFEM methods is the same as the mesh shown in Figure 4, whereas the foreground mesh exhibits slight variations. In particular, the solid foreground mesh is extended to include a surrounding artificial fluid region with a fitted interface. This modification enables the use of a Nitsche-type formulation with ghost-penalty stabilisation, specifically adapted to this mesh configuration.*

4.1 Immersed finite element methods (Method 7)

The immersed finite element methods (IFEM) originated from the Immersed Boundary Method (IBM), first introduced by Peskin [74], and have achieved considerable success in bioscience and biomedical applications [12, 150]. Variants of the IBM include the implicit IBM [118, 119], the monolithic IBM [117, 120, 121], and extensions to immersed interfaces to capture sharp interface behaviour [105, 122–131]. Other developments include level-set IBMs [107–111], finite element IBMs [95, 97], isogeometric IBMs [132–134], and immersed boundary–Lattice Boltzmann methods [119, 135–143].

In the classical IFEM, the solid equations are not solved explicitly; instead, the solid dynamics are incorporated on the right-hand side of the fluid equations as an FSI force $\mathbf{f}$. These modified fluid equations are then solved in an augmented domain that contains both the fluid and the fictitious fluid. Following this approach, the momentum equation of the fluid can be written as:

$$\rho^f \mathcal{D}_t \mathbf{u} = \nabla \cdot \boldsymbol{\sigma} + \rho^f \mathbf{g} + \mathbf{f}\left(\mathbf{u}\right), \tag{67}$$

where $\mathbf{f}$ is a singular force evaluated from the solid. Equation (67) illustrates that the essence of IFEM is to solve a modified fluid equation to emulate FSI behaviour. It's the singular force $\mathbf{f}$ that induces solid-like behaviour to the fictitious fluid. Notably, the method is more suitable and easier for simulation when dealing with solids and fluids exhibiting similarities (such as a very soft solid). However, there exist specifically designed IFEM methods tailored for rigid bodies, which are considered special cases [159, 160].

IFEM methods solve the fluid equations (67) and (2) in the entire domain $\Omega = \Omega_t^f \cup \Omega_t^s$. However, the computation of the FSI force can only be performed on the solid mesh Ω_t^s, denoted as $\mathbf{f}^s$. Therefore, a function must be defined to appropriately distribute $\mathbf{f}^s$ computed on the solid mesh to $\mathbf{f}$, which is applied on the background mesh. The following functions are commonly used to distribute $\mathbf{f}^s$ to $\mathbf{f}$ and interpolate the solution velocity $\mathbf{u}$ from the background mesh to the solid mesh $\mathbf{u}^s$, and accumulate the solid displacement $\mathbf{d}$ and further compute $\mathbf{f}^s$.

1. Discretised δ function [76–79]:

$$\delta^h(\mathbf{x}) = \frac{1}{h^d}\prod_i^d \varphi(x_i), \quad \varphi(r) = \begin{cases} \frac{1}{4}\left[1+\cos\left(\frac{\pi|r|}{2h}\right)\right], & |r| \le 2h \\ 0, & |r| > 2h \end{cases}, \tag{68}$$

with h being the mesh size (uniform mesh) and r being the distance between a solid node and the surrounding fluid node.
2. The shape function used in the Reproducing Kernel Particle Method (RKPM) [54, 75–78] (a type of mesh-free method [161]).
3. The finite element isoparametric interpolation functions [76].

Note that $\mathbf{f}^s$ can be evaluated in Ω_t^s using the following formula [54, 75–78]:

$$\mathbf{f}^s = -\rho^\delta \partial_t \mathbf{u}^s + \nabla \cdot \boldsymbol{\sigma}^s - \nabla \cdot \boldsymbol{\sigma}^f + \rho^\delta \mathbf{g}, \tag{69}$$

with $\rho^\delta = \rho^s - \rho^f$. By employing the finite element method, the corresponding weak form of (69), for a given test function $\delta\mathbf{d}$, is:

$$\int_{\Omega_t^s} \mathbf{f}^s \cdot \delta\mathbf{d} = -\int_{\Omega_t^s} \rho^\delta \partial_t \mathbf{u}^s \cdot \delta\mathbf{d} - \int_{\Omega_t^s} \boldsymbol{\sigma}^s : \nabla\delta\mathbf{d} + \int_{\Omega_t^s} \boldsymbol{\sigma}^f : \nabla\delta\mathbf{d} + \int_{\Omega_t^s} \rho^\delta \mathbf{g} \cdot \delta\mathbf{d}. \tag{70}$$

The boundary integrals cancel out due to the boundary condition (6), which enforces consistency of the normal force. Assuming the solid is incompressible and has the same pressure as the fictitious fluid, the fluid deviatoric stress $\boldsymbol{\tau}^f$ is sometimes neglected (i.e., assuming $\mu^f \ll \mu^s$) [78], which is equivalent to assuming that the solid has the same viscosity as the fluid. The above expression can then be simplified as follows:

$$\int_{\Omega_t^s} \mathbf{f}^s \cdot \delta\mathbf{d} = -\int_{\Omega_t^s} \rho^\delta \partial_t \mathbf{u}^s \cdot \delta\mathbf{d} - \int_{\Omega_t^s} \boldsymbol{\tau}^s : \nabla\delta\mathbf{d} + \int_{\Omega_t^s} \rho^\delta \mathbf{g} \cdot \delta\mathbf{d}. \tag{71}$$

Finally, the solution algorithm for the classical IFEM method is summarised in **Algorithm** 2 as follows.

Algorithm 2 Immersed Finite Element Method

Given the solid configuration $\mathbf{x}_n^s$ and the velocity $\mathbf{u}_n$ on the background mesh at time t_n.

1. Compute the FSI force $\mathbf{f}_n^s$ on the solid mesh using equation (71).
2. Distribute $\mathbf{f}_n^s$ to $\mathbf{f}_n$ on the background mesh.
3. Solve the fluid equations (67) and (2) on the background mesh to obtain velocity $\mathbf{u}_{n+1}$.
4. Interpolate the velocity $\mathbf{u}_{n+1}$ to $\mathbf{u}_{n+1}^s$ on the solid mesh.
5. Update the solid mesh using $\mathbf{u}_{n+1}^s$ and return to Step 1.

4.2 Modified immersed finite element methods (Method 5.1)

In the previous section, it was illustrated that the IFEM method does not directly solve the solid equation (25). Instead, the solid equation is used to compute the FSI force, as depicted in formula (71). The Modified Immersed Finite Element Method

(mIFEM) tackles the solid equation (25) with specific boundary conditions on the interface Γ_t and boundary $\Gamma_D \cap \overline{\Omega}_t^s$ [54]:

$$\mathbf{d} = \mathbf{d}_n + \Delta t \mathbb{P}(\mathbf{u}_n) \quad \text{on} \quad \Gamma_D \cap \overline{\Omega}_t^s, \tag{72}$$

and

$$\boldsymbol{\sigma}^s{}_{n+1} \cdot \mathbf{n}^s = \mathbb{P}(\boldsymbol{\sigma}_n) \cdot \mathbf{n}^s \quad \text{on} \quad \Gamma_t, \tag{73}$$

where $\mathbb{P}(\cdot)$ represents the projection/interpolation from the background mesh to the solid mesh, choosing from any of the three interpolation strategies discussed in the previous section. By modifying the algorithm of the IFEM method in **Algorithm** 2, we derive the solution algorithm for mIFEM in **Algorithm** 3.

Algorithm 3 Modified Immersed Finite Element Method

Given the solid configuration $\mathbf{x}_n^s$ and the velocity $\mathbf{u}_n$ on the background mesh at time t_n.

1. Compute the stress $\boldsymbol{\sigma}_n$ using (4), and further compute $\mathbb{P}(\mathbf{u}_n)$ and $\mathbb{P}(\boldsymbol{\sigma}_n)$.
2. Solve the solid equation (25) with boundary conditions (72) and (73) to obtain solid displacement $\mathbf{d}_{n+1}$.
3. Compute the FSI force $\mathbf{f}_{n+1}^s$ on the solid mesh, using equation (71).
4. Distribute $\mathbf{f}_{n+1}^s$ to $\mathbf{f}_{n+1}$ on the background mesh.
5. Solve the fluid equations (67) and (2) on the background mesh to get velocity $\mathbf{u}_{n+1}$.
6. Update the solid mesh using $\mathbf{d}_{n+1}$ and go back to Step 1.

The IFEM method relies significantly on the fluid equations, often necessitating a small time step to obtain an accurate solution. The approximations made within the method generally do not result in significant errors when the solid behaviour closely resembles that of the fluid. However, in scenarios where the solid behaviour becomes more predominant, the velocity obtained from solving the fluid equation might cause unrealistic solid deformations.

The mIFEM method aims to present more reasonable solid deformations by explicitly solving the solid equation. Nonetheless, the boundary conditions for the solid equation still derive from the values of the previous time step, which remains the same as the approach in the IFEM method.

4.3 Fictitious domain methods with distributed Lagrange multiplier (Method 6.1)

There are Fictitious Domain Methods with Distributed Lagrange Multiplier (FDM/DLM) developed for deformable solids [56–63], as well as for rigid solid bodies [59, 64–69]. These methods might be implicitly fully-coupled, as evident in studies such as [59, 60, 62, 63], or explicitly split, as observed in references like [57, 64, 66]. For demonstration, we focus on the fully-coupled implicit scheme applicable to general deformable solids in introducing these FDM/DLM methods [59, 60, 63].

Similar to the previous presentation of the Monolithic Eulerian method utilising an interface-fitted mesh in Section 3.3, the FDM/DLM method we discuss here is also fully-coupled. However, in contrast to the earlier weak formulations (56) and (50), there exist three significant differences:

- There is no need to solve a mesh equation (50) here since two distinct meshes are employed: a static Eulerian mesh for the augmented fluid domain Ω and an updated Lagrangian mesh for the solid domain Ω_t^s.
- Due to the definition of solid displacement and background velocity on different meshes (varied finite element spaces), the velocity must be interpolated to the solid mesh in order to apply the Lagrange multiplier. We may employ any of the interpolation (or distribution) functions introduced in Section 4.1 for the IFEM methods. Let $\mathbb{P}(\cdot)$ represent interpolation from the background mesh Ω to the solid mesh Ω_t^s, and $\mathbb{P}^\top(\cdot)$ denote distribution from the solid mesh Ω_t^s to the background mesh Ω.
- Since solid displacement is also a part of the solution, there exists a consistency equation (45), which, in contrast to the previous equation (49) integrated on Γ_t, is integrated across the entire domain Ω_t^s.

For simplicity, we assume that the solid has the same viscosity as the fluid. Readers seeking a more general case may refer to [63]. The weak form shares the same structure as (56), but we modify it to integrate solely within the domains Ω and Ω_t^s:

$$\begin{aligned}
&\int_\Omega \rho^f \mathcal{D}_t\mathbf{u}\cdot\delta\mathbf{u} + \int_{\Omega_t^s} \rho^\delta \mathcal{D}_t\mathbf{u}\cdot\delta\mathbf{u} + \frac{\mu^f}{2}\int_\Omega \mathrm{D}\mathbf{u} : \mathrm{D}\delta\mathbf{u} \\
&- \int_\Omega p\nabla\cdot\delta\mathbf{u} - \int_\Omega \delta p\nabla\cdot\mathbf{u} + \int_{\Omega_t^s} \boldsymbol{\tau}^s : \nabla\delta\mathbf{u} \\
&= \int_\Omega \rho^f \mathbf{g}\cdot\delta\mathbf{u} + \int_{\Omega_t^s} \rho^\delta \mathbf{g}\cdot\delta\mathbf{u} + \int_{\Gamma_N} \bar{\mathbf{h}}\cdot\delta\mathbf{u},
\end{aligned} \tag{74}$$

where $\rho^\delta = \rho^s - \rho^f$. Consider the consistency equation (45), which can be enforced in variational form by introducing a Lagrange multiplier as follows:

$$\int_{\Omega_t^s} (\partial_t\mathbf{d} - \mathbb{P}(\mathbf{u}))\cdot\delta\mathbf{L}(\mathbf{x}) = 0. \tag{75}$$

Introducing the Lagrange multiplier, we derive the following three equations based on the equation (74):

$$\begin{aligned}
&\int_\Omega \rho^f \mathcal{D}_t\mathbf{u}\cdot\delta\mathbf{u} + \frac{\mu^f}{2}\int_\Omega \mathrm{D}\mathbf{u} : \mathrm{D}\delta\mathbf{u} - \int_\Omega p\nabla\cdot\delta\mathbf{u} \\
&- \int_\Omega \mathbb{P}^\top(\mathbf{L}(\mathbf{x}))\cdot\delta\mathbf{u} = \int_{\Gamma_N} \bar{\mathbf{h}}\cdot\delta\mathbf{u} + \int_\Omega \rho^f\mathbf{g}\cdot\delta\mathbf{u},
\end{aligned} \tag{76}$$

$$-\int_\Omega \delta p\nabla\cdot\mathbf{u} = 0, \tag{77}$$

and

$$\begin{aligned}&\int_{\Omega_t^s}\rho^\delta\frac{\partial\mathbf{u}^s}{\partial t}\cdot\delta\mathbf{u}+\mu^s\int_{\Omega_t^s}J^{-1}\left(\mathbf{F}^\top\mathbf{F}-\mathbf{I}\right):\nabla\delta\mathbf{u}\\&+\int_{\Omega_t^s}\mathbf{L}(\mathbf{x})\cdot\delta\mathbf{u}=\int_{\Omega_t^s}\rho^\delta\mathbf{g}\cdot\delta\mathbf{u},\end{aligned}\tag{78}$$

by inserting the expression of $\boldsymbol{\tau}^s$ from (20) into (78). Let $\delta\mathbf{d}=\Delta t\delta\mathbf{u}\left(\mathbf{x}\left(\mathbf{X},t\right)\right)$, equation (78) may be expressed, by integral transformation, as:

$$\begin{aligned}&\int_{\Omega_0^s}\rho^\delta\frac{\partial^2\mathbf{d}}{\partial t^2}\cdot\frac{\delta\mathbf{d}}{\Delta t}+\mu^s\int_{\Omega_0^s}\mathbf{F}:\nabla_{\mathbf{X}}\frac{\delta\mathbf{d}}{\Delta t}-\mu^s\int_{\Omega_t^s}J^{-1}\nabla\cdot\frac{\delta\mathbf{d}}{\Delta t}\\&+\int_{\Omega_0^s}\mathbf{L}\left(\mathbf{x}\left(\mathbf{X},\mathbf{t}\right)\right)\cdot\frac{\delta\mathbf{d}}{\Delta t}=\int_{\Omega_0^s}\rho^\delta\mathbf{g}\cdot\frac{\delta\mathbf{d}}{\Delta t}.\end{aligned}\tag{79}$$

Notice that by introducing Δt in the above, we can obtain a symmetric linear system. From the perspective of virtual work or variations, $\delta\mathbf{d}$ and $\Delta t\delta\mathbf{u}$ have the same physical units.

Remark 10. *In the continuous case, the projection (or interpolation) $\mathbb{P}$ can be interpreted as an identity when restricted to domain Ω_t^s, i.e., $\forall\mathbf{u}\in\Omega$, $\mathbb{P}\left(\mathbf{u}\right)=\mathbf{u}|_{\Omega_t^s}=\mathbf{u}^s$. Regarding the distribution $\mathbb{P}^\top$, $\forall\mathbf{L}\in\Omega_t^s$ (where $\mathbf{L}$ can be interpreted as a force vector defined in domain Ω_t^s), $\mathbb{P}^\top\left(\mathbf{L}\right)\big|_{\Omega_t^s}=\mathbf{L}$ and $\mathbb{P}^\top\left(\mathbf{L}\right)\big|_{\Omega_t^f}=\mathbf{0}$. It's important to note that the sum of the three equations (76), (77), and (79) results in equation (74).*

Using the Characteristic-Galerkin time discretisation scheme discussed from (51) to (53) in Section 3.2, with the mesh velocity being $\mathbf{w}=\mathbf{u}_n^s$, we derive the following weak form based on equations (75) to (79).

$$\begin{aligned}&\int_\Omega\rho^f\frac{\mathbf{u}-\mathbf{u}_n\circ\mathbb{Y}_n}{\Delta t}\cdot\delta\mathbf{u}+\frac{\mu^f}{2}\int_\Omega\mathrm{D}\mathbf{u}:\mathrm{D}\delta\mathbf{u}\\&-\int_\Omega p\nabla\cdot\delta\mathbf{u}-\int_\Omega\delta p\nabla\cdot\mathbf{u}+\frac{\rho^\delta}{\Delta t}\int_{\Omega_0^s}\frac{\mathbf{d}-2\mathbf{d}_n+\mathbf{d}_{n-1}}{\Delta t^2}\cdot\delta\mathbf{d}\\&+\frac{\mu^s}{\Delta t}\int_{\Omega_0^s}\mathbf{F}:\nabla_{\mathbf{X}}\delta\mathbf{d}+\frac{1}{\Delta t}\int_{\Omega_0^s}\mathbf{L}\left(\mathbf{x}\left(\mathbf{X},\mathbf{t}\right)\right)\cdot\delta\mathbf{d}-\int_{\Omega_{n+1}^s}\mathbb{P}^\top\left(\mathbf{L}(\mathbf{x})\right)\cdot\delta\mathbf{u}\\&-\frac{\mu^s}{\Delta t}\int_{\Omega_{n+1}^s}J^{-1}\nabla\cdot\delta\mathbf{u}+\int_{\Omega_0^s}\frac{\mathbf{d}-\mathbf{d}_n}{\Delta t}\cdot\delta\mathbf{L}\left(\mathbf{x}\left(\mathbf{X},t\right)\right)-\int_{\Omega_{n+1}^s}\mathbb{P}(\mathbf{u})\cdot\delta\mathbf{L}(\mathbf{x})\\&=\int_{\Gamma_N}\bar{\mathbf{h}}\cdot\delta\mathbf{u}+\int_\Omega\rho^f\mathbf{g}\cdot\delta\mathbf{u}+\frac{1}{\Delta t}\int_{\Omega_0^s}\rho^\delta\mathbf{g}\cdot\delta\mathbf{d}.\end{aligned}\tag{80}$$

The above equation (80) leads to a linear system as follows:

$$\begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} & \mathbf{B}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{12}^\top & \mathbf{F}_{22} & \mathbf{B}_2 & \mathbf{0} & \mathbf{P}\mathbf{C}_1 \\ \mathbf{B}_1^\top & \mathbf{B}_2^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S} & \mathbf{C}_2 \\ \mathbf{0} & \mathbf{C}_1^\top \mathbf{P}^\top & \mathbf{0} & \mathbf{C}_2^\top & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{u}^f \\ \mathbf{u}^s \\ \mathbf{p} \\ \mathbf{d} \\ \mathbf{L} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{0} \\ \mathbf{g}^s \\ \mathbf{0} \end{pmatrix}. \tag{81}$$

In the above, $\mathbf{u}^f$ includes the velocity components in domain Ω_t^f, and $\mathbf{u}^s$ represents the remaining velocity components, which may also encompass some components in domain Ω_t^f because the interface is not fitted. $\mathbf{d}$ denotes the solid displacement in domain Ω_t^s. Matrices $\begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \mathbf{F}_{12}^\top & \mathbf{F}_{22} \end{bmatrix}$ and $\mathbf{S}$ represent the discretisation of fluid velocity integrals and solid displacement integrals, respectively, encompassing mass matrices and stiffness matrices. The matrix $\begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{pmatrix}$ indicates the coupling between fluid velocity and pressure. Matrix $\mathbf{P}\mathbf{C}_1$ arises from the coupling between the fictitious fluid velocity $\mathbf{u}^s$ and the Lagrange multiplier $\mathbf{L}$. Matrix $\mathbf{C}_2$ arises from the coupling between the solid displacement $\mathbf{d}$ and the Lagrange multiplier $\mathbf{L}$. Note that the matrix $\mathbf{P}$ represents a discretisation of the interpolation function $\mathbb{P}(\cdot)$.

4.4 One-velocity fictitious domain method (Method 8)

In the one-velocity fictitious domain method, the solid displacement $\mathbf{d}$ is expressed as a function of velocity $\mathbf{u}^s$ in the foreground solid domain, which should align with the velocity $\mathbf{u}$ of the fictitious fluid in the background domain. The need for utilising the Lagrange multiplier to enforce this consistency is not necessary. Instead, the interpolation $\mathbf{u}^s = \mathbb{P}(\mathbf{u})$ directly ensures this consistency. Similar to the FDM/DLM, we assume the solid possesses the same viscosity as the fluid, although a general viscosity term may be introduced into the constitutive equation of the solid [81, 82]. The weak form of this one-velocity FDM method remains identical to (74). By substituting the incompressible neo-Hookean solid model (20) into (74), we obtain:

$$\begin{aligned} &\int_\Omega \rho^f \mathcal{D}_t \mathbf{u} \cdot \delta\mathbf{u} + \int_{\Omega_t^s} \rho^\delta \mathcal{D}_t \mathbf{u} \cdot \delta\mathbf{u} + \frac{\mu^f}{2} \int_\Omega \mathrm{D}\mathbf{u} : \mathrm{D}\delta\mathbf{u} \\ &- \int_\Omega p \nabla \cdot \delta\mathbf{u} - \int_\Omega \delta p \nabla \cdot \mathbf{u} + \mu^s \int_{\Omega_t^s} J^{-1} \left(\mathbf{F}\mathbf{F}^\top - \mathbf{I}\right) : \nabla \delta\mathbf{u} \\ &= \int_\Omega \rho^f \mathbf{g} \cdot \delta\mathbf{u} + \int_{\Omega_t^s} \rho^\delta \mathbf{g} \cdot \delta\mathbf{u} + \int_{\Gamma_N} \bar{\mathbf{h}} \cdot \delta\mathbf{u}. \end{aligned} \tag{82}$$

Here, we use the $\mathbf{F}$-scheme, as discussed in Section 3.3, to illustrate the one-velocity method. However, alternatives such as the $\boldsymbol{\tau}$-scheme (61) or the $\mathbf{d}$-scheme (62) in 2D and (63) in 3D could be employed similarly. Hence, we conduct the following integral

transforms:

$$\int_{\Omega_t^s} J^{-1}\mathbf{F}\mathbf{F}^\top : \nabla\delta\mathbf{u} = \int_{\Omega_t^s} J^{-1}\mathbf{F} : (\nabla\delta\mathbf{u}\mathbf{F}) = \int_{\Omega_0^s} \mathbf{F} : (\nabla_{\mathbf{X}}\delta\mathbf{u}), \tag{83}$$

and

$$\int_{\Omega_t^s} \rho^\delta \mathbf{g}\cdot\delta\mathbf{u} = \int_{\Omega_t^s} J^{-1}\rho_0^\delta \mathbf{g}\cdot\delta\mathbf{u} = \int_{\Omega_0^s} \rho_0^\delta \mathbf{g}\cdot\delta\mathbf{u}, \tag{84}$$

where $\rho_0^\delta = \rho_0^s - \rho_0^f$. Since both the fluid and solid are incompressible, maintaining a constant density across the domain for all the time, there's no practical necessity to differentiate between ρ_0^s (or ρ_0^f) and ρ^s (or ρ^f). However, in the case of a numerical scheme unable to precisely ensure incompressibility ($J = 1$), the adoption of the integral within the reference domain is more accurate.

After time discretisation, the solid's coordinates can be updated using $\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t\mathbf{u}_{n+1}$. Consequently, the update for $\mathbf{F}$ becomes $\mathbf{F}_{n+1} = \mathbf{F}_n + \Delta t\nabla_{\mathbf{X}}\mathbf{u}_{n+1}$, thereby

$$\int_{\Omega_0^s} \mathbf{F}_{n+1} : \nabla_{\mathbf{X}}\delta\mathbf{u} = \int_{\Omega_0^s} \mathbf{F}_n : \nabla_{\mathbf{X}}\delta\mathbf{u} + \Delta t\int_{\Omega_0^s} \nabla_{\mathbf{X}}\mathbf{u}_{n+1} : \nabla_{\mathbf{X}}\delta\mathbf{u}. \tag{85}$$

Substituting equations (83), (84), and (85) into equation (82), and employing the Characteristic-Galerkin time discretisation scheme from (51) to (53) with the mesh velocity being $\mathbf{w} = \mathbf{u}_n^s$, yields the following weak form after time discretisation:

$$\begin{aligned}
&\int_\Omega \rho^f \frac{\mathbf{u} - \mathbf{u}_n\circ\mathbb{Y}_n}{\Delta t}\cdot\delta\mathbf{u} + \frac{\mu^f}{2}\int_\Omega \mathrm{D}\mathbf{u} : \mathrm{D}\delta\mathbf{u} - \int_\Omega p\nabla\cdot\delta\mathbf{u} - \int_\Omega \delta p\nabla\cdot\mathbf{u} \\
&+ \int_{\Omega_0^s} \rho^\delta \frac{\mathbf{u} - \mathbf{u}_n}{\Delta t}\cdot\delta\mathbf{u} + \Delta t\int_{\Omega_0^s} \mu^s\nabla_{\mathbf{X}}\mathbf{u} : \nabla_{\mathbf{X}}\delta\mathbf{u} = \int_{\Omega_{n+1}^s} \mu^s J^{-1}\nabla\cdot\delta\mathbf{u} \\
&- \int_{\Omega_0^s} \mu^s\mathbf{F}_n\nabla_{\mathbf{X}}\delta\mathbf{u} + \int_\Omega \rho^f\mathbf{g}\cdot\delta\mathbf{u} + \int_{\Omega_0^s} \rho^\delta\mathbf{g}\cdot\delta\mathbf{u} + \int_{\Gamma_N} \bar{\mathbf{h}}\cdot\delta\mathbf{u}.
\end{aligned} \tag{86}$$

After spatial discretisation, the following linear system of equations can be obtained:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix}, \tag{87}$$

where

$$\mathbf{A} = \mathbf{M}/\Delta t + \mathbf{K} + \mathbf{P}^\top(\mathbf{M}^s/\Delta t + \mathbf{K}^s)\mathbf{P}, \tag{88}$$

and

$$\mathbf{b} = \mathbf{f} + \mathbf{P}^\top\mathbf{f}^s + \mathbf{M}\mathbf{u}_n/\Delta t + \mathbf{P}^\top\mathbf{M}^s\mathbf{D}\mathbf{u}_n/\Delta t. \tag{89}$$

In the above, $\mathbf{M}$ and $\mathbf{K}$ represent the mass and stiffness matrices, respectively, assembled from the background fluid mesh, while $\mathbf{M}^s$ and $\mathbf{K}^s$ denote the mass and stiffness matrices, respectively, assembled from the foreground solid mesh. $\mathbf{f}$ and $\mathbf{f}^s$ are the force vectors derived from the right-hand side of equation (86) and assembled from the background fluid mesh and the foreground solid mesh, respectively.

4.5 Summary of the advantages and disadvantages

For methods using two meshes, the main advantage is their convenience in dealing with FSI problems involving large solid deformations. The disadvantage is that the interface between the fluid and the solid is not clearly captured, which may lead to low accuracy or stability issues – a rigorous stability analysis is challenging within the framework of classical finite element methods.
IFEM (Method 7):

- Advantages: Because all the solid information is modelled as a singular force term on the right-hand side of the fluid equation, it is easy to implement based on existing fluid codes (either finite element or finite difference). The method generally works well when the solids behave similarly to the fluids, such as very soft solids.
- Disadvantages: It is challenging to apply the method to general FSI problems, such as when the densities of the fluid and solid differ significantly (see Remark 11) and/or when the solid behaves very differently from the fluid.

Modified IFEM (Method 5.1):

- Advantages:To some extent, it retains the advantages of the classical IFEM mentioned above, with the exception of an additional step required to solve the solid equation.
- Disadvantages: The boundary condition of the solid equation is difficult to prescribe accurately.

FDM/DLM (Method 6.1):

- Advantages: This method does not require explicit tracking of the solid motion, yet it remains capable of handling FSI problems involving large solid deformation or displacement (see Fig. 11 in Section 6.2).
- Disadvantages: It requires the solution of a large system of equations involving the solid displacement, fluid velocity and pressure, as well as a Lagrange multiplier field on the solid domain. It also requires the construction of a mapping between the reference and current solid configurations in order to compute the integrals.

One-velocity FDM (Method 8):

- Advantages: This method can be viewed as a combination of IFEM and FDM/DLM: it moves as much of the solid information as possible to the left-hand side of the equations, and uses interpolation to link the two meshes instead of a Lagrange multiplier. It is designed to handle general FSI problems with a broad range of solid parameters, and it requires solving only one velocity and pressure field across the whole FSI domain.
- Disadvantages: Rigorous stability analysis is challenging.

5 Methods using one mesh without interface fitting

A single Eulerian mesh without interface fitting is widely used to solve multi-phase flow problems [162–167]. When applying such a purely Eulerian approach to fluid–structure interaction, as illustrated in Figure 5, [109] employs the level set method to capture the fluid–structure interface, together with a Lagrange multiplier and penalty method to achieve fluid–solid coupling. In [44, 84, 87, 89, 90], an Initial Points (IP) set is used to track the interface, and the unknown variables are extended to the whole domain using a characteristic function, allowing the system to be solved in a monolithic manner. A recent variant of the CutFEM methods [91] solves a fully coupled monolithic system on a single mesh, with local cutting of elements for integration. Furthermore, [112, 113] introduces a phase variable to track the fluid–solid interface and derives the governing equations from the total free energy.

Immersed boundary methods (IBMs) also fall into this category of employing a single mesh without interface fitting. Most classical IBMs are based on structural meshes, including curvilinear grids [99]. A recent variant is the finite element immersed boundary Method [62, 168], in which the fluid mesh is not restricted to be structured. We do not review IBMs in detail here, as it is closely related to the immersed finite element method presented in Section 4.1. Moreover, it is a well-established methodology, and two recent review papers already provide comprehensive coverage of this topic [2, 3].

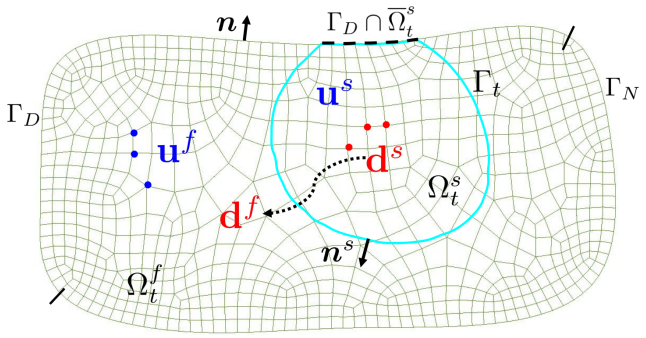

Fig. 5: Schematic diagram for a mesh without interface fitting, $\Omega = \Omega_t^f \cup \Omega_t^s$, $\partial\Omega = \Gamma_D \cup \Gamma_N$.

5.1 Fully Eulerian formulation (Method 9.1)

The finite element weak form for this method resembles the monolithic Eulerian formulation (56). However, a key distinction lies in the absence of discrete fluid or solid subdomains. Consequently, direct integration on the fluid or solid mesh is not feasible. The notation $1_{\Omega_t^f}$ or $1_{\Omega_t^s}$ holds a different significance at the discrete level because the interface has not been tracked: the IP set is utilised to capture the fluid and solid regions [84]. Introducing an alternative notation for the indicator function explicitly

clarifies the interpretation of the IP set method:

$$\chi^f(\mathbf{x}) = \begin{cases} 1, & \mathbf{x}-\mathbf{d} \in \Omega_0^f \setminus \Gamma_t \\ 0, & \mathbf{x}-\mathbf{d} \in \Omega_0^s \end{cases}, \qquad \chi^s(\mathbf{x}) = 1 - \chi^f(\mathbf{x}), \tag{90}$$

Let $\rho = \chi^f \rho^f + \chi^s \rho^s$ and $\boldsymbol{\tau} = \chi^f \boldsymbol{\tau}^f + \chi^s \boldsymbol{\tau}^s$, the weak form can then be expressed:

$$\begin{aligned} &\int_\Omega \rho \mathcal{D}_t \mathbf{u} \cdot \delta\mathbf{u} - \int_\Omega p \nabla \cdot \delta\mathbf{u} - \int_\Omega \delta p \nabla \cdot \mathbf{u} \\ &+ \int_\Omega \chi^f \frac{\mu^f}{2} \mathrm{D}\mathbf{u} : \mathrm{D}\delta\mathbf{u} + \chi^s \mu^s J^{-1} \left(\mathbf{F}^\top \mathbf{F} - \mathbf{I}\right) : \nabla \delta\mathbf{u} \\ &= \int_{\Gamma_N} \bar{\mathbf{h}} \cdot \delta\mathbf{u} + \rho \int_\Omega \mathbf{g} \cdot \delta\mathbf{u}. \end{aligned} \tag{91}$$

In the spirit of a fully Eulerian formulation, the deformation tensor may be expressed, in the current configuration, as:

$$\mathbf{F} = \nabla_{\mathbf{X}} \mathbf{d} + \mathbf{I} = \nabla \mathbf{d} \mathbf{F} + \mathbf{I} \Rightarrow \mathbf{F} = (\mathbf{I} - \nabla \mathbf{d})^{-1}. \tag{92}$$

The displacement $\mathbf{d}$ is now defined in the whole domain $\Omega = \Omega_t^f \cup \Omega_t^s$, which is a smooth extension of the solid displacement $\mathbf{d}$ in Ω_t^s as shown in Figure 5. $\mathcal{D}_t(\mathbf{d}1_{\Omega_t^s}) = \mathbf{u}^s$, but $\mathcal{D}_t(\mathbf{d}1_{\Omega_t^f}) \neq \mathbf{u}^f$ generally. Let

$$\mathcal{D}_t \mathbf{d} = \partial_t \mathbf{d} + (\mathbf{v} \cdot \nabla)\, \mathbf{d} = \mathbf{v}, \tag{93}$$

then the extension is defined by velocity $\mathbf{v}$ as follows (harmonic continuation of the solid velocity):

$$\int_\Omega \chi^s \left(\mathbf{u}^s - \mathbf{v}\right) \cdot \delta\mathbf{v} + \alpha_{\mathbf{u}} \int_\Omega \chi^f \nabla \mathbf{v} : \nabla \delta\mathbf{v} = 0, \tag{94}$$

where $\alpha_{\mathbf{u}}$ is a small positive constant. Usually choosing $\alpha_{\mathbf{u}} = \alpha_{\mathbf{u}}^0 h^2$ and $\alpha_p = \alpha_p^0 h$, where h is the local mesh size with $\alpha_{\mathbf{u}}^0 = \alpha_p^0 \sim 0.01$ [87].

This fully Eulerian formulation appears elegant; however, the challenge lies in accurately performing the integration. Adaptive local mesh refinement may be required for precise quadrature. Additionally, it necessitates solving for the enlarged velocity $\mathbf{u}$ and displacement $\mathbf{d}$ fields, along with an additional velocity field $\mathbf{v}$.

5.2 CutFEM (Method 9.2)

There are different versions of CutFEM-based methods, as shown in Table 1. Here, we first explain the idea of CutFEM using the one-velocity-field formulation (56): For meshes that do not fit the interface, as shown in Figure 5, one question is how to compute the integrals over Ω_t^s and Ω_t^f after spatial discretisation.

The idea of CutFEM is to cut the local elements near the interface into smaller cells, as illustrated by the red triangles in Figure 6 (a zoomed-in view of Figure 5), and then compute the integrals over these small cells. Note that the shape functions are

also cut (i.e. projected onto the cut elements). Consequent problems arise from cutting local elements: the cut triangles may be very small, leading to a poorly conditioned matrix system or to stability issues.

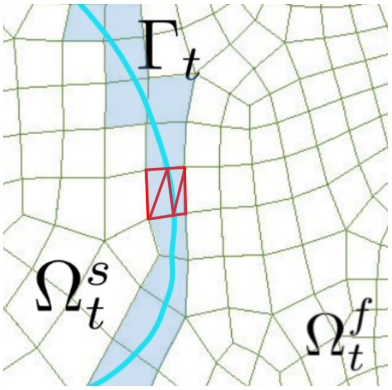

Fig. 6: Mesh Ω_h, which is a discretisation of Ω, without fitting the fluid-solid interface Γ_t.

We now introduce the idea of stabilisation in CutFEM methods based on the fully Eulerian formulation in [91, 169], solving for the fluid velocity $\mathbf{u}^f$ and pressure p, and the solid velocity $\mathbf{u}^s$ and displacement $\mathbf{d}$. The weak form for the fluid is given by (26) and (27), with $\mathbf{w} = 0$ for a fully Eulerian setting:

$$\begin{aligned}&\int_{\Omega_t^f} \rho^f \partial_t \mathbf{u}^f \cdot \delta\mathbf{u}^f + \int_{\Omega_t^f} \rho^f \left(\mathbf{u}^f \cdot \nabla\right) \mathbf{u}^f \cdot \delta\mathbf{u}^f - \int_{\Omega_t^f} \delta p \nabla \mathbf{u}^f \\ &= \int_{\Gamma_t} \left(\boldsymbol{\sigma}^f \cdot \mathbf{n}^f\right) \cdot \delta\mathbf{u}^f + \int_{\Gamma_N} \bar{\mathbf{h}} \cdot \delta\mathbf{u}^f - \int_{\Omega_t^f} \boldsymbol{\sigma}^f : \nabla\delta\mathbf{u}^f + \int_{\Omega_t^f} \rho^f \mathbf{g} \cdot \delta\mathbf{u}^f,\end{aligned} \tag{95}$$

The weak form for the solid is similar to (31), but we now solve for $\mathbf{u}^s$ and perform the integration in the current configuration, with an additional convection term:

$$\begin{aligned}&\int_{\Omega_t^s} \rho^s \partial_t \mathbf{u}^s \cdot \delta\mathbf{u}^s + \int_{\Omega_t^s} \rho^s \left(\mathbf{u}^s \cdot \nabla\right) \mathbf{u}^s \cdot \delta\mathbf{u}^s + \int_{\Omega_t^s} \boldsymbol{\sigma}^s : \nabla\delta\mathbf{u}^s \\ &= \int_{\Gamma_t} \left(\boldsymbol{\sigma}^s \cdot \mathbf{n}^s\right) \cdot \delta\mathbf{u}^s + \int_{\Omega_t^s} \rho^s \mathbf{g} \cdot \delta\mathbf{u}^s,\end{aligned} \tag{96}$$

together with equation (93) for the solid displacement, written in its weak form:

$$\int_{\Omega_t^s} \partial_t \mathbf{d} \cdot \delta\mathbf{d} + \int_{\Omega_t^s} \left(\mathbf{u}^s \cdot \nabla\right) \mathbf{d} \cdot \delta\mathbf{d} = \int_{\Omega_t^s} \mathbf{u}^s \cdot \delta\mathbf{d}. \tag{97}$$

The stabilisation involves the use of Nitsche's method [169, 170] to weakly enforce the interface conditions (5) and (6), together with the so-called ghost penalty terms

defined on the interface elements – the shaded elements in Figure 6. This results in a final monolithic formulation, obtained by rearranging (95), (96) and (97) together as:

$$
\begin{aligned}
&\int_{\Omega_t^f} \rho^f \partial_t \mathbf{u}^f \cdot \delta\mathbf{u}^f + \int_{\Omega_t^f} \rho^f \left(\mathbf{u}^f \cdot \nabla\right) \mathbf{u}^f \cdot \delta\mathbf{u}^f + \int_{\Omega_t^f} \boldsymbol{\sigma}^f : \nabla\delta\mathbf{u}^f - \int_{\Omega_t^f} \delta p \nabla \mathbf{u}^f \\
&+ \int_{\Omega_t^s} \rho^s \partial_t \mathbf{u}^s \cdot \delta\mathbf{u}^s + \int_{\Omega_t^s} \rho^s \left(\mathbf{u}^s \cdot \nabla\right) \mathbf{u}^s \cdot \delta\mathbf{u}^s + \int_{\Omega_t^s} \boldsymbol{\sigma}^s : \nabla\delta\mathbf{u}^s \\
&- \int_{\Gamma_t} \left(\boldsymbol{\sigma}^f \left(\mathbf{u}^f, p\right) \cdot \mathbf{n}^f\right) \cdot \left(\delta\mathbf{u}^f - \delta\mathbf{u}^s\right) - \int_{\Gamma_t} \left(\boldsymbol{\sigma}^f \left(\delta\mathbf{u}^f, -\delta p\right) \cdot \mathbf{n}^f\right) \cdot \left(\mathbf{u}^f - \mathbf{u}^s\right) \\
&+ \int_{\Gamma_t} \alpha \mu^f h^{-1} \left(\mathbf{u}^f - \mathbf{u}^s\right) \cdot \left(\delta\mathbf{u}^f - \delta\mathbf{u}^s\right) + \int_{\Omega_t^f} \beta \mu^f h^{-2} \nabla p \cdot \nabla \delta p \\
&+ \int_{\Omega_t^s} \left(\partial_t \mathbf{d} + \left(\mathbf{u}^s \cdot \nabla\right) \mathbf{d} - \mathbf{u}^s\right) \cdot \delta\mathbf{d} + g_h(\mathbf{u}^f, \mathbf{u}^s, \mathbf{d}, p; \delta\mathbf{u}^f, \delta\mathbf{u}^s, \delta\mathbf{d}, \delta p) \\
&= \int_{\Gamma_N} \bar{\mathbf{h}} \cdot \delta\mathbf{u}^f + \int_{\Omega_t^f} \rho^f \mathbf{g} \cdot \delta\mathbf{u}^f + \int_{\Omega_t^s} \rho^s \mathbf{g} \cdot \delta\mathbf{u}^s.
\end{aligned}
\tag{98}
$$

In the above, the term with coefficient α is the penalty term arising from the use of Nitsche's method to impose the interface conditions, and the term with coefficient β provides pressure stabilisation, where h is the maximal mesh size. The quantity g_h represents all ghost penalty terms, introduced to enhance robustness of the discretised matrix system; these are defined on the locally cut elements around the interface, denoted by $\mathcal{G}_h$, as shown by the shaded elements in Figure 6. Let $\mathcal{F}_G$ be the set of faces associated with $\mathcal{G}_h$. This means that for every face $F \in \mathcal{F}_G$, there exist elements K and K' such that $F = K \cap K'$, with at least one of them belonging to $\mathcal{G}_h$. The ghost penalty g_h is given by

$$
g_h = \sum_{F \in \mathcal{F}_G} \int_F \left(\sum_{i=1}^{3} \gamma_i [\![\tilde{\partial}_n \mathbf{v}^i]\!] \cdot [\![\tilde{\partial}_n \delta\mathbf{v}^i]\!] + \gamma_4 [\![\tilde{\partial}_n p]\!] \cdot [\![\tilde{\partial}_n \delta p]\!] \right). \tag{99}
$$

where $\mathbf{v}^0 = \mathbf{d}$, $\mathbf{v}^1 = \mathbf{u}^f$, $\mathbf{v}^2 = \mathbf{u}^s$, and γ_i $(i = 1, 2, 3)$ are penalty parameters depending on the local mesh size in the region $\mathcal{G}_h$ (see [112, 113] for details). $[\![\tilde{\partial}_n(\cdot)]\!] = \mathbf{n}_F \cdot \nabla(\cdot)|_K - \mathbf{n}_F \cdot \nabla(\cdot)|_{K'}$ denotes the jump of the normal gradient operator.

5.3 Phase-field methods (Method 11)

As commonly adopted in the literature on phase-field methods for multi-phase fluids, a phase variable $0 \leq \phi(\mathbf{x}, t) \leq 1$ is introduced to regularise the coefficients and variables in this type of FSI methods. This phase variable resembles the characteristic function mentioned in (90). Instead of employing the IP set or level-set method, it solves an additional phase equation to track the fluid-solid interface, such as the Allen–Cahn type phase-field equation [112], expressed in its weak form as:

$$
\int_\Omega \mathcal{D}_t \phi \delta\phi = \int_\Omega \left(\partial_t \phi + \mathbf{u} \cdot \nabla\phi\right) \delta\phi = -\delta F(\phi, \xi; \delta\phi), \tag{100}
$$

with

$$F(\phi,\xi)=\int_\Omega \frac{\beta}{2}|\nabla\phi|^2+\gamma W(\phi)-\xi(\int_\Omega \phi-C) \tag{101}$$

being the free energy, where $W(\phi)=(\phi-1)^2\phi^2$ represents the Ginzburg–Landau double-well potential, and C is a constant representing the area or volume of the solid. The parameters β and γ regulate the interface; a larger β smoothens the interface, while a larger γ sharpens it. Additionally, we introduce a Lagrange multiplier ξ in (101) to enforce the following equation, ensuring the conservation of the entire volume:

$$\int_\Omega \mathcal{D}_t\phi=\int_\Omega(\partial_t\phi+\mathbf{u}\cdot\nabla\phi)=0. \tag{102}$$

Note that the aforementioned equation (102) does not conflict with the phase equation (100). Since ξ is a scalar, the corresponding weak form of (102) is

$$-\delta F(\phi,\xi;\delta\xi)=\delta\xi\int_\Omega(\partial_t\phi+\mathbf{u}\cdot\nabla\phi)=0, \tag{103}$$

Let $\phi=1_{\Omega_t^s}$, $\rho=\phi\rho^s+(1-\phi)\rho^f$, and $\mathbf{u}=\phi\mathbf{u}^s+(1-\phi)\mathbf{u}^f$. Similar to equation (91), but utilising the **d**-scheme (62), a phase-field formulation for FSI problems may be expressed as follows [112, 113].

$$\begin{aligned}&\int_\Omega \rho\mathcal{D}_t\mathbf{u}\cdot\delta\mathbf{u}-\int_\Omega p\nabla\cdot\delta\mathbf{u}-\int_\Omega\delta p\nabla\cdot\mathbf{u}\\&+\int_\Omega(1-\phi)\frac{\mu^f}{2}\mathrm{D}\mathbf{u}:\mathrm{D}\delta\mathbf{u}+\phi\mu^s\left(\mathrm{D}\mathbf{u}-\nabla^\top\mathbf{u}\nabla\mathbf{u}\right):\nabla\delta\mathbf{u}\\&+\eta\int_\Omega(\nabla\phi\otimes\nabla\phi):\nabla\mathbf{u}=\int_{\Gamma_N}\bar{\mathbf{h}}\cdot\delta\mathbf{u}+\int_\Omega\rho\mathbf{g}\cdot\delta\mathbf{u},\end{aligned} \tag{104}$$

where η is the surface tension parameter. It can be seen that the phase equation and FSI equations are coupled with each other. The primary challenge faced by phase-field methods lies in accurately capturing the fluid-solid interface as time progresses.

5.4 Summary of the advantages and disadvantages

The advantage of using a single mesh without interface fitting lies in the convenience of employing only one mesh to model the entire FSI problem. The disadvantage is that one has to capture the interface based on an additional algorithm.
Fully Eulerian formulation (Method 9.1):

- Advantages: Elegant mathematical formulations.
- Disadvantages: Challenging to compute the integrals; adaptive local mesh refinement may be required.

CutFEM methods (Method 9.2):

- Advantages: Stability analysis is available.
- Disadvantages: The implementation is challenging.

Phase-field methods (Method 11):

- Advantages: Elegant mathematical formulations.
- Disadvantages: Local mesh refinement may be required to increase the accuracy (see numerical tests in Sections 6.2 and 6.3).

6 Numerical examples

In this section, we compare six numerical methods, mainly monolithic methods, based on three FSI benchmarks. All tests are implemented in `FreeFEM++` [171], and the corresponding code and animations are available in the public GitHub repository: https://yongxingwang.github.io/fsiReview/.

6.1 Oscillating flag

This FSI example was originally proposed in [172], and has since been widely used to compare or validate numerical methods within the FSI community. We employ it to compare the one-velocity FDM (Method 8), the one-velocity monolithic ALE method (Method 4), and the monolithic ALE approach using a Lagrange multiplier at the interface (Method 2.1).

The computational domain is a rectangle ($L \times H$) with a circular hole of radius r, centred at (c, c), as shown in Fig. 7. The geometric parameters are: $L = 2.5$, $H = 0.41$, $l = 0.35$, $h = 0.02$, $c = 0.2$, and $r = 0.05$. The fluid and solid parameters are: $\rho^f = \rho^s = 10^3$, $\mu^f = 1$, and $c_1 = 2.0 \times 10^6$. The inlet flow is prescribed as:

$$u_x = \frac{12y}{H^2}(H - y), \quad u_y = 0. \tag{105}$$

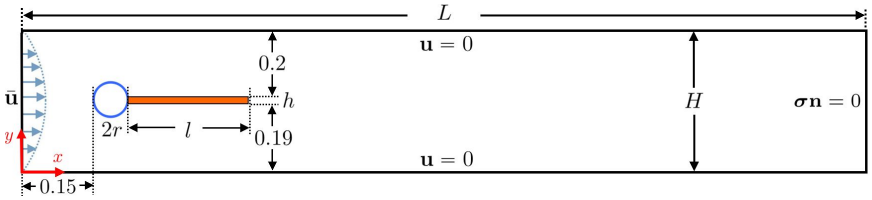

Fig. 7: Computational domain and boundary conditions for the oscillating flag.

We compare the three methods using the same mesh of 6207 triangles with 3025 vertices for both fluid and solid domains, as can be seen in Fig. 8. For the one-velocity FDM method, an additional solid mesh with 425 triangles and 213 vertices is adopted. A converged time step of $\Delta t = 10^{-3}$ is used for all the three tests – converged time in the sense that both the frequency and magnitude of the oscillation at the tip of the flag would not change if the time steps are further reduced. As shown in Fig. 8 and 9, all the methods produce similar oscillation panterns: almost identical periods of oscillation ($5.3 \times 10^{-3} s$) although the magnitudes are slightly different. Here we focus on the comparison of three methods using the same mesh. Readers may refer to [53] for the test of mesh convergence.

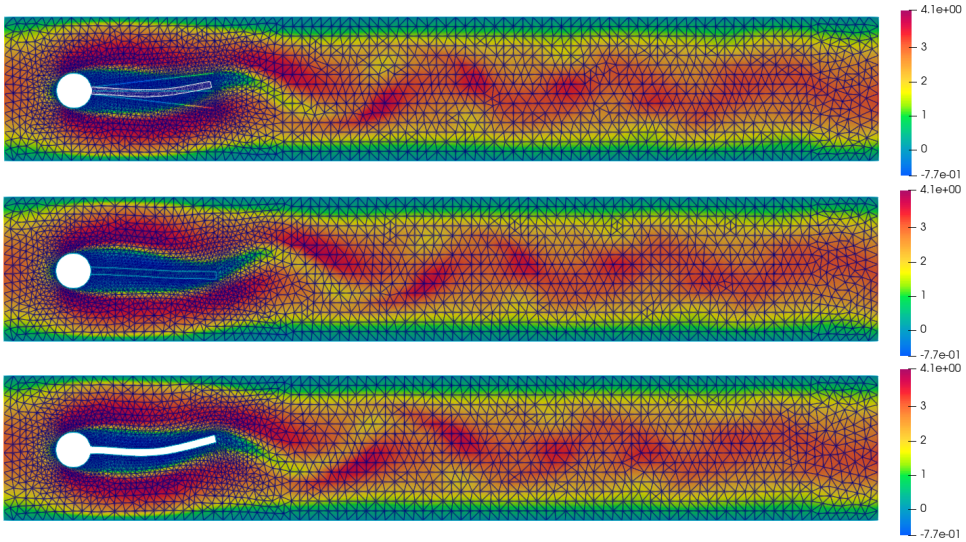

Fig. 8: Horizontal velocity at $t = 6$ for three methods, from top to bottom: the one-velocity FDM (Method 8), the one-velocity monlithic ALE method (Method 4) and the Monolithic ALE via a Lagrange multiplier at the interface (Method 2.1).

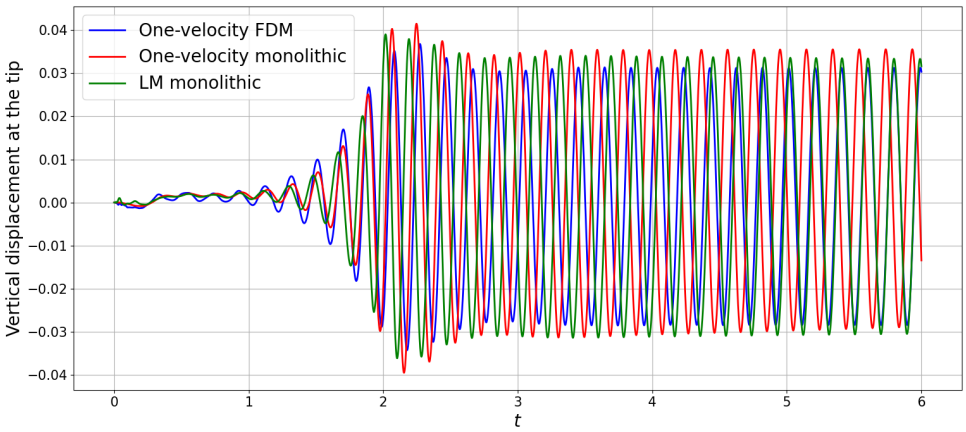

Fig. 9: Vertical displacement at the tip of the flag for different methods.

Among these three methods, the one-velocity monolithic ALE method (Method 4) outperforms the other two in terms of accuracy and efficiency, as it solves fewer degrees of freedom based on an interface-fitted mesh. Although we did not test mesh convergence, when using the same mesh, the magnitude of oscillation for the one-velocity monolithic ALE method is closer to 0.00148 ± 0.03438, as originally proposed in [172].

6.2 Falling disc

This example is widely used in the FSI community to validate numerical results against empirical formula. The computational domain is a vertical channel with a rigid disc placed at the top, as illustrated in Fig. 10, where $W = 2$, $H = 4$, $h = 0.5$, and $R = 0.25$. In this test, $\rho^f = 1$, $\rho^s = 1.2$, $\mu^f = 0.1$, and the gravitational acceleration is $g = 981$. The fluid velocity is fixed at zero on all boundaries except the top, where the normal stress is set to zero.

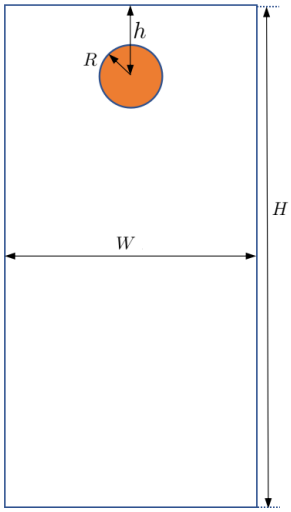

Fig. 10: Computational domain for the falling disc.

The disc falls under gravity and gradually reaches a constant velocity. An empirical formula for this terminal velocity is given by [160, 173]:

$$U = \frac{(\rho^s - \rho^f) g R^2}{4\mu^f} \left[\ln\left(\frac{W}{2R}\right) - 0.9157 + 1.7244 \left(\frac{2R}{W}\right)^2 - 1.7302 \left(\frac{2R}{W}\right)^4 \right]. \quad (106)$$

We compare four numerical methods: the one-velocity monolithic method with remeshing (Method 4), the one-velocity FDM (Method 8), the FDM with a distributed Lagrange multiplier (Method 6.1), and the phase-field method (Method 11). The test is performed on a relatively coarse mesh – 40 segments along the boundary of the solid disc, as shown in Fig. 11, where the vertical velocity field is plotted on the current mesh.

For both FDM-based methods, namely the one-velocity FDM and the FDM/DLM, a separate solid mesh is required for the simulation. The difference between them is that the former updates the solid mesh during the computation, whereas the latter only requires a reference solid mesh. This feature represents the main advantage of the FDM/DLM approach. Its disadvantage, however, lies in the implementation of

the Lagrange multiplier term in equation (80), which requires a mapping between the reference and current configurations.

For the phase-field method, it is not necessary to solve the phase equation (100) for this problem, as the disc is rigid. Instead, the phase can be prescribed as

$$\phi = \frac{1}{1+e^{\epsilon d}}, \quad d = \sqrt{(x-x_c)^2+(y-y_c)^2} - R, \tag{107}$$

where ϵ is a parameter controlling the width of the interface, and (x_c, y_c) denotes the centre of the disc, which is updated during the simulation.

The evolution of the vertical velocity at the centre of the solid is shown in Fig. 12. It can be observed that all results agree well with the empirical formula, except for the one obtained using the phase-field method, which shows a noticeable discrepancy. The simulation was then repeated using finer meshes, which resolved this inconsistency, as illustrated in Fig. 13. It should be noted that a very fine mesh, together with a narrow interface (achieved by choosing a large value of ϵ in (107)), is required to obtain the same level of accuracy as the other methods.

From Fig. 12, it can be observed that all the methods, except the phase-field method, exhibit a zigzag feature when the disc reaches its maximum speed. This behaviour is caused by interpolation errors. For the one-velocity monolithic method, although a single mesh is used, the remeshing process introduces this zigzag oscillation. For the other two methods, since two meshes are employed, interpolation is intrinsically involved.

6.3 Oscillating disc

This test considers a solid disc in an enclosed flow ($\mathbf{n}\cdot\mathbf{u} = 0$) within a square domain $\Omega = [0,1]\times[0,1]$, subject to periodic boundary conditions. The radius of the solid disc is 0.2, and it is located at the centre of the square. The solid begins to oscillate due to an initial velocity profile prescribed throughout the entire domain Ω. The velocity profile is defined by the following stream function:

$$\Psi(x,y) = \Psi_0 \sin(ax)\sin(by), \tag{108}$$

where $\Psi_0 = 0.05$ and $a = b = 2\pi$. In this test, $\rho^f = 1$, $\mu^f = 0.01$, $\rho^s = 1$, and $c_1 = 1$. We compare three methods: the one-velocity monolithic method with remeshing (Method 4), the immersed finite element method (IFEM, Method 7), and the phase-field method (Method 11), using meshes of similar size. The results obtained from the three methods are very similar, as shown in Fig. 14.

We examine mass conservation by plotting the area of the solid in Fig. 15, from which it can be seen that all three methods conserve mass well. However, the phase-field method performs slightly better, even though the initialisation of the phase does not conserve mass exactly, as shown in Fig. 15. This improvement arises because the Lagrange multiplier approach is employed in the free energy functional (101) to directly enforce mass conservation in the phase-field model. Note that the results of the phase-field model are sensitive to the regularisation parameters β and γ in (101). In our test, we set $\beta = 0.005$ and $\gamma = 300$.

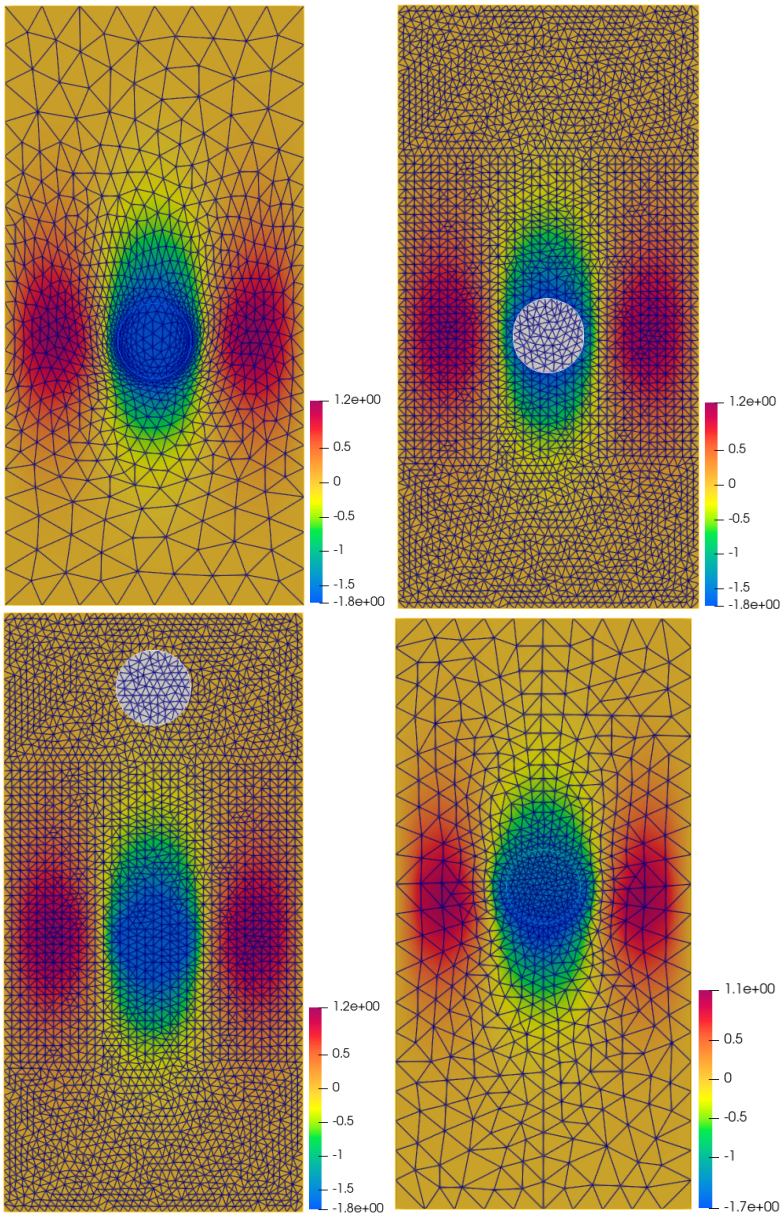

Fig. 11: Vertical velocity at $t = 1$ for four different methods, arranged from left to right and top to bottom: the one-velocity monolithic ALE method with remeshing (Method 4), the one-velocity FDM (Method 8), the FDM/DLM (Method 6.1), and the phase-field method (Method 11).

Remark 11. *We also tested a more challenging case in which the density exhibits a jump across the fluid–solid interface:* $\rho^s = 2$ *and* $\rho^f = 1$. *The IFEM fails to simulate this problem, whereas the one-velocity FDM successfully handles it. Although the*

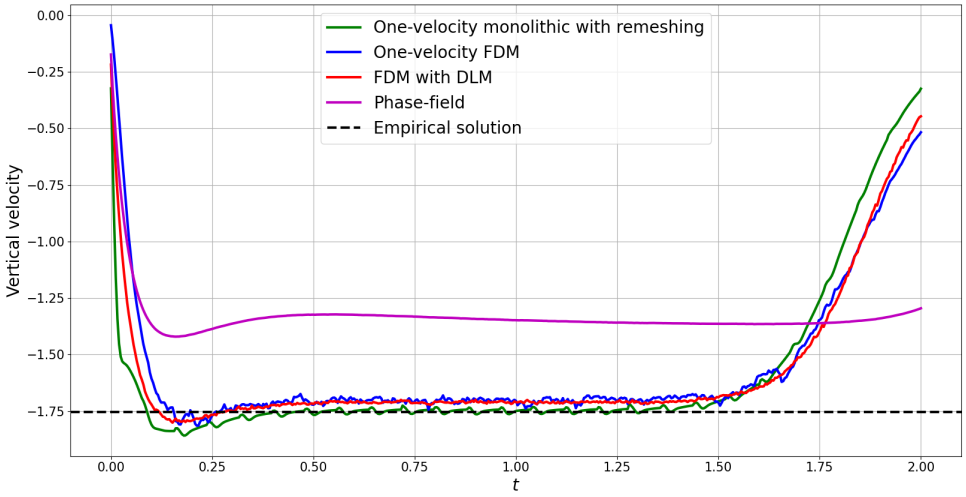

Fig. 12: Vertical velocity at the centre of the disc for different methods.

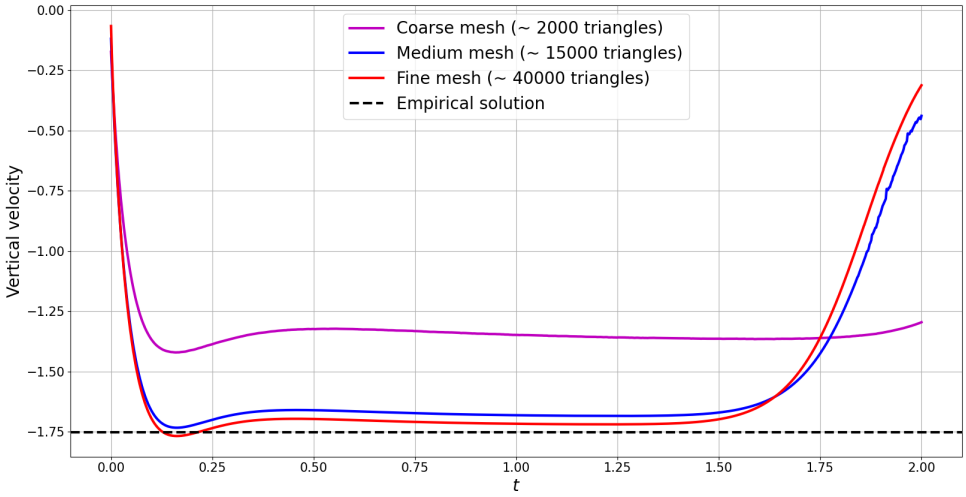

Fig. 13: Vertical velocity at the centre of the disc for three different meshes using the phase-field method. The parameter ϵ (which controls the width of the interface) in (107) is set to 100, 500, and 800 for the coarse, medium, and fine meshes, respectively.

results are not presented here, the `FreeFEM` *script and the corresponding animations are available in the public GitHub repository: https://yongxingwang.github.io/.*

7 Applications and implementations of FSI methods

We consider three levels of application for the FSI methods reviewed in this article: methods that are mature and have been implemented and widely used through

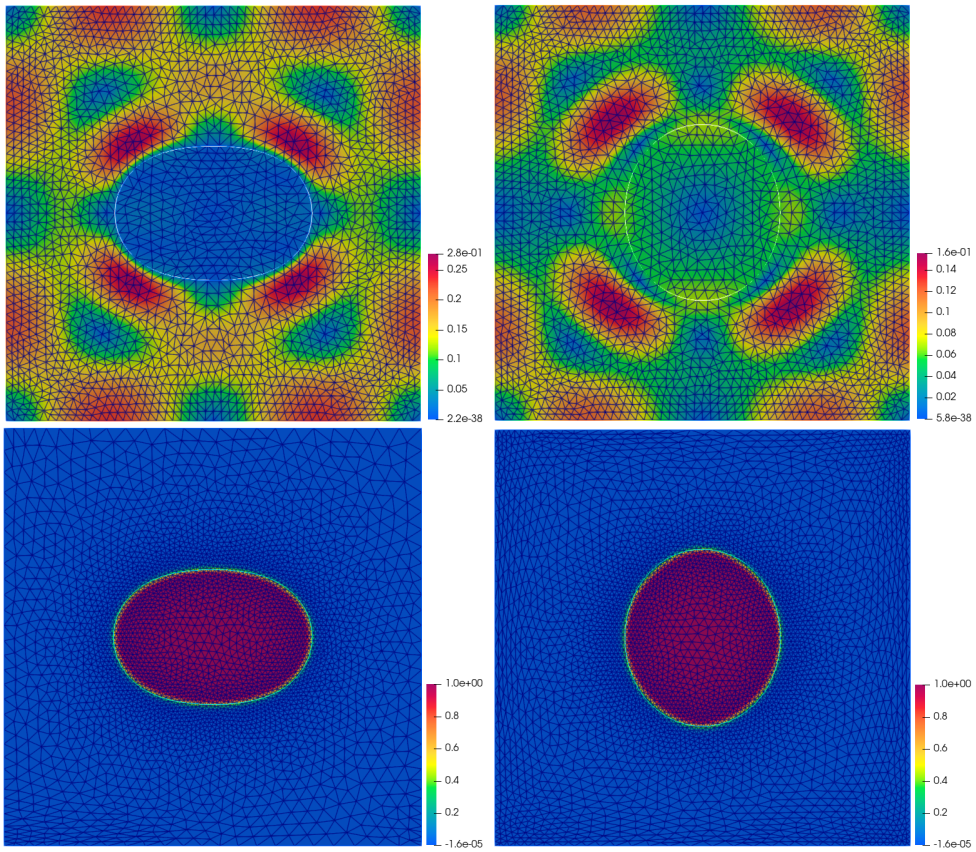

Fig. 14: Comparison among three methods when the solid is maximally deformed in the x and y directions. Top panel: Solid boundary (white), using IFEM, plotted over the velocity magnitude using the one-velocity monolithic method with remeshing (Method 4). Bottom panel: Solid boundary (green), using IFEM, plotted over the phase field using the Phase-field method (Method 11).

commercial software; methods that have been implemented and used mainly within academic groups via open-source libraries; and methods that have been proposed in academic articles and implemented only by their authors. Based on the available literature, It appears that the classical partitioned methods (Method 1), the monolithic Method 2.1, and IFEM are among the few that have reached the level of real-world application. Other FSI methods have been presented primarily in the literature and may have potential applications in specific areas, but they remain at an academic level.

Classical interface-fitted partitioned methods (Method 1) have been implemented in commercial software such as ANSYS Fluids and COMSOL, and have been used to solve a wide range of problems, from low-Reynolds micro-biomechanics [174–178] to high-Reynolds blood flow [179–182] and aerodynamics [183–188]. The monolithic ALE method (Method 2.1) has also recently been implemented in COMSOL [189, 190].

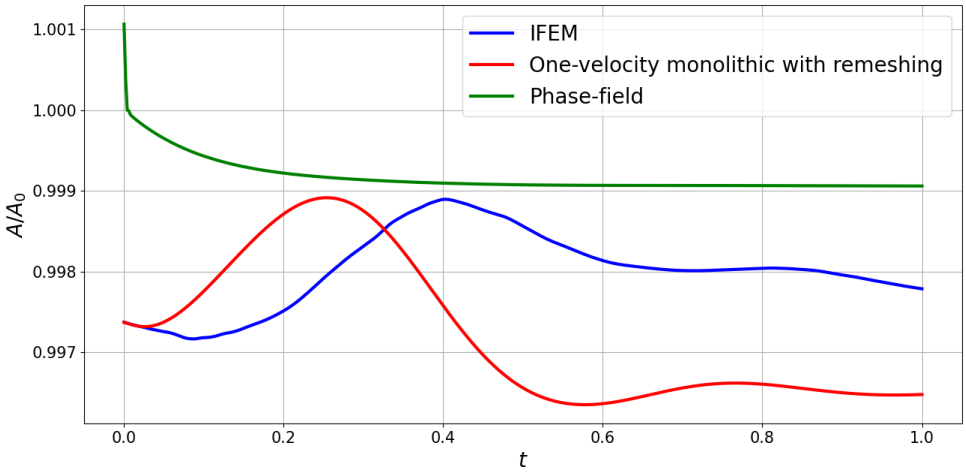

Fig. 15: The evolution of the solid's area, where A_0 denotes the initial area and A denotes the area at time t.

Immersed methods (Methods 7 and 10) have been widely applied to biological systems [3, 102, 191, 192], the modelling of composite materials [193, 194], simulations of plasma–material interactions [195, 196], the modelling of wave propagation [197, 198], turbulence simulations [199–201], and other areas.

There are many open-source libraries that are well suited to the implementation of particular methods based on their features. For example, `FreeFEM++` [171], which uses triangles in 2D and tetrahedra in 3D, is convenient for implementing a wide range of FSI methods. The partitioned method (Method 1) has been used for FSI shape optimisation [202–204]; several monolithic FSI methods have been implemented in `FreeFEM++`, including Method 4 [51], Method 5 [81], Method 6.1 [121], as well as the six methods presented in this article (see numerical example in Section 6). The monolithic ALE method (Method 2.1) has been implemented in `oomph-lib` and applied to internal physiological flows [205, 206]. Fully Eulerian methods (Method 9) and the locally modified FEM (Method 2.2) have been implemented in `deal.II` [45, 207].

There is a dedicated CutFEM library [208] designed to facilitate the implementation of CutFEM. Applications of CutFEM include shape optimisation [209, 210], the simulation of composites [211], and laser manufacturing [212]. Phase-field methods have shown potential for applications in biomechanics [114].

8 Conclusion

In this work, we have provided a detailed and structured review of numerical methods developed for solving Fluid-Structure Interaction (FSI) problems, with a primary emphasis on finite element-based techniques and a comparative acknowledgment of finite difference methods. Our classification scheme, grounded on three fundamental criteria – mesh type, coupling strategy, and variables solved – enables a systematic

categorisation of FSI methods and offers a unified framework for comparing their characteristics, strengths, and limitations.

Classical partitioned methods, although historically foundational and relatively easy to implement, continue to face challenges in stability and convergence, especially in strongly coupled regimes. Monolithic approaches, particularly those based on Arbitrary Lagrangian-Eulerian (ALE) and more recent developments such as locally modified FEM and CutFEM, offer improved robustness, especially for problems involving large deformations. Furthermore, methods that avoid interface-fitted meshes – such as the fully Eulerian, cutFEM, and immersed finite element methods – enhance computational adaptability to complex geometries.

The formulation of weak forms for each method, and their linearisation via Newton's method at the continuous level, offers a pathway for consistent implementation across general-purpose finite element platforms. Despite the diversity of techniques, this unified formulation framework helps bridge their theoretical underpinnings with practical numerical implementation.

The growing need for high-fidelity simulations in biomechanics, aerospace, and industrial processes underscores the importance of hybrid methods that combine the stability of monolithic schemes with the geometric flexibility of unfitted or immersed methods. Moreover, increased attention to computational efficiency, adaptivity, and software interoperability will continue to guide the evolution of FSI solvers. This review serves as both a reference and a guide for future research in developing and applying robust numerical strategies for FSI problems.

Appendix A Gateaux variation

Definition 1. *The first order Gateaux variation of a functional* $\mathcal{F}(\mathbf{d})$ *in the direction* $\mathbf{h}$ *is defined by [12, 154]*

$$\delta\mathcal{F}(\mathbf{d};\mathbf{h}) = \frac{d}{d\epsilon}\mathcal{F}\left(\mathbf{d}+\epsilon\mathbf{h}\right)\Big|_{\epsilon=0}, \tag{A1}$$

and the second order Gateaux variation of a functional $\mathcal{F}(\mathbf{d})$ *in the direction* $\mathbf{h}_1$ *and* $\mathbf{h}_2$ *is defined by [154, Page 157]*

$$\delta^2\mathcal{F}(\mathbf{d};\mathbf{h}_1,\mathbf{h}_2) = \frac{\partial^2}{\partial\epsilon_1\partial\epsilon_2}\mathcal{F}\left(\mathbf{d}+\epsilon_1\mathbf{h}_1+\epsilon_2\mathbf{h}_2\right)\Big|_{\epsilon_1=\epsilon_2=0}. \tag{A2}$$

If $\mathcal{F}(\mathbf{d})$ *is a vector or matrix the above derivative* $\frac{d}{d\epsilon}$ *or* $\frac{\partial^2}{\partial\epsilon_1\partial\epsilon_2}$ *is taken based on components.*

The following properties of Gateaux variation are straightforward to obtain from the above definition.

1. For two arbitrary functionals $\mathcal{F}(\cdot)$ and $\mathcal{G}(\cdot)$:

$$\delta\left(\mathcal{F}\mathcal{G}\right) = \left(\delta\mathcal{F}\right)\mathcal{G} + \mathcal{F}\left(\delta\mathcal{G}\right). \tag{A3}$$

2. For a linear functional $\mathcal{L}(\cdot)$ and an arbitrary functional $\mathcal{F}(\cdot)$:

$$\delta\left(\mathcal{L}\circ\mathcal{F}\right)=\mathcal{L}\circ\delta\mathcal{F}. \tag{A4}$$

3. For the linear functional $\mathcal{L}(\mathbf{d})=\mathbf{d}$, taking the first order variation in direction $\mathbf{h}$ and taking the second order variation in any two directions:

$$\delta\mathcal{L}=\delta\mathbf{d}=\mathbf{h},\quad \delta^2\mathcal{L}=\delta^2\mathbf{d}=0. \tag{A5}$$

4. For a constant functional $\mathcal{F}(\mathbf{d})=\mathbf{c}$:

$$\delta\mathcal{F}=\mathbf{0}. \tag{A6}$$

5. For a functional $\mathcal{F}(\cdot)$ and two directions $\mathbf{h}_1$ and $\mathbf{h}_2$:

$$\delta^2\mathcal{F}(\mathbf{d};\mathbf{h}_1,\mathbf{h}_2)=\left.\frac{d}{d\epsilon}\delta\mathcal{F}\left(\mathbf{d}+\epsilon\mathbf{h}_1;\mathbf{h}_2\right)\right|_{\epsilon=0}. \tag{A7}$$

In this paper, we omit the direction of a variation which refers to an arbitrary direction $\delta\mathbf{d}$, i.e.: $\delta\mathcal{F}\left(\mathbf{d}\right)=\delta\mathcal{F}(\mathbf{d};\delta\mathbf{d})$, or $\delta^2\mathcal{F}\left(\mathbf{d}\right)=\delta^2\mathcal{F}(\mathbf{d};\delta\mathbf{d},\delta\mathbf{d})$. We may further omit the independent variable $\mathbf{d}$ if it is not a specified point (such as $\mathbf{d}^0$, $\bar{\mathbf{d}}$ or $\tilde{\mathbf{d}}$). This is consistent with the differentiation of a scalar function $f\left(x\right)$: $df=f'dx$ or $d^2f=f''dx^2$. The terminology *arbitrary* $\delta\mathbf{d}$ is also consistent with the finite element *arbitrary* test function.

Declarations

Competing interests. There is no competing interest.

Funding. There is no funding for this paper.

Acknowledgements. Not applicable.

References

[1] Hou, G., Wang, J., Layton, A.: Numerical methods for fluid-structure interaction–a review. Communications in Computational Physics **12**(2), 337–377 (2012)

[2] Kim, W., Choi, H.: Immersed boundary methods for fluid-structure interaction: A review. International Journal of Heat and Fluid Flow **75**, 301–309 (2019)

[3] Griffith, B.E., Patankar, N.A.: Immersed methods for fluid-structure interaction. Annual Review of Fluid Mechanics **52**, 421–448 (2020)

[4] Takagi, S., Sugiyama, K., Ii, S., Matsumoto, Y.: A review of full Eulerian methods for fluid structure interaction problems. Journal of Applied Mechanics **79**(1), 010911 (2011) https://doi.org/10.1115/1.4005184

[5] Borazjani, I.: A review of fluid-structure interaction simulations of prosthetic heart valves. Journal of Long-term Effects of Medical Implants **25**(1-2) (2015)

[6] Hirschhorn, M., Tchantchaleishvili, V., Stevens, R., Rossano, J., Throckmorton, A.: Fluid-structure interaction modeling in cardiovascular medicine–a systematic review 2017–2019. Medical Engineering & Physics **78**, 1–13 (2020)

[7] Norouzi, H., Younesian, D.: Fluid-structure interactions in nonlinear plates subjected to sub and supersonic airflow: A review. Thin-Walled Structures **192**, 111144 (2023)

[8] Bungartz, H.-J., Schäfer, M.: Fluid-structure Interaction: Modelling, Simulation, Optimisation vol. 53. Springer, Berlin, Heidelberg (2006)

[9] Vierendeels, J., Lanoye, L., Degroote, J., Verdonck, P.: Implicit coupling of partitioned fluid-structure interaction problems with reduced order models. Computers & structures **85**(11-14), 970–976 (2007)

[10] Galdi, G.P., Rannacher, R.: Fundamental Trends in Fluid-structure Interaction vol. 1. World Scientific, Munich (2010)

[11] Rannacher, R., Richter, T.: An adaptive finite element method for fluid-structure interaction problems based on a fully Eulerian formulation. In: Fluid–Structure Interaction II: Modelling, Simulation, Optimization, pp. 159–191. Springer, Berlin, Heidelberg (2010)

[12] Bazilevs, Y., Takizawa, K., Tezduyar, T.E.: Computational Fluid-structure Interaction: Methods and Applications. John Wiley & Sons, New Jersey (2013)

[13] Richter, T.: Fluid-structure Interactions: Models, Analysis and Finite Elements vol. 118. Springer, Cham (2017)

[14] Moubachir, M., Zolesio, J.-P.: Moving Shape Analysis and Control: Applications to Fluid–structure Interactions. CRC Press, New York (2006)

[15] Souli, M., Benson, D.J.: Arbitrary Lagrangian Eulerian and Fluid-structure Interaction: Numerical Simulation. John Wiley & Sons, New Jersey (2013)

[16] Duprat, C., Shore, H.A.: Fluid-structure Interactions in low-Reynolds-number Flows. Royal Society of Chemistry, Silverchair (2016)

[17] Jaiman, R.K., Joshi, V.: Computational Mechanics of Fluid-Structure Interaction. Springer, Singapore (2022)

[18] Wall, W.A., Mok, D.P., Ramm, E.: Partitioned analysis approach of the transient coupled response of viscous fluids and flexible structures. In: Solids, Structures and Coupled Problems in Engineering, Proceedings of the European Conference on Computational Mechanics ECCM, vol. 99 (1999)

[19] Mok, D.P., Wall, W.: Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures. Trends in Computational Structural Mechanics, Barcelona (2001)

[20] Le Tallec, P., Mouro, J.: Fluid structure interaction with large structural displacements. Computer Methods in Applied Mechanics and Engineering **190**(24), 3039–3067 (2001)

[21] Matthies, H.G., Steindorf, J.: Partitioned strong coupling algorithms for fluid–structure interaction. Computers & Structures **81**(8), 805–812 (2003)

[22] Causin, P., Gerbeau, J.-F., Nobile, F.: Added-mass effect in the design of partitioned algorithms for fluid–structure problems. Computer Methods in Applied Mechanics and Engineering **194**(42), 4506–4527 (2005)

[23] Fernández, M.Á., Moubachir, M.: A Newton method using exact Jacobians for solving fluid–structure coupling. Computers & Structures **83**(2), 127–142 (2005)

[24] Förster, C., Wall, W.A., Ramm, E.: Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. Computer Methods in Applied Mechanics and Engineering **196**(7), 1278–1293 (2007)

[25] Küttler, U., Wall, W.A.: Fixed-point fluid–structure interaction solvers with dynamic relaxation. Computational Mechanics **43**(1), 61–72 (2008) https://doi.org/10.1007/s00466-008-0255-5

[26] Degroote, J., Bruggeman, P., Haelterman, R., Vierendeels, J.: Stability of a coupling technique for partitioned solvers in FSI applications. Computers & Structures **86**(23), 2224–2234 (2008)

[27] Küttler, U., Wall, W.A.: Fixed-point fluid–structure interaction solvers with dynamic relaxation. Computational Mechanics **43**(1), 61–72 (2008)

[28] Degroote, J., Bathe, K.-J., Vierendeels, J.: Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction. Computers & Structures **87**(11-12), 793–801 (2009) https://doi.org/10.1016/j.compstruc.2008.11.013

[29] Habchi, C., Russeil, S., Bougeard, D., Harion, J.-L., Lemenand, T., Ghanem, A., Della Valle, D., Peerhossaini, H.: Partitioned solver for strongly coupled fluid–structure interaction. Computers & Fluids **71**, 306–319 (2013)

[30] Degroote, J., Hojjat, M., Stavropoulou, E., Wüchner, R., Bletzinger, K.-U.: Partitioned solution of an unsteady adjoint for strongly coupled fluid-structure interactions and application to parameter identification of a one-dimensional problem. Structural and Multidisciplinary Optimization **47**(1), 77–94 (2013)

[31] Yamada, T., Hong, G., Kataoka, S., Yoshimura, S.: Parallel partitioned coupling analysis system for large-scale incompressible viscous fluid–structure interaction problems. Computers & Fluids **141**, 259–268 (2016)

[32] Haelterman, R., Bogaers, A.E., Scheufele, K., Uekermann, B., Mehl, M.: Improving the performance of the partitioned qn-ils procedure for fluid–structure interaction problems: Filtering. Computers & Structures **171**, 9–17 (2016)

[33] Basting, S., Quaini, A., Čanić, S., Glowinski, R.: Extended ALE method for fluid–structure interaction problems with large structural displacements. Journal of Computational Physics **331**, 312–336 (2017)

[34] Spenke, T., Hosters, N., Behr, M.: A multi-vector interface quasi-Newton method with linear complexity for partitioned fluid–structure interaction. Computer Methods in Applied Mechanics and Engineering **361**, 112810 (2020)

[35] Naseri, A., Totounferoush, A., González, I., Mehl, M., Pérez-Segarra, C.D.: A scalable framework for the partitioned solution of fluid–structure interaction problems. Computational Mechanics **66**, 471–489 (2020)

[36] Bukač, M., Trenchea, C.: Adaptive, second-order, unconditionally stable partitioned method for fluid–structure interaction. Computer Methods in Applied Mechanics and Engineering **393**, 114847 (2022)

[37] Bukač, M., Fu, G., Seboldt, A., Trenchea, C.: Time-adaptive partitioned method for fluid-structure interaction problems with thick structures. Journal of Computational Physics **473**, 111708 (2023)

[38] Heil, M.: An efficient solver for the fully coupled solution of large-displacement fluid–structure interaction problems. Computer Methods in Applied Mechanics and Engineering **193**(1-2), 1–23 (2004) https://doi.org/10.1016/j.cma.2003.09.006

[39] Heil, M., Hazel, A.L., Boyle, J.: Solvers for large-displacement fluid–structure interaction problems: segregated versus monolithic approaches. Computational Mechanics **43**(1), 91–101 (2008) https://doi.org/10.1007/s00466-008-0270-6

[40] Muddle, R.L., Mihajlović, M., Heil, M.: An efficient preconditioner for monolithically-coupled large-displacement fluid–structure interaction problems with pseudo-solid mesh updates. Journal of Computational Physics **231**(21), 7315–7334 (2012) https://doi.org/10.1016/j.jcp.2012.07.001

[41] Hübner, B., Walhorn, E., Dinkler, D.: A monolithic approach to fluid–structure interaction using space–time finite elements. Computer Methods in Applied Mechanics and Engineering **193**(23-26), 2087–2104 (2004)

[42] Hron, J., Turek, S.: A monolithic FEM/multigrid solver for an ALE formulation

of fluid-structure interaction with applications in biomechanics. Fluid-Structure Interaction, 146–170 (2006)

[43] Frei, S., Richter, T.: A locally modified parametric finite element method for interface problems. SIAM Journal on Numerical Analysis **52**(5), 2315–2334 (2014)

[44] Frei, S.: Eulerian finite element methods for interface problems and fluid-structure interactions. PhD thesis, Universität Heidelberg (2016)

[45] Frei, S., Richter, T., Wick, T.: Locmodfe: Locally modified finite elements for approximating interface problems in DEAL.II. Software Impacts **8**, 100070 (2021)

[46] Burman, E., Fernández, M.A., Frei, S., Gerosa, F.M.: A mechanically consistent model for fluid–structure interactions with contact including seepage. Computer Methods in Applied Mechanics and Engineering **392**, 114637 (2022)

[47] Benra, F.-K., Dohmen, H.J., Pei, J., Schuster, S., Wan, B., et al.: A comparison of one-way and two-way coupling methods for numerical analysis of fluid-structure interactions. Journal of Applied Mathematics **2011** (2011)

[48] Hagmeyer, N., Mayr, M., Steinbrecher, I., Popp, A.: One-way coupled fluid–beam interaction: capturing the effect of embedded slender bodies on global fluid flow and vice versa. Advanced Modeling and Simulation in Engineering Sciences **9**(1), 9 (2022)

[49] Zou, B., Thierry, N.N.B., Tang, H., Xu, L., Dong, S., Hu, F.: The deformation characteristics and flow field around knotless polyethylene netting based on fluid structure interaction (FSI) one-way coupling. Aquaculture and Fisheries **7**(1), 89–102 (2022)

[50] Pironneau, O.: Numerical study of a monolithic fluid–structure formulation. In: Variational Analysis and Aerospace Engineering, pp. 401–420. Springer, Switzerland (2016). https://doi.org/10.1007/978-3-319-45680-5_15

[51] Hecht, F., Pironneau, O.: An energy stable monolithic Eulerian fluid-structure finite element method. International Journal for Numerical Methods in Fluids **85**(7), 430–446 (2017) https://doi.org/10.1002/fld.4388

[52] Chiang, C.-Y., Pironneau, O., Sheu, T., Thiriet, M.: Numerical study of a 3D Eulerian monolithic formulation for incompressible fluid-structures systems. Fluids **2**(2), 34 (2017) https://doi.org/10.3390/fluids2020034

[53] Wang, Y., Jimack, P.K., Walkley, M.A., Pironneau, O.: An energy stable one-field monolithic arbitrary Lagrangian-Eulerian formulation for fluid-structure interaction. Journal of Fluids and Structures **98**, 103117 (2020) https://doi.org/

10.1016/j.jfluidstructs.2020.103117

[54] Wang, X., Zhang, L.T.: Modified immersed finite element method for fully-coupled fluid–structure interactions. Computer Methods in Applied Mechanics and Engineering **267**, 150–169 (2013) https://doi.org/10.1016/j.cma.2013.07.019

[55] Massing, A., Larson, M., Logg, A., Rognes, M.: A Nitsche-based cut finite element method for a fluid-structure interaction problem. Communications in Applied Mathematics and Computational Science **10**(2), 97–120 (2015)

[56] Baaijens, F.P.: A fictitious domain/mortar element method for fluid-structure interaction. International Journal for Numerical Methods in Fluids **35**(7), 743–761 (2001) https://doi.org/10.1002/1097-0363(20010415)35:7⟨743::AID-FLD109⟩3.0.CO;2-A

[57] Yu, Z.: A DLM/FD method for fluid/flexible-body interactions. Journal of Computational Physics **207**(1), 1–27 (2005) https://doi.org/10.1016/j.jcp.2004.12.026

[58] Yu, Z.-S., Shao, X.-M.: A three-dimensional fictitious domain method for the simulation of fluid-structure interactions. Journal of Hydrodynamics, Ser. B **22**(5), 178–183 (2010) https://doi.org/10.1016/s1001-6058(09)60190-6

[59] Hesch, C., Gil, A.J., Carreño, A.A., Bonet, J., Betsch, P.: A mortar approach for fluid–structure interaction problems: Immersed strategies for deformable and rigid bodies. Computer Methods in Applied Mechanics and Engineering **278**, 853–882 (2014) https://doi.org/10.1016/j.cma.2014.06.004

[60] Kadapa, C., Dettmer, W.G., Perić, D.: A fictitious domain/distributed Lagrange multiplier based fluid–structure interaction scheme with hierarchical B-Spline grids. Computer Methods in Applied Mechanics and Engineering **301**, 1–27 (2016) https://doi.org/10.1016/j.cma.2015.12.023

[61] Boffi, D., Cavallini, N., Gardini, F., Gastaldi, L.: Local mass conservation of Stokes finite elements. Journal of Scientific Computing **52**(2), 383–400 (2011) https://doi.org/10.1007/s10915-011-9549-4

[62] Boffi, D., Cavallini, N., Gastaldi, L.: The finite element immersed boundary method with distributed Lagrange multiplier. SIAM Journal on Numerical Analysis **53**(6), 2584–2604 (2015) https://doi.org/10.1137/140978399

[63] Boffi, D., Gastaldi, L.: A fictitious domain approach with Lagrange multiplier for fluid-structure interactions. Numerische Mathematik **135**(3), 711–732 (2016) https://doi.org/10.1007/s00211-016-0814-1

[64] Glowinski, R., Pan, T.-W., Hesla, T.I., Joseph, D.D.: A distributed Lagrange

multiplier/fictitious domain method for particulate flows. International Journal of Multiphase Flow **25**(5), 755–794 (1999) https://doi.org/10.1016/s0301-9322(98)00048-2

[65] Glowinski, R., Pan, T.W., Hesla, T.I., Joseph, D.D., Périaux, J.: A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow. Journal of Computational Physics **169**(2), 363–426 (2001) https://doi.org/10.1006/jcph.2000.6542

[66] Glowinski, R.: Finite element methods for incompressible viscous flow. In: Handbook of Numerical Analysis, pp. 3–1176. Elsevier, ??? (2003). https://doi.org/10.1016/s1570-8659(03)09003-3

[67] Patankar, N.A., Singh, P., Joseph, D.D., Glowinski, R., Pan, T.-W.: A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows. International Journal of Multiphase Flow **26**(9), 1509–1524 (2000) https://doi.org/10.1016/s0301-9322(99)00100-7

[68] Hao, J., Pan, T.-W., Glowinski, R., Joseph, D.D.: A fictitious domain/distributed Lagrange multiplier method for the particulate flow of Oldroyd-B fluids: A positive definiteness preserving approach. Journal of Non-Newtonian Fluid Mechanics **156**(1-2), 95–111 (2009) https://doi.org/10.1016/j.jnnfm.2008.07.006

[69] Yiantsios, S.G.: On the distributed Lagrange multiplier/fictitious domain method for rigid-particle-laden flows: a proposition for an alternative formulation of the Lagrange multipliers. International Journal for Numerical Methods in Fluids **70**(8), 1027–1047 (2011) https://doi.org/10.1002/fld.2729

[70] Shi, X., Lim, S.P.: A LBM–DLM/FD method for 3D fluid–structure interactions. Journal of Computational Physics **226**(2), 2028–2043 (2007)

[71] Robinson-Mosher, A., Schroeder, C., Fedkiw, R.: A symmetric positive definite formulation for monolithic fluid structure interaction. Journal of Computational Physics **230**(4), 1547–1566 (2011) https://doi.org/10.1016/j.jcp.2010.11.021

[72] Schott, B., Ager, C., Wall, W.A.: A monolithic approach to fluid-structure interaction based on a hybrid Eulerian-ALE fluid domain decomposition involving cut elements. International Journal for Numerical Methods in Engineering **119**(3), 208–237 (2019)

[73] Ager, C., Seitz, A., Wall, W.A.: A consistent and versatile computational approach for general fluid-structure-contact interaction problems. International Journal for Numerical Methods in Engineering **122**(19), 5279–5312 (2021)

[74] Peskin, C.S.: The immersed boundary method. Acta Numerica **11**, 479–517

(2002) https://doi.org/10.1016/j.cma.2015.12.023

[75] Wang, X., Wang, C., Zhang, L.T.: Semi-implicit formulation of the immersed finite element method. Computational Mechanics **49**(4), 421–430 (2011) https://doi.org/10.1007/s00466-011-0652-z

[76] Wang, X., Zhang, L.T.: Interpolation functions in the immersed boundary and finite element methods. Computational Mechanics **45**(4), 321–334 (2009) https://doi.org/10.1007/s00466-009-0449-5

[77] Zhang, L.T., Gay, M.: Immersed finite element method for fluid-structure interactions. Journal of Fluids and Structures **23**(6), 839–857 (2007) https://doi.org/10.1016/j.jfluidstructs.2007.01.001

[78] Zhang, L., Gerstenberger, A., Wang, X., Liu, W.K.: Immersed finite element method. Computer Methods in Applied Mechanics and Engineering **193**(21), 2051–2067 (2004) https://doi.org/10.1016/j.cma.2003.12.044

[79] Liu, W.K., Kim, D.W., Tang, S.: Mathematical foundations of the immersed finite element method. Computational Mechanics **39**(3), 211–222 (2007)

[80] Wang, Y., Jimack, P.K., Walkley, M.A.: A one-field monolithic fictitious domain method for fluid–structure interactions. Computer Methods in Applied Mechanics and Engineering **317**, 1146–1168 (2017) https://doi.org/10.1016/j.cma.2017.01.023

[81] Wang, Y.: A one-field fictitious domain method for fluid-structure interactions. PhD thesis, University of Leeds (2018)

[82] Wang, Y., Jimack, P.K., Walkley, M.A.: Energy analysis for the one-field fictitious domain method for fluid-structure interactions. Applied Numerical Mathematics **140**, 165–182 (2019) https://doi.org/10.1016/j.apnum.2019.02.003

[83] Wang, Y., Jimack, P.K., Walkley, M.A.: A theoretical and numerical investigation of a family of immersed finite element methods. Journal of Fluids and Structures **91**, 102754 (2019)

[84] Dunne, T., Rannacher, R.: Adaptive finite element approximation of fluid-structure interaction based on an Eulerian variational formulation. In: Lecture Notes in Computational Science and Engineering, pp. 110–145. Springer, Berlin, Heidelberg (2006). https://doi.org/10.1007/3-540-34596-5_6

[85] Dunne, T.: An Eulerian approach to fluid–structure interaction and goal-oriented mesh adaptation. International Journal for Numerical Methods in Fluids **51**(9-10), 1017–1039 (2006) https://doi.org/10.1002/fld.1205

[86] Wall, W.A., Gerstenberger, A., Gamnitzer, P., Forster, C., Ramm, E.: Large deformation fluid-structure interaction-advances in ALE methods and new fixed grid approaches. Lecture Notes in Computational Science and Engineering **53**, 195 (2006)

[87] Richter, T., Wick, T.: Finite elements for fluid–structure interaction in ALE and fully Eulerian coordinates. Computer Methods in Applied Mechanics and Engineering **199**(41-44), 2633–2642 (2010) https://doi.org/10.1016/j.cma.2010.04.016

[88] Rannacher, R., Richter, T.: An adaptive finite element method for fluid-structure interaction problems based on a fully Eulerian formulation. In: Fluid–Structure Interaction II: Modelling, Simulation, Optimization, pp. 159–191. Springer, Berlin, Heidelberg (2010)

[89] Richter, T.: A fully Eulerian formulation for fluid–structure-interaction problems. Journal of Computational Physics **233**, 227–240 (2013) https://doi.org/10.1016/j.jcp.2012.08.047

[90] Wick, T.: Coupling of fully Eulerian and arbitrary Lagrangian–Eulerian methods for fluid-structure interaction computations. Computational Mechanics **52**(5), 1113–1124 (2013) https://doi.org/10.1007/s00466-013-0866-3

[91] Frei, S., Knoke, T., Steinbach, M.C., Wenske, A.-K., Wick, T.: Numerical simulations of fully eulerian fluid-structure contact interaction using a ghost-penalty cut finite element approach. arXiv preprint arXiv:2503.17145 (2025)

[92] Tu, C., Peskin, C.S.: Stability and instability in the computation of flows with moving immersed boundaries: a comparison of three methods. SIAM Journal on Scientific and Statistical Computing **13**(6), 1361–1376 (1992)

[93] Mayo, A.A., Peskin, C.S.: An implicit numerical method for fluid dynamics problems with immersed elastic boundaries. Contemporary Mathematics **141**, 261–261 (1992)

[94] Lai, M.-C., Peskin, C.S.: An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. Journal of computational Physics **160**(2), 705–719 (2000)

[95] Boffi, D., Gastaldi, L.: A finite element approach for the immersed boundary method. Computers & structures **81**(8-11), 491–501 (2003)

[96] Mittal, R., Iaccarino, G.: Immersed boundary methods. Annu. Rev. Fluid Mech. **37**, 239–261 (2005)

[97] Boffi, D., Gastaldi, L., Heltai, L., Peskin, C.S.: On the hyper-elastic formulation of the immersed boundary method. Computer Methods in Applied Mechanics

and Engineering **197**(25-28), 2210–2231 (2008)

[98] Huang, W.-X., Sung, H.J.: An immersed boundary method for fluid–flexible structure interaction. Computer Methods in Applied Mechanics and Engineering **198**(33-36), 2650–2661 (2009)

[99] Roman, F., Napoli, E., Milici, B., Armenio, V.: An improved immersed boundary method for curvilinear grids. Computers & fluids **38**(8), 1510–1527 (2009)

[100] Sotiropoulos, F., Yang, X.: Immersed boundary methods for simulating fluid–structure interaction. Progress in Aerospace Sciences **65**, 1–21 (2014)

[101] De, A.K.: A diffuse interface immersed boundary method for complex moving boundary problems. Journal of Computational Physics **366**, 226–251 (2018)

[102] Huang, W.-X., Tian, F.-B.: Recent trends and progress in the immersed boundary method. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science **233**(23-24), 7617–7636 (2019)

[103] Salih, S.Q., Aldlemy, M.S., Rasani, M.R., Ariffin, A., Ya, T.M.Y.S.T., Al-Ansari, N., Yaseen, Z.M., Chau, K.-W.: Thin and sharp edges bodies-fluid interaction simulation using cut-cell immersed boundary method. Engineering Applications of Computational Fluid Mechanics **13**(1), 860–877 (2019)

[104] Tschisgale, S., Fröhlich, J.: An immersed boundary method for the fluid-structure interaction of slender flexible structures in viscous fluid. Journal of Computational Physics **423**, 109801 (2020)

[105] Li, Z., Lai, M.-C.: The immersed interface method for the Navier–Stokes equations with singular forces. Journal of Computational Physics **171**(2), 822–842 (2001)

[106] Verzicco, R.: Immersed boundary methods: Historical perspective and future outlook. Annual Review of Fluid Mechanics **55**, 129–155 (2023)

[107] Cottet, G.-H., Maitre, E.: A level-set formulation of immersed boundary methods for fluid–structure interaction problems. Comptes Rendus Mathematique **338**(7), 581–586 (2004)

[108] Cottet, G.-H., Maitre, E.: A level set method for fluid-structure interactions with immersed surfaces. Mathematical Models and Methods in Applied Sciences **16**(03), 415–438 (2006)

[109] Legay, A., Chessa, J., Belytschko, T.: An Eulerian–Lagrangian method for fluid–structure interaction based on level sets. Computer Methods in Applied Mechanics and Engineering **195**(17), 2070–2087 (2006)

[110] Cottet, G.-H., Maitre, E., Milcent, T.: Eulerian formulation and level set models

for incompressible fluid-structure interaction. ESAIM: Mathematical Modelling and Numerical Analysis **42**(3), 471–492 (2008)

[111] He, P., Qiao, R.: A full-Eulerian solid level set method for simulation of fluid–structure interactions. Microfluidics and Nanofluidics **11**, 557–567 (2011)

[112] Sun, P., Xu, J., Zhang, L.: Full Eulerian finite element method of a phase field model for fluid–structure interaction problem. Computers & Fluids **90**, 1–8 (2014)

[113] Mokbel, D., Abels, H., Aland, S.: A phase-field model for fluid–structure interaction. Journal of Computational Physics **372**, 823–840 (2018)

[114] Valizadeh, N., Zhuang, X., Rabczuk, T.: A monolithic finite element method for phase-field modeling of fully eulerian fluid–structure interaction. Computer Methods in Applied Mechanics and Engineering **435**, 117618 (2025)

[115] Burman, E., Hansbo, P.: Edge stabilization for the generalized stokes problem: a continuous interior penalty method. Computer Methods in Applied Mechanics and Engineering **195**(19-22), 2393–2410 (2006)

[116] Burman, E.: Ghost penalty. Comptes Rendus. Mathématique **348**(21-22), 1217–1220 (2010)

[117] Wang, L., Xie, C., Huang, W.: A monolithic projection framework for constrained fsi problems with the immersed boundary method. Computer Methods in Applied Mechanics and Engineering **371**, 113332 (2020)

[118] Wang, X.S.: An iterative matrix-free method in implicit immersed boundary/continuum methods. Computers & structures **85**(11-14), 739–748 (2007)

[119] Li, W., Wang, W.-Q., Yan, Y., Yu, Z.-F.: A strong-coupled method combined finite element method and Lattice Boltzmann method via an implicit immersed boundary scheme for fluid structure interaction. Ocean Engineering **214**, 107779 (2020)

[120] Boffi, D., Cavallini, N., Gastaldi, L.: The finite element immersed boundary method with distributed Lagrange multiplier. SIAM Journal on Numerical Analysis **53**(6), 2584–2604 (2015)

[121] Boffi, D., Gastaldi, L., Heltai, L.: A distributed Lagrange formulation of the finite element immersed boundary method for fluids interacting with compressible solids. Mathematical and Numerical Modeling of The Cardiovascular System and Applications, 1–21 (2018)

[122] LeVeque, R.J., Li, Z.: Immersed interface methods for stokes flow with elastic boundaries or surface tension. SIAM Journal on Scientific Computing **18**(3),

709–735 (1997)

[123] Li, Z.: The immersed interface method using a finite element formulation. Applied Numerical Mathematics **27**(3), 253–267 (1998)

[124] Li, Z.: Immersed interface methods for moving interface problems. Numerical Algorithms **14**(4), 269–293 (1997)

[125] Wiegmann, A., Bube, K.P.: The explicit-jump immersed interface method: finite difference methods for pdes with piecewise smooth solutions. SIAM Journal on Numerical Analysis **37**(3), 827–862 (2000)

[126] Li, Z.: An overview of the immersed interface method and its applications. Taiwanese Journal of Mathematics **7**(1), 1–49 (2003)

[127] Le, D.-V., Khoo, B.C., Peraire, J.: An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries. Journal of Computational Physics **220**(1), 109–138 (2006)

[128] Xu, S., Wang, Z.J.: An immersed interface method for simulating the interaction of a fluid with moving boundaries. Journal of Computational Physics **216**(2), 454–493 (2006)

[129] Dillon, R.H., Li, Z.: An introduction to the immersed boundary and the immersed interface methods. Interface Problems and Methods in Biological and Physical Flows, 1–67 (2009)

[130] Kolahdouz, E.M., Bhalla, A.P.S., Scotten, L.N., Craven, B.A., Griffith, B.E.: A sharp interface Lagrangian-Eulerian method for rigid-body fluid-structure interaction. Journal of Computational Physics **443**, 110442 (2021)

[131] Kees, C.E., Collins, J.H., Zhang, A.: Simple, accurate, and efficient embedded finite element methods for fluid–solid interaction. Computer Methods in Applied Mechanics and Engineering **389**, 114404 (2022)

[132] Casquero, H., Bona-Casas, C., Gomez, H.: A NURBS-based immersed methodology for fluid–structure interaction. Computer Methods in Applied Mechanics and Engineering **284**, 943–970 (2015)

[133] Nitti, A., Kiendl, J., Reali, A., Tullio, M.D.: An immersed-boundary/isogeometric method for fluid–structure interaction involving thin shells. Computer Methods in Applied Mechanics and Engineering **364**, 112977 (2020)

[134] Agrawal, V., Kulachenko, A., Scapin, N., Tammisola, O., Brandt, L.: An efficient isogeometric/finite-difference immersed boundary method for the fluid–structure interactions of slender flexible structures. Computer Methods in

Applied Mechanics and Engineering **418**, 116495 (2024)

[135] Feng, Z.-G., Michaelides, E.E.: The immersed boundary-Lattice Boltzmann method for solving fluid–particles interaction problems. Journal of Computational Physics **195**(2), 602–628 (2004)

[136] Strack, O.E., Cook, B.K.: Three-dimensional immersed boundary conditions for moving solids in the lattice-boltzmann method. International Journal for Numerical Methods in Fluids **55**(2), 103–125 (2007)

[137] Wu, J., Shu, C.: Implicit velocity correction-based immersed boundary-Lattice Boltzmann method and its applications. Journal of Computational Physics **228**(6), 1963–1979 (2009)

[138] Wu, J., Shu, C.: Particulate flow simulation via a boundary condition-enforced immersed boundary-Lattice Boltzmann scheme. Communications in Computational Physics **7**(4), 793 (2010)

[139] Kefayati, G.R., Tang, H., Chan, A.: Immersed boundary-finite difference Lattice Boltzmann method through fluid–structure interaction for viscoplastic fluids. Journal of Fluids and Structures **83**, 238–258 (2018)

[140] Cai, Y., Wang, S., Lu, J., Li, S., Zhang, G.: Efficient immersed-boundary Lattice Boltzmann scheme for fluid-structure interaction problems involving large solid deformation. Physical Review E **99**(2), 023310 (2019)

[141] Li, Z., Cao, W., Le Touzé, D.: On the coupling of a direct-forcing immersed boundary method and the regularized Lattice Boltzmann method for fluid-structure interaction. Computers & Fluids **190**, 470–484 (2019)

[142] Ma, J., Wang, Z., Young, J., Lai, J.C., Sui, Y., Tian, F.-B.: An immersed boundary-Lattice Boltzmann method for fluid-structure interaction problems involving viscoelastic fluids and complex geometries. Journal of Computational Physics **415**, 109487 (2020)

[143] Yang, F., Gu, X., Xia, X., Zhang, Q.: A peridynamics-immersed boundary-Lattice Boltzmann method for fluid-structure interaction analysis. Ocean Engineering **264**, 112528 (2022)

[144] Cheng, J., Yu, F., Zhang, L.T.: OpenIFEM: a high performance modular open-source software of the immersed finite element method for fluid-structure interactions. Computer Modeling in Engineering & Sciences: CMES **119**(1), 91 (2019)

[145] Jiang, C., Zhang, Z.-Q., Gao, G.-J., Liu, G.: A modified immersed smoothed FEM with local field reconstruction for fluid–structure interactions. Engineering Analysis with Boundary Elements **107**, 218–232 (2019)

[146] Prenter, F., Verhoosel, C.V., Brummelen, E.H., Larson, M.G., Badia, S.: Stability and conditioning of immersed finite element methods: analysis and remedies. Archives of Computational Methods in Engineering, 1–40 (2023)

[147] Fromm, J.E., Wunsch, N., Xiang, R., Zhao, H., Maute, K., Evans, J.A., Kamensky, D.: Interpolation-based immersed finite element and isogeometric analysis. Computer Methods in Applied Mechanics and Engineering **405**, 115890 (2023)

[148] Caginalp, G.: An analysis of a phase field model of a free boundary. Archive for Rational Mechanics and Analysis **92**, 205–245 (1986)

[149] Feireisl, E., Petzeltová, H., Rocca, E., Schimperna, G.: Analysis of a phase-field model for two-phase compressible fluids. Mathematical Models and Methods in Applied Sciences **20**(07), 1129–1160 (2010)

[150] Bazilevs, Y., Hsu, M.-C., Zhang, Y., Wang, W., Kvamsdal, T., Hentschel, S., Isaksen, J.G.: Computational vascular fluid–structure interaction: methodology and application to cerebral aneurysms. Biomechanics and Modeling in Mechanobiology **9**(4), 481–498 (2010) https://doi.org/10.1007/s10237-010-0189-7

[151] Pironneau, O.: An energy preserving monolithic Eulerian fluid-structure numerical scheme. arXiv:1607.08083v1 [cs.CE] 27 Jul 2016 (2016)

[152] Belytschko, T., Liu, W.K., Moran, B., Elkhodary, K.: Nonlinear Finite Elements for Continua and Structures. John Wiley & Sons, New Jersey (2014)

[153] Magnus, J.R., Neudecker, H.: Matrix differential calculus with applications in statistics and econometrics. Wiley Series in Probability and Mathematical Statistics (1988)

[154] Rall, L.B.: Nonlinear Functional Analysis and Applications: Proceedings of an Advanced Seminar Conducted by the Mathematics Research Center, the University of Wisconsin, Madison, October 12–14, 1970. Mathematics Research Center Lecture Notes, vol. 26. Elsevier, Amsterdam (2014)

[155] Liang, G., Zhou, Y.: Finite Element Language and Its Application (in Chinese). Science Press, Beijing (2013)

[156] Boukir, K., Maday, Y., Métivet, B., Razafindrakoto, E.: A high-order characteristics/finite element method for the incompressible Navier-Stokes equations. International Journal for Numerical Methods in Fluids **25**(12), 1421–1454 (1997)

[157] Chiang, C.-Y., Pironneau, O., Sheu, T., Thiriet, M.: Numerical study of a 3D Eulerian monolithic formulation for incompressible fluid-structures systems. Fluids **2**(2), 34 (2017)

[158] Schroeder, C., Chowdhury, R.R., Shinar, T.: Local divergence-free polynomial interpolation on MAC grids. Journal of Computational Physics **468**, 111500 (2022)

[159] Lee, T.-R., Chang, Y.-S., Choi, J.-B., Kim, D.W., Liu, W.K., Kim, Y.-J.: Immersed finite element method for rigid body motions in the incompressible Navier–Stokes flow. Computer Methods in Applied Mechanics and Engineering **197**(25-28), 2305–2316 (2008) https://doi.org/10.1016/j.cma.2007.12.013

[160] Zhang, L.T., Gay, M.: Imposing rigidity constraints on immersed objects in unsteady fluid flows. Computational Mechanics **42**(3), 357–370 (2008) https://doi.org/10.1007/s00466-008-0244-8

[161] Liu, G.-R., Gu, Y.-T.: An Introduction to Meshfree Methods and Their Programming. Springer, Dordrecht (2005)

[162] Ambati, M., Gerasimov, T., De Lorenzis, L.: A review on phase-field models of brittle fracture and a new fast hybrid formulation. Computational Mechanics **55**(2), 383–405 (2015)

[163] Boyer, F., Lapuerta, C., Minjeaud, S., Piar, B., Quintard, M.: Cahn–Hilliard/Navier–Stokes model for the simulation of three-phase flows. Transport in Porous Media **82**(3), 463–483 (2010)

[164] Bollada, P., Jimack, P., Mullis, A.: Bracket formalism applied to phase field models of alloy solidification. Computational Materials Science **126**, 426–437 (2017)

[165] Fries, T.-P., Byfut, A., Alizada, A., Cheng, K.W., Schöder, A.: Hanging nodes and XFEM. International Journal for Numerical Methods in Engineering **86**(4-5), 404–430 (2010) https://doi.org/10.1002/nme.3024

[166] Cheng, K.-W., Fries, T.-P.: XFEM with hanging nodes for two-phase incompressible flow. Computer Methods in Applied Mechanics and Engineering **245**, 290–312 (2012)

[167] Sauerland, H., Fries, T.-P.: The stable XFEM for two-phase flows. Computers & Fluids **87**, 41–49 (2013)

[168] Daniele Boffi, L.H. Lucia Gastaldi: A distributed Lagrange formulation of the finite element immersed boundary method for fluids interacting with compressible solids (2017) arXiv:1712.02545 [math]

[169] Burman, E., Fernández, M.A.: An unfitted Nitsche method for incompressible fluid–structure interaction using overlapping meshes. Computer Methods in Applied Mechanics and Engineering **279**, 497–514 (2014)

[170] Burman, E., Claus, S., Hansbo, P., Larson, M.G., Massing, A.: CutFEM: discretizing geometry and partial differential equations. International Journal for Numerical Methods in Engineering **104**(7), 472–501 (2015)

[171] Hecht, F.: New development in FreeFem++. Journal of Numerical Mathematics **20**(3-4), 251–265 (2012)

[172] Turek, S., Hron, J.: Proposal for numerical benchmarking of fluid–structure interaction between an elastic object and laminar incompressible flow, 371–385 (2006)

[173] Hesch, C., Gil, A., Carreno, A.A., Bonet, J., Betsch, P.: A mortar approach for fluid–structure interaction problems: Immersed strategies for deformable and rigid bodies. Computer Methods in Applied Mechanics and Engineering **278**, 853–882 (2014)

[174] Singh, R.K., Kant, R., Singh, S., Suresh, E., Gupta, A., Bhattacharya, S.: A novel helical micro-valve for embedded micro-fluidic applications. Microfluidics and Nanofluidics **19**(1), 19–29 (2015)

[175] Martell, L.P.: Computational fluid dynamics techniques for fixed-bed biofilm systems modeling: numerical simulations and experimental characterization. PhD thesis, Polytechnic University of Catalonia, Spain (2018)

[176] Agostini, E.: Computational fluid dynamics simulations of flows in stirred tanks: application to bio-reactors. PhD thesis, Politecnico di Torino (2019)

[177] Senevirathne, S., Hasan, J., Mathew, A., Woodruff, M., Yarlagadda, P.: Simulation of bacterial motion under flow inside micro channel using CFD and DPM. In: Global Congress on Manufacturing and Management, pp. 95–108 (2021). Springer

[178] Ferreira, M., Carvalho, V., Ribeiro, J., Lima, R.A., Teixeira, S., Pinho, D.: Advances in microfluidic systems and numerical modeling in biomedical applications: A review. Micromachines **15**(7), 873 (2024)

[179] Hron, J., Mádlík, M.: Fluid-structure interaction with applications in biomechanics. Nonlinear Analysis: Real World Applications **8**(5), 1431–1458 (2007)

[180] Sturla, F., Votta, E., Stevanella, M., Conti, C.A., Redaelli, A.: Impact of modeling fluid–structure interaction in the computational analysis of aortic root biomechanics. Medical engineering & physics **35**(12), 1721–1730 (2013)

[181] Fadhil, N.A., Hammoodi, K.A., Jassim, L., Al-Asadi, H.A., Habeeb, L.J.: Multiphysics analysis for fluid–structure interaction of blood biological flow inside three-dimensional artery. Curved and Layered Structures **10**(1), 20220187 (2023)

[182] Fandaros, M., Li, Y.Y., Cao, J.J., Yin, W.: A spatiotemporal analysis of the left coronary artery biomechanics using fluid–structure interaction models. Medical & Biological Engineering & Computing **61**(6), 1533–1548 (2023)

[183] Lee, K., Huque, Z., Kommalapati, R., Han, S.-E.: The evaluation of aerodynamic interaction of wind blade using fluid structure interaction method. Journal of Clean Energy Technologies **3**(4), 270–275 (2015)

[184] Wang, L., Quant, R., Kolios, A.: Fluid structure interaction modelling of horizontal-axis wind turbine blades based on CFD and FEA. Journal of Wind Engineering and Industrial Aerodynamics **158**, 11–25 (2016)

[185] Hami, A.E., Radi, B.: Fluid–Structure Interactions and Uncertainties: ANSYS and Fluent Tools. John Wiley & Sons, Hoboken, NJ (2017)

[186] Liu, X., Gui, N., Wu, H., Yang, X., Tu, J., Jiang, S.: Numerical simulation of flow past stationary and oscillating deformable circles with fluid-structure interaction. Experimental and Computational Multiphase Flow **2**(3), 151–161 (2020)

[187] Zhang, D., Liu, Z., Li, W., Zhang, J., Cheng, L., Hu, G.: Fluid-structure interaction analysis of wind turbine aerodynamic loads and aeroelastic responses considering blade and tower flexibility. Engineering Structures **301**, 117289 (2024)

[188] Malikov, Z., Madaliev, M., Chernyshev, S., Ionov, A.: Validation of a two-fluid turbulence model in COMSOL multiphysics for the problem of flow around aerodynamic profiles. Scientific Reports **14**(1), 2306 (2024)

[189] Dutta, S., Jog, C.: Monolithic finite element modeling of compressible fluid-structure-electrostatics interactions in MEMS devices. International Journal for Numerical Methods in Fluids **96**(12), 2006–2050 (2024)

[190] Xu, X., Cheng, J.: Fluid-structure interaction simulation of the three-leaflet aortic valve using COMSOL. Computer Methods in Biomechanics and Biomedical Engineering, 1–13 (2024)

[191] Liu, W.K., Liu, Y., Farrell, D., Zhang, L., Wang, X.S., Fukui, Y., Patankar, N., Zhang, Y., Bajaj, C., Lee, J., *et al.*: Immersed finite element method and its applications to biological systems. Computer Methods in Applied Mechanics and Engineering **195**(13), 1722–1749 (2006)

[192] Brown, J.A., Lee, J.H., Smith, M.A., Wells, D.R., Barrett, A., Puelz, C., Vavalle, J.P., Griffith, B.E.: Patient–specific immersed finite element–difference model of transcatheter aortic valve replacement. Annals of Biomedical Engineering **51**(1), 103–116 (2023)

[193] Kafafy, R., Lin, T., Lin, Y., Wang, J.: Three-dimensional immersed finite element methods for electric field simulation in composite materials. International Journal for Numerical Methods in Engineering **64**(7), 940–972 (2005)

[194] Groeneveld, A.B., Chen, P.: An immersed finite element method for interfacial damage and debonding. Computational Mechanics **75**(5), 1555–1583 (2025)

[195] Han, D., He, X., Lund, D., Zhang, X.: Pife-pic: Parallel immersed finite element particle-in-cell for 3-d kinetic simulations of plasma-material interactions. SIAM Journal on Scientific Computing **43**(3), 235–257 (2021)

[196] Han, Y., Xia, G., Kang, H., Lu, C., Chen, C., Saetchnikov, V.: Three-dimensional direct-implicit particle-in-cell model using trilinear anisotropic immersed-finite-element for plasma propulsion. AIAA Journal **63**(1), 323–335 (2025)

[197] Wu, C., Young, D.-L., Chiu, C.-L.: Simulation of wave–structure interaction by hybrid Cartesian/immersed boundary and arbitrary Lagrangian–Eulerian finite-element method. Journal of Computational Physics **254**, 155–183 (2013)

[198] Bürchner, T., Radtke, L., Kopp, P., Kollmannsberger, S., Rank, E., Düster, A.: On the efficiency of explicit and semi-explicit immersed boundary finite element methods for wave propagation problems. arXiv preprint arXiv:2501.19013 (2025)

[199] Iaccarino, G., Verzicco, R.: Immersed boundary technique for turbulent flow simulations. Applied Mechanics Reviews **56**(3), 331–347 (2003)

[200] Capizzano, F.: Turbulent wall model for immersed boundary methods. AIAA Journal **49**(11), 2367–2381 (2011)

[201] Yan, K., Wu, Y., Zhu, Q., Cui, Y., Khoo, B.C.: Efficient wall-modeling diffused-interface immersed boundary method for solving turbulent flows with high-order finite difference schemes. Physics of Fluids **36**(11) (2024)

[202] Agyekum, G.O.: Homogenization method for topology optimization of a coupled thermal fluid-structures built with composite materials. PhD thesis, Université Paris Cité (2022)

[203] Katamine, E., Kawai, R., Yoshida, Y.: Shape optimisation of fluid-structure interactive field considering geometrically nonlinear structure. International Journal of Computational Fluid Dynamics **36**(2), 138–151 (2022)

[204] NARUSE, S., KATAMINE, E.: Shape design for controlling structural displacement distribution in unsteady fluid-structure-interaction. Mechanical Engineering Journal **12**(2), 24–00423 (2025)

[205] Heil, M., Hazel, A.L., Boyle, J.: Solvers for large-displacement fluid–structure

interaction problems: segregated versus monolithic approaches. Computational Mechanics **43**(1), 91–101 (2008)

[206] Heil, M., Hazel, A.L.: Fluid-structure interaction in internal physiological flows. Annual Review of Fluid Mechanics **43**, 141–162 (2011)

[207] Wick, T.: Solving monolithic fluid-structure interaction problems in arbitrary Lagrangian Eulerian coordinates with the DEAL.II library. Archive of Numerical Software **1**(1), 1–19 (2013)

[208] Myrbäck, S., Zahedi, S.: A high-order conservative cut finite element method for problems in time-dependent domains. Computer Methods in Applied Mechanics and Engineering **431**, 117245 (2024)

[209] Burman, E., Elfverson, D., Hansbo, P., Larson, M.G., Larsson, K.: Shape optimization using the cut finite element method. Computer Methods in Applied Mechanics and Engineering **328**, 242–261 (2018)

[210] Berggren, M., Bernland, A., Massing, A., Noreland, D., Wadbro, E.: A better compression driver? CutFEM 3D shape optimization taking viscothermal losses into account. arXiv preprint arXiv:2403.17963 (2024)

[211] Kerfriden, P., Claus, S., Mihai, I.: A mixed-dimensional CutFEM methodology for the simulation of fibre-reinforced composites. Advanced Modeling and Simulation in Engineering Sciences **7**(1), 18 (2020)

[212] Claus, S., Bigot, S., Kerfriden, P.: CutFEM method for Stefan–Signorini problems with application in pulsed laser ablation. SIAM Journal on Scientific Computing **40**(5), 1444–1469 (2018)