Received: 15 May 2023 | Revised: 27 July 2023 | Accepted: 14 August 2023

DOI: 10.1002/nme.7350

RESEARCH ARTICLE

WILEY

Exploiting high-contrast Stokes preconditioners to efficiently solve incompressible fluid–structure interaction problems

Michał Wichrowski[1] | Piotr Krzyżanowski[2] | Luca Heltai[3] | Stanisław Stupkiewicz[4]

[1]Interdisciplinary Center for Scientific Computing, Heidelberg University, Heidelberg, Germany

[2]Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Warsaw, Poland

[3]Mathematics Area, SISSA – International School for Advanced Studies, Trieste, Italy

[4]Institute of Fundamental Technological Research, Polish Academy of Sciences, Warsaw, Poland

Correspondence
Michał Wichrowski, Interdisciplinary Center for Scientific Computing, Heidelberg University, Im Neuenheimer Feld 205, 69120, Heidelberg, Germany.
Email: mt.wichrowsk@uw.edu.pl

Funding information
Narodowe Centrum Nauki, Grant/Award Number: 2015/19/N/ST8/03924; EffectFact project, Grant/Award Number: 101008140; H2020 Programme, MSC Action, Grant/Award Number: RISE-2022

Abstract
In this work, we develop a new algorithm to solve large-scale incompressible time-dependent fluid–structure interaction problems using a matrix-free finite element method in arbitrary Lagrangian–Eulerian frame of reference. We derive a semi-implicit time integration scheme which improves the geometry-convective explicit scheme for problems involving the interaction between incompressible hyperelastic solids and incompressible fluids. The proposed algorithm relies on the reformulation of the time-discrete problem as a generalized Stokes problem with strongly variable coefficients, for which optimal preconditioners have recently been developed. The resulting algorithm is scalable, optimal, and robust: we test our implementation on model problems that mimic classical Turek–Hron benchmarks in two and three dimensions, and investigate timing and scalability results.

KEYWORDS
arbitrary Lagrangian–Eulerian, finite element method, fluid–structure interaction, geometric multigrid, matrix-free method, monolithic scheme

1 | INTRODUCTION

The study of fluid–structure interaction (FSI) is crucial in numerous fields of science and engineering. One of the earliest references to FSI is attributed to Taylor,[1] who described the interaction between a flexible swimming animal and the surrounding fluid. Since then, FSI has remained an active area of research, with significant advancements in computational methods,[2-5] time stepping techniques,[6] coupling strategies,[7-9] and preconditioning techniques[10,11] (see, for example, the monographs by Bazilevs et al.[12] or Richter[13] for an overview on the topic).

The governing equations for FSI problems can be formulated in different reference frames. In this article, we focus on the arbitrary Lagrangian–Eulerian (ALE) formulation,[14] although fully Lagrangian[15] and fully Eulerian[16] formulations are also used in the context of FSI problems. The ALE formulation is the preferred choice when no topological changes occur during the simulations and for relatively small domain changes. If the problem requires such features, then one typically resorts to non-matching methods, for instance, fictitious domain methods,[17] distributed Lagrange multiplier methods,[18] or immersed methods.[19,20]

The combination of two (generally nonlinear) continuum mechanics models brings with it the inherent complexity of both of them, which is further compounded by the challenges that arise due to their coupling. It is not surprising that the solution of a fully-coupled monolithic FSI problem[2,21-23] is typically avoided by employing simplified coupling models, such as partitioned loosely coupled schemes (explicit schemes)[24] or partitioned fully coupled schemes (fixed-point or implicit schemes)[4,9,25,26] (see, for example, References 7,27, and 8 for a review of the various methods).

In the case of FSI problems involving incompressible fluids, fully partitioned schemes are often unstable due to problems with added-mass effect.[25,28,29] The natural remedy of leaning towards classical fixed-point methods is often considered too expensive,[30] however, some successful approaches have been developed.[31]

Implicit schemes lead to a system of nonlinear equations to be solved at each time step. The most complex computational scheme—and most stable—is the fully implicit scheme,[2] where the whole problem is treated implicitly, and all Jacobians are computed exactly, resulting in a system of equations that couples in a nonlinear way the fluid motion, the deformation of the solid, and the evolution of the domain geometry. In References 32 and 11 block preconditioners for such a system were proposed.

Although this is indeed a very robust method, solving such a nonlinear problem is not necessarily needed, since stable integration schemes can be obtained by first using explicit methods to predict the deformed domain, and then using an implicit scheme to compute the flow and deformation. This is the idea behind the geometry-convective explicit (GCE) scheme.[6,33-35] Well-posedness of the linear system arising from that discretization has been proven in Reference 36, and some block preconditioners have been developed.

In this article, we design and evaluate an efficient and scalable, fully-coupled, semi-implicit FSI solver based on a stabilized finite element method, which uses parallel matrix-free computing to deal with large problem sizes and exploits preconditioners for high-contrast Stokes problems. In the article, we summarize the main results of the PhD thesis[37] and report new results on a 3D FSI problem. Our FSI solver has much better stability properties than the GCE scheme, while retaining a relatively low cost as compared to fully implicit methods. It uses a velocity-based formulation that allows for a natural coupling between the fluid and solid equations. We model the fluid as an incompressible Newtonian fluid and the solid as an incompressible Mooney–Rivlin hyperelastic material[38] and complement the solid model with a volume stabilization technique that improves incompressibility of the solid. Incompressibility is a highly relevant assumption for a wide range of rubber-like materials and in many biological applications. On the one hand, solid incompressibility makes the computational treatment of the problem more difficult compared to the case of compressible and nearly-incompressible solids, leading to an additional computational burden. On the other hand, treating the solid as incompressible opens the way to the reinterpretation of the FSI problem as a high-contrast Stokes problem.

One of the key ingredients of the solver presented in this article is a semi-implicit scheme that improves on the GCE scheme by applying similar ideas to the nonlinear terms in the fluid and solid equations, treating them semi-implicitly with a predictor–corrector algorithm, crafted via a careful modification of backward differentiation formulas (BDF). Such rewriting allows us to reinterpret the major step of the resulting improved GCE scheme as a high-contrast Stokes problem, where the overall viscosity presents jumps with a ratio of six to nine orders of magnitude. The resolution of the solid problem in terms of solid velocities is classical, but it is usually discarded from the computational point of view, due to the lack of efficient solvers for such high contrast coefficients. In this work we revamp this formulation and exploit efficient preconditioners that have recently been developed, for example, References 37,39-41. While direct application of a high-contrast Stokes solver (as in References 37 and 40) is natural when following the assumption of incompressibility of the solid, such an assumption can be relaxed with minor modifications on the overall scheme, provided that the compressibility of the solid is treated semi-implicitly.

A major advantage of the proposed FSI solver is its ability to make a consistent use of matrix-free computing, a technique where matrix operations are performed directly on the data, without the need to store the matrix explicitly, thus increasing CPU cache efficiency and lowering the overall memory footprint.[42] This is only possible because—thanks to the crucial connection with the generalized Stokes problem provided by our time-stepping method (see Section 3.1.1)—we are able to adapt to our problem a specialized preconditioner,[40] matrix-free by design, and suitable for high-contrast problems.

The article is organized as follows: in Section 2, we describe the mathematical formulation of the FSI problem and in Section 3 we introduce a semi-implicit time integration scheme and the spatial discretization using the finite-element method. Section 4 discusses the preconditioned iterative solvers employed to efficiently solve the resulting linear system of equations. In Section 5, we present the numerical experiments carried out to validate our method, including a test resembling Turek–Hron[43] benchmarks. We investigate the stability and efficiency of our solver for a range of parameters, including mesh size and time step. Finally, in Section 6, we summarize the main findings of our work and discuss some perspectives for future research.

2 | FLUID–STRUCTURE INTERACTION MODEL

2.1 | Weak form of the conservation equations

Let us consider an initial (reference) configuration given by the domain $\hat{\Omega} \subset \mathbb{R}^d$, consisting of two non-overlapping sub-domains: a fluid domain $\hat{\Omega}_f \subset \hat{\Omega}$, and a solid domain $\hat{\Omega}_s \subset \hat{\Omega}$ ($\hat{\Omega} = \hat{\Omega}_s \cup \hat{\Omega}_f$). In FSI problems, one expects changes in both the solid configuration and the fluid domain in time. We denote the actual domain at time $t \in [0, T]$ by $\Omega(t)$, with the convention that $\Omega(0) = \hat{\Omega}$. The deformed solid then occupies the domain $\Omega_s(t)$, while the fluid domain $\Omega_f(t)$ occupies the region $\Omega(t) \setminus \Omega_s(t)$. When it will be clear from the context which instant t is referred to, we will usually drop the time argument of the domains.

We use plain fonts to indicate scalar Eulerian variables (e.g., p) and indicate vector and tensor Eulerian variables using boldface characters (e.g., $\mathbf{v}$ or $\boldsymbol{\sigma}$).

We indicate global fields for both the solid and the fluid without subscripts, and indicate with subscript s fields referring to the solid domain Ω_s, and with subscript f fields referring to the fluid domain Ω_f, that is,

$$\mathbf{v}(t,\mathbf{x}) = \begin{cases} \mathbf{v}_s(t,\mathbf{x}) & \mathbf{x} \in \Omega_s \\ \mathbf{v}_f(t,\mathbf{x}) & \mathbf{x} \in \Omega_f. \end{cases} \tag{1}$$

The field $\mathbf{v}$ represents the velocity of whatever type of particle happens to be at point $\mathbf{x}$ at time t. Similarly, we define a global Cauchy stress tensor $\boldsymbol{\sigma}$, and a global density field for the coupled problem, given by

$$\boldsymbol{\sigma}(t,\mathbf{x}) = \begin{cases} \boldsymbol{\sigma}_s(t,\mathbf{x}) & \mathbf{x} \in \Omega_s \\ \boldsymbol{\sigma}_f(t,\mathbf{x}) & \mathbf{x} \in \Omega_f, \end{cases} \qquad \rho(t,\mathbf{x}) = \begin{cases} \rho_s(t,\mathbf{x}) & \mathbf{x} \in \Omega_s \\ \rho_f(t,\mathbf{x}) & \mathbf{x} \in \Omega_f, \end{cases} \tag{2}$$

which will depend on the constitutive properties of the fluid and of the solid. For brevity, we will skip the time and space dependence of the variables when confusion is not possible.

By imposing conservation of mass and of momentum,[13] one obtains the following formal system of equations:

$$\begin{cases} \rho \frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} - \operatorname{div} \boldsymbol{\sigma} = \mathbf{g}, \\ \frac{\mathrm{D}\rho}{\partial t} + \rho \operatorname{div} \mathbf{v} = 0 \end{cases} \qquad \text{in } \Omega(t) \setminus \Gamma_i(t), \tag{3}$$

where $\mathbf{g}(t)$ is a given external force field (per unit volume), $\boldsymbol{\sigma}$ is the Cauchy stress tensor defined above, and $\Gamma_i(t)$ is the fluid–solid interface $\Gamma_i(t) = \partial\Omega_s(t) \cap \partial\Omega_f(t)$. Conservation of angular momentum is guaranteed if the Cauchy stress tensor is symmetric.

We indicate with "grad" and "div" the spatial differential operators with respect to the coordinates $\mathbf{x}$ in the deformed configuration, and indicate with $\frac{\mathrm{D}\boldsymbol{\kappa}}{\mathrm{D}t}$ the material derivative, that is, for a vector field $\boldsymbol{\kappa}$ and a scalar field α, these are defined as time derivatives along the flow:

$$\frac{\mathrm{D}\boldsymbol{\kappa}}{\mathrm{D}t} = \frac{\partial}{\partial t}\boldsymbol{\kappa} + (\operatorname{grad} \boldsymbol{\kappa})\mathbf{v}, \qquad \frac{\mathrm{D}\alpha}{\mathrm{D}t} = \frac{\partial}{\partial t}\alpha + \mathbf{v} \cdot \operatorname{grad} \alpha. \tag{4}$$

We partition the boundary $\Gamma(t) = \partial\Omega(t)$ into Neumann $\Gamma_N(t)$ and Dirichlet $\Gamma_D(t)$ parts, and we set the following transmission and boundary conditions

$$\begin{cases} \boldsymbol{\sigma}_f \mathbf{n}_f + \boldsymbol{\sigma}_s \mathbf{n}_s = \mathbf{0} & \text{on } \Gamma_i \\ \mathbf{v}_f = \mathbf{v}_s & \text{on } \Gamma_i \\ \mathbf{v} = \mathbf{v}_D^* & \text{on } \Gamma_D \\ \boldsymbol{\sigma}\mathbf{n} = \boldsymbol{\tau}^* & \text{on } \Gamma_N, \end{cases} \tag{5}$$

where $\mathbf{v}_D^*(t)$ is a given prescribed velocity, $\boldsymbol{\tau}^*(t)$ is a given prescribed traction, and $\mathbf{n}$, $\mathbf{n}_s$, and $\mathbf{n}_f$ denote the unit outer normals to $\partial\Omega(t)$, $\partial\Omega_s(t)$, and $\partial\Omega_f(t)$, respectively.

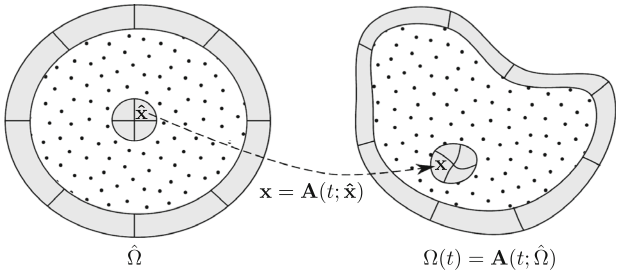
$\hat{\mathbf{x}}$
$\mathbf{x}$
$\mathbf{x} = \mathbf{A}(t; \hat{\mathbf{x}})$
$\hat{\Omega}$
$\Omega(t) = \mathbf{A}(t; \hat{\Omega})$

FIGURE 1 Initial domain $\hat{\Omega}$ and deformed domain Ω(t). Solid marked with gray, fluid marked with dots, lines represent transformation of material points by $\mathbf{A}_s(t)$.

Using standard notations for Sobolev spaces, we indicate with $Q = L^2(\Omega)$, $\mathcal{V}_0 := \{\mathbf{v} \in H^1(\Omega)^d \text{ s.t. } \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D\}$, the affine space $\mathcal{V}_D := \{\mathbf{v} \in H^1(\Omega)^d \text{ s.t. } \mathbf{v} = \mathbf{v}_D^* \text{ on } \Gamma_D\}$, and we indicate with $\mathcal{V}_0'$ the dual space of $\mathcal{V}_0$. With the conditions expressed in (5), for any time t, we can formally derive a global weak form of the conservation equations for both the solid and the fluid as

Problem 1 (Formal weak formulation of the conservation equations). *Given* $\mathbf{g}(t) \in \mathcal{V}_0'$ *and* $\boldsymbol{\tau}^*(t) \in H^{-1/2}(\Gamma_N)^d$ *for each time t in* $[0, T]$, *find* $(\mathbf{v}, \rho) \in \mathcal{V}_D \times Q$ *such that*

$$\begin{aligned} &\int_\Omega \rho \frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} \cdot \boldsymbol{\phi} \,\mathrm{d}\mathbf{x} + \int_\Omega \boldsymbol{\sigma} : \epsilon(\boldsymbol{\phi}) \,\mathrm{d}\mathbf{x} = \int_\Omega \mathbf{g} \cdot \boldsymbol{\phi} \,\mathrm{d}\mathbf{x} + \int_{\Gamma_N} \boldsymbol{\tau}^* \cdot \boldsymbol{\phi} \,\mathrm{d}s \qquad && \forall \boldsymbol{\phi} \in \mathcal{V}_0 \\ &\int_\Omega \frac{\mathrm{D}\rho}{\mathrm{D}t} q \,\mathrm{d}\mathbf{x} + \int_\Omega \rho \,\mathrm{div}(\mathbf{v}) q \,\mathrm{d}\mathbf{x} = 0 && \forall q \in Q, \end{aligned} \tag{6}$$

where $\epsilon(\boldsymbol{\kappa}) := \frac{1}{2}(\mathrm{grad}\ \boldsymbol{\kappa} + (\mathrm{grad}\ \boldsymbol{\kappa})^T)$ *denotes the symmetric gradient of a vector field* $\boldsymbol{\kappa}$.

In order to close the system and transform the *formal* definition of the weak formulation above in a well posed problem, we will need to provide constitutive equations, discussed in Section 2.3, initial conditions, and a suitable representation for the evolution of the domain $\Omega(t)$, discussed in the next section.

2.2 | Arbitrary Lagrangian–Eulerian formulation

We concretize the representation of the time-changing domain $\Omega(t)$ (Figure 1), by introducing a diffeomorphism $\mathbf{A}(t) : \hat{\Omega} \to \mathbb{R}^d$—the ALE map—that maps points $\hat{\mathbf{x}}$ in the reference domain $\hat{\Omega}$ to points $\mathbf{x}(t, \hat{\mathbf{x}}) = \mathbf{A}(t;\ \hat{\mathbf{x}})$ on the deformed domain $\Omega(t)$, and such that $\Omega_s(t) = \mathbf{A}(t; \hat{\Omega}_s)$ and $\Omega_f(t) = \mathbf{A}(t; \hat{\Omega}_f)$.

In general, we are free to choose the map $\mathbf{A}(t)$ arbitrarily, provided that the solid and fluid domains are mapped correctly at each time t. We choose an ALE map that coincides with the solid deformation map in the solid domain $\hat{\Omega}_s$. On $\hat{\Omega}_f$, we set $\mathbf{A}(t)$ as a pseudo-elastic extension in the fluid domain $\hat{\Omega}_f$.[13] Finally, we assume the ALE transformation equals the identity at time $t = 0$, that is, $\mathbf{A}(t = 0; \hat{\mathbf{x}}) = \hat{\mathbf{x}}$ in $\hat{\Omega} \equiv \Omega(0)$.

We indicate with $\hat{\mathbf{u}}(t, \hat{\mathbf{x}}) = \mathbf{A}(t; \hat{\mathbf{x}}) - \hat{\mathbf{x}}$ the pseudo-displacement field that represents the domain deformation, and with $\hat{\mathbf{w}}$ its time derivative, representing the domain velocity, and we adopt consistently the $\hat{\cdot}$ convention to indicate functions defined on $\hat{\Omega}$, or the pullback through $\mathbf{A}(t)$ of Eulerian fields defined on $\Omega(t)$ to $\hat{\Omega}$, that is, $\hat{\boldsymbol{\kappa}} = \boldsymbol{\kappa} \circ \mathbf{A}$ and $\boldsymbol{\kappa} = \hat{\boldsymbol{\kappa}} \circ \mathbf{A}^{-1}$, and "Grad" and "Div" to indicate differential operators w.r.t. $\hat{\mathbf{x}}$. Subscripts s, f indicate the restriction of the field to the solid or fluid domains $\hat{\Omega}_s, \hat{\Omega}_f$ or $\Omega_s(t), \Omega_f(t)$.

We use a Lagrangian setting in the solid domain, and let $\hat{\mathbf{u}}_s$ coincide with the displacement of solid particles in $\hat{\Omega}_s$, so that $\partial_t \hat{\mathbf{u}}_s = \hat{\mathbf{v}}_s$ in $\hat{\Omega}_s$. In the domain $\hat{\Omega}_f$, at every time t, we construct a pseudo-elastic extension $\hat{\mathbf{u}}_f$ of $\hat{\mathbf{u}}_s$ onto $\hat{\Omega}_f$, namely

$$\hat{\mathbf{u}}_f = \mathrm{Ext}(\hat{\mathbf{u}}_s), \tag{7}$$

where $\mathrm{Ext}(\hat{\mathbf{u}}_s)$ is defined as the solution $\hat{\mathbf{u}}_f$ to

$$\begin{cases} -\,\mathrm{Div}(\mu_A \mathrm{E}(\hat{\mathbf{u}}_f)) = \mathbf{0} & \text{in } \hat{\Omega}_f \\ \hat{\mathbf{u}}_f = \hat{\mathbf{u}}_s & \text{on } \partial\hat{\Omega}_s \\ \hat{\mathbf{u}}_f = \mathbf{0} & \text{on } \partial\hat{\Omega}_f \setminus \partial\hat{\Omega}_s. \end{cases} \tag{8}$$

Here $\mathrm{E}(\hat{\mathbf{u}}_f) = \frac{1}{2}(\mathrm{Grad}\ \hat{\mathbf{u}}_f + (\mathrm{Grad}\ \hat{\mathbf{u}}_f)^T)$ is the symmetric gradient of $\hat{\mathbf{u}}_f$ w.r.t. the coordinates $\hat{\mathbf{x}}$, μ_A is a possibly variable coefficient, and for the sake of exposition we assume that the boundary of the fluid domain is not moving, even though arbitrary deformations could be applied to $\partial\hat{\Omega}_f \setminus \partial\hat{\Omega}_s$. Other (more computationally intensive) choices for the definition of Ext are possible, for example, using a biharmonic extension.[13,44,45]

The equations of motion (3) are complemented with an evolution equation for $\hat{\mathbf{u}}$ with zero initial conditions, defined through the domain velocity field $\hat{\mathbf{w}}$, that is,

$$\hat{\mathbf{w}} := \begin{cases} \hat{\mathbf{v}}_s & \text{in } \hat{\Omega}_s \\ \frac{\partial \mathrm{Ext}(\hat{\mathbf{u}}_s)}{\partial t} & \text{in } \hat{\Omega}_f, \end{cases} \qquad \begin{cases} \frac{\partial \hat{\mathbf{u}}}{\partial t}(t,\hat{\mathbf{x}}) = \hat{\mathbf{w}}(t,\hat{\mathbf{x}}) & \text{in } \hat{\Omega} \\ \hat{\mathbf{u}}(t=0,\hat{\mathbf{x}}) = \mathbf{0} & \hat{\mathbf{x}} \in \hat{\Omega}. \end{cases} \tag{9}$$

Notice that, strictly speaking, only the equation for $\hat{\mathbf{u}}_s$ should be interpreted as an ODE with zero initial condition, while the equation for $\hat{\mathbf{w}}_f$ is simply a time derivative evaluation.

We rewrite the weak formulation of the equations of motion in ALE form by introducing the ALE time derivative in Eulerian coordinates, defined as the time derivative of a field at fixed $\hat{\mathbf{x}}$. For an Eulerian vector field $\boldsymbol{\kappa}$ or an Eulerian scalar field α, we have

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{\kappa}}{\mathrm{d}t}(t,\mathbf{A}(t;\hat{\mathbf{x}})) &:= \frac{\partial\hat{\boldsymbol{\kappa}}}{\partial t}(t,\hat{\mathbf{x}}) = \frac{\partial\boldsymbol{\kappa}}{\partial t}(t,\mathbf{A}(t;\hat{\mathbf{x}})) + \big((\mathrm{grad}\ \boldsymbol{\kappa})\mathbf{w}\big)(t,\mathbf{A}(t;\hat{\mathbf{x}})), \\ \frac{\mathrm{d}\alpha}{\mathrm{d}t}(t,\mathbf{A}(t;\hat{\mathbf{x}})) &:= \frac{\partial\hat{\alpha}}{\partial t}(t,\hat{\mathbf{x}}) = \frac{\partial\alpha}{\partial t}(t,\mathbf{A}(t;\hat{\mathbf{x}})) + \big(\mathbf{w}\cdot\mathrm{grad}\ \alpha\big)(t,\mathbf{A}(t;\hat{\mathbf{x}})), \end{aligned} \tag{10}$$

giving the following relation with the material time derivative:

$$\begin{aligned} \frac{\mathrm{D}\boldsymbol{\kappa}}{\mathrm{D}t} &= \frac{\mathrm{d}\boldsymbol{\kappa}}{\mathrm{d}t} + (\mathrm{grad}\ \boldsymbol{\kappa})(\mathbf{v}-\mathbf{w}), \\ \frac{\mathrm{D}\alpha}{\mathrm{D}t} &= \frac{\mathrm{d}\alpha}{\mathrm{d}t} + (\mathrm{grad}\ \alpha)\cdot(\mathbf{v}-\mathbf{w}), \end{aligned} \tag{11}$$

which coincide with the partial time derivative in the Lagrangian case ($\mathbf{v} = \mathbf{w}$) and with the classical material time derivative in the purely Eulerian case ($\mathbf{w} = 0$).

We consider the mapped Sobolev spaces $\hat{\mathcal{Q}} := \mathcal{Q}\circ\mathbf{A}$, $\hat{\mathcal{V}} := \mathcal{V}\circ\mathbf{A}$, and define, moreover, the space $\hat{\mathcal{V}}_0 := H^1_0(\hat{\Omega})^d$.

The weak form of the complete FSI problem in ALE form then reads

Problem 2 (Formal ALE weak formulation of conservation equations). *Given* $\mathbf{g} \in \mathcal{V}_0'$ *and* $\boldsymbol{\tau}^* \in H^{-1/2}(\Gamma_N)^d$, *for each time t in* $[0,T]$, *find* $(\mathbf{v},\rho,\hat{\mathbf{u}}) \in \mathcal{V}_D \times \mathcal{Q} \times \hat{\mathcal{V}}$ *such that*

$$\begin{aligned} &\int_\Omega \rho\Big(\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} + (\mathrm{grad}\ \ \mathbf{v})(\mathbf{v}-\mathbf{w})\Big)\cdot\boldsymbol{\phi}\,\mathrm{d}\mathbf{x} + \int_\Omega \boldsymbol{\sigma} : \epsilon(\boldsymbol{\phi})\,\mathrm{d}\mathbf{x} = \int_\Omega \mathbf{g}\cdot\boldsymbol{\phi}\,\mathrm{d}\mathbf{x} + \int_{\Gamma_N} \boldsymbol{\tau}^*\cdot\boldsymbol{\phi}\,\mathrm{d}s && \forall\boldsymbol{\phi}\in\mathcal{V}_0 \\ &\int_\Omega \left(\frac{\mathrm{d}\rho}{\mathrm{d}t} + (\mathbf{v}-\mathbf{w})\cdot\mathrm{grad}\ \rho\right)q\,\mathrm{d}\mathbf{x} + \int_\Omega \rho\ \mathrm{div}(\mathbf{v})q\,\mathrm{d}\mathbf{x} = 0 && \forall q\in\mathcal{Q} \\ &\int_{\hat{\Omega}_f} \mu_A \mathrm{E}(\hat{\mathbf{u}}) : \mathrm{E}(\hat{\boldsymbol{\phi}})\,\mathrm{d}\hat{\mathbf{x}} + \int_{\hat{\Omega}_s} \left(\frac{\partial\hat{\mathbf{u}}}{\partial t} - \hat{\mathbf{w}}\right)\cdot\hat{\boldsymbol{\phi}}\,\mathrm{d}\hat{\mathbf{x}} = 0 && \forall\hat{\boldsymbol{\phi}}\in\hat{\mathcal{V}}, \end{aligned} \tag{12}$$

with $\mathbf{w}$ *defined in* (9).

For simplicity of exposition, the first two equations in (12) *are written in Eulerian form, but are solved on the reference domain* $\hat{\Omega}$, *using the spaces* $\hat{\mathcal{V}}$, *and* $\hat{\mathcal{Q}}$.

From the numerical point of view, it may be convenient to express the dependency between $\hat{\mathbf{u}}$ and $\hat{\mathbf{w}}$ in a strong form, rather than using the weak form presented in (12). Time discretization and how to write the time evolution of the grid is presented in Section 3.1. Similarly to Problem 1, also Problem 2 is at this stage only formally defined: in order to close the system we need to provide constitutive equations.

2.3 | Constitutive models

2.3.1 | Incompressible Newtonian fluid

We consider a classic incompressible Newtonian fluid with constant density ρ_f so that

$$\boldsymbol{\sigma}_f = 2\eta_f \epsilon(\mathbf{v}_f) - p_f \mathbf{I}, \tag{13}$$

where $p_f = -\frac{1}{3}\mathrm{tr}\boldsymbol{\sigma}_f$ is the fluid pressure, η_f is the viscosity and $\mathbf{I}$ is the identity tensor. In this particular case, mass conservation given in Equation (3) reduces to volume conservation, namely,

$$\mathrm{div}\,\mathbf{v}_f = 0. \tag{14}$$

We note that the pressure p_f in Equation (13) plays the role of a Lagrange multiplier enforcing the incompressibility constraint (14), and is solved for in place of the density ρ_f which is, in this constitutive model, a material constant of the fluid.

2.3.2 | Incompressible Mooney–Rivlin solid

We assume that the solid is an incompressible hyperelastic material governed by the Mooney–Rivlin model.[46,47] In the solid domain, we define the deformation gradient

$$\hat{\mathbf{F}} := \mathrm{Grad}\,\mathbf{A}_s = \mathbf{I} + \mathrm{Grad}\,\hat{\mathbf{u}}_s, \tag{15}$$

and we have $\mathbf{F} = \hat{\mathbf{F}} \circ \mathbf{A}_s^{-1}$. The inverse deformation gradient is

$$\mathbf{F}^{-1} = \mathrm{grad}\left(\mathbf{A}_s^{-1}\right) = \mathbf{I} - \mathrm{grad}\,\mathbf{u}_s. \tag{16}$$

Incompressibility implies the following constraint on the determinant of $\hat{\mathbf{F}}$, $\hat{J} = \det(\hat{\mathbf{F}})$ $(J = \hat{J} \circ \mathbf{A}_s^{-1})$,

$$\hat{J} = 1, \tag{17}$$

which is equivalent to $\mathrm{div}\mathbf{v}_s = 0$.

In the case of an incompressible Mooney–Rivlin solid, the Cauchy stress can be expressed using the left Cauchy–Green deformation tensor $\mathbf{B} := \mathbf{F}\mathbf{F}^T$ as follows[47]

$$\boldsymbol{\sigma}_s = \mu_1 \mathbf{B} - \mu_2 \mathbf{B}^{-1} - p_s^* \mathbf{I}, \tag{18}$$

where p_s^* is again a Lagrange multiplier that enforces incompressibility and $\mu_1 \geq 0$ and $\mu_2 \geq 0$ are material parameters such that $\mu_s = \mu_1 + \mu_2 > 0$ is the shear modulus in the reference configuration. For $\mu_2 = 0$, the incompressible neo-Hookean model is obtained as a special case of the incompressible Mooney–Rivlin model.

The Cauchy stress (18) can be equivalently expressed as

$$\boldsymbol{\sigma}_s = \mu_1(\mathbf{B} - \mathbf{I}) + \mu_2(\mathbf{I} - \mathbf{B}^{-1}) - p_s \mathbf{I}, \qquad p_s = p_s^* - \mu_1 + \mu_2, \tag{19}$$

where p_s is a shifted Lagrange multiplier that vanishes whenever $\boldsymbol{\sigma}_s = \mathbf{0}$. Note, however, that p_s is not equal to the solid pressure, that is, $p_s \neq -\frac{1}{3}\mathrm{tr}\boldsymbol{\sigma}_s$.

Substituting Equations (15) and (16) into Equation (19) the Cauchy stress over the solid domain can now be expressed in the following form:

$$\boldsymbol{\sigma}_s = \mu_1\left(\left(2\mathrm{E}(\hat{\mathbf{u}}_s) + \mathrm{Grad}\,\hat{\mathbf{u}}_s(\mathrm{Grad}\,\hat{\mathbf{u}}_s)^T\right) \circ \mathbf{A}_s^{-1}\right) + \mu_2\left(2\epsilon(\mathbf{u}_s) - (\mathrm{grad}\,\mathbf{u}_s)^T \mathrm{grad}\,\mathbf{u}_s\right) - p_s \mathbf{I}. \tag{20}$$

We note that the second and the third term on the right-hand side are evaluated entirely in the current configuration. At the same time, the first term involves evaluation of the displacement gradient in the reference configuration, and the corresponding terms are then pushed forward through $\mathbf{A}_s^{-1}$ to the current configuration, where the integration is performed. In the present implementation, we restrict ourselves to the special case of $\mu_1 = 0$ and $\mu_2 = \mu_s$, so that the first term in Equation (20) vanishes. Note that in 2D plane-strain problems the incompressible Mooney–Rivlin model is equivalent to the neo-Hookean model with one shear modulus $\mu_s = \mu_1 + \mu_2$,[47] hence the above assumption affects only 3D cases.

2.4 | Incompressible fluid–structure interaction problem in ALE form

Let us summarize the results from this section by presenting the weak form of the FSI problem in the ALE frame of reference. In the incompressible formulation we are using in this work, the conservation of mass transforms to conservation of volume. The (possibly different) densities ρ_s and ρ_f are constitutive constants of the fluid and of the solid, and the primal unknowns are the velocity field $\mathbf{v}$, the pressure-like field p (acting as a Lagrange multiplier for the incompressibility constraints on both the solid and the fluid) and the pseudo-displacement $\hat{\mathbf{u}}$, corresponding to the solid displacement in Ω_s and to the domain displacement in Ω_f.

Problem 3 (ALE weak form of the incompressible FSI problem). *Given* $\mathbf{g} \in \mathcal{V}_0'$ *and* $\boldsymbol{\tau}^* \in H^{-1/2}(\Gamma_N)^d$, *for each time* t *in* $[0, T]$, *find* $(\mathbf{v}, p, \hat{\mathbf{u}}) \in \mathcal{V}_D \times \mathcal{Q} \times \hat{\mathcal{V}}$ *such that*

$$\begin{aligned}
&\int_\Omega \rho\left(\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} + (\mathrm{grad}\,\mathbf{v})(\mathbf{v}-\mathbf{w})\right)\cdot\boldsymbol{\phi}\,\mathrm{d}\mathbf{x} + \int_\Omega \boldsymbol{\sigma} : \epsilon(\boldsymbol{\phi})\,\mathrm{d}\mathbf{x} = \int_\Omega \mathbf{g}\cdot\boldsymbol{\phi}\,\mathrm{d}\mathbf{x} + \int_{\Gamma_N} \boldsymbol{\tau}^*\cdot\boldsymbol{\phi}\,\mathrm{d}s && \forall\boldsymbol{\phi}\in\mathcal{V}_0\\
&\int_\Omega \mathrm{div}(\mathbf{v})\,q\,\mathrm{d}\mathbf{x} = 0 && \forall q\in\mathcal{Q} \qquad (21)\\
&\int_{\hat{\Omega}_f} \mu_A \mathrm{E}(\hat{\mathbf{u}}) : \mathrm{E}(\hat{\boldsymbol{\phi}})\,\mathrm{d}\hat{\mathbf{x}} + \int_{\hat{\Omega}_s}\left(\frac{\partial\hat{\mathbf{u}}}{\partial t} - \hat{\mathbf{v}}\right)\cdot\hat{\boldsymbol{\phi}}\,\mathrm{d}\hat{\mathbf{x}} = 0 && \forall\hat{\boldsymbol{\phi}}\in\hat{\mathcal{V}},
\end{aligned}$$

where, according to the definition in Equation (9), $\hat{\mathbf{w}}_f = \frac{\partial \mathrm{Ext}(\hat{\mathbf{u}}_s)}{\partial t}$ *and* $\mathbf{w}_s = \mathbf{v}_s$. *The Cauchy stress* $\boldsymbol{\sigma}$ *takes the form*:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{v}, \mathbf{u}, p) = p\mathbf{I} + \begin{cases} 2\eta_f \epsilon(\mathbf{v}_f) & \text{in } \Omega_f \\ 2\mu_s\epsilon(\mathbf{u}_s) - \mu_s(\mathrm{grad}\,\mathbf{u}_s)^T \mathrm{grad}\,\mathbf{u}_s & \text{in } \Omega_s. \end{cases} \qquad (22)$$

The subdomains Ω_s *and* Ω_f *appearing in integrals in Equation* (21) *are defined as the images of the undeformed ones*: $\Omega_s = \mathbf{A}(\hat{\Omega}_s)$ *and* $\Omega_f = \mathbf{A}(\hat{\Omega}_f)$, *even though the computation is formally performed on the reference domain* $\hat{\Omega}$. *Additionally, the following initial conditions have to be met*:

$$\begin{cases} \mathbf{v}_s(t=0,\ \hat{\mathbf{x}}) = \mathbf{v}_s^* & \text{in } \hat{\Omega}_s \\ \hat{\mathbf{u}}_s(t=0,\ \hat{\mathbf{x}}) = \mathbf{u}_s^* & \text{in } \hat{\Omega}_s \\ \mathbf{v}_f(t=0,\ \hat{\mathbf{x}}) = \mathbf{v}_f^* & \text{in } \hat{\Omega}_f. \end{cases} \qquad (23)$$

This formulation is similar to the ones appearing in the literature,[12,33,36] the major difference being the incompressibility assumption on the solid constitutive equations. At this stage the problem is still fully nonlinear. The most difficult nonlinearity comes from the fact that all domains are dependent on the (unknown) mapping $\mathbf{A}$, which in turn is hidden also in the definition of the "grad" and "div" differential operators, when they are expressed in the reference domain $\hat{\Omega}$.

Note that the interface conditions $(5)_{1,2}$ are naturally included in the weak form $(21)_1$, and likewise in its earlier formal definitions $(6)_1$ and $(12)_1$. The kinematic continuity condition $(5)_2$ is satisfied automatically thanks to the continuity of the velocity $\mathbf{v}$ (and of the pseudo-displacement $\hat{\mathbf{u}}$, which ensures the consistency of the ALE mapping). On the other hand, by virtue of the continuity of the test function $\boldsymbol{\phi}$, the equality of traction vectors at the interface, Equation $(5)_1$, is satisfied in a weak manner, see the related discussions in References 48-50.

3 | DISCRETIZATION

3.1 | Time integration scheme

We consider a uniform time discretization of the time interval $[0, T]$ with a set of equidistant points $\{t^0, \dots, t^N\}$ where $t^n = n\tau$, with the time step size $\tau = T/N$. At the nth time step t^n, we define $\hat{\mathbf{v}}^n$, $\hat{\mathbf{u}}_s^n$, and $\hat{\mathbf{u}}^n$ as an approximation of $\hat{\mathbf{v}}$, $\hat{\mathbf{u}}_s$, and $\hat{\mathbf{u}}$, respectively. The approximate pseudo-displacement defines the approximation $\mathbf{A}^n$ of the mapping $\mathbf{A}(t^n;\cdot)$:

$$\mathbf{A}(t^n;\hat{\mathbf{x}}) \approx \mathbf{A}^n(\hat{\mathbf{x}}) = \hat{\mathbf{x}} + \hat{\mathbf{u}}^n(\hat{\mathbf{x}}) \qquad \hat{\mathbf{x}} \in \hat{\Omega}, \tag{24}$$

which in turn defines the approximate domains:

$$\Omega(t^n) \approx \Omega^n := \mathbf{A}^n(\hat{\Omega}), \qquad \Omega_s(t^n) \approx \Omega_s^n := \mathbf{A}^n(\hat{\Omega}_s), \qquad \Omega_f(t^n) := \Omega^n \setminus \Omega_s^n. \tag{25}$$

At the time step n, the relation between the point $\mathbf{x}^n \in \Omega^n$ and the point $\hat{\mathbf{x}} \in \hat{\Omega}$ is defined by the mapping $\mathbf{A}^n$, that is: $\mathbf{x}^n(\hat{\mathbf{x}}) = \mathbf{A}^n(\hat{\mathbf{x}})$. That is, the spatial derivatives

$$\mathrm{grad}\ \kappa = \frac{\partial}{\partial x}\kappa, \qquad \epsilon(\kappa) = \frac{1}{2}\left(\mathrm{grad}\ \kappa + (\mathrm{grad}\ \kappa)^T\right) \tag{26}$$

are associated with transformation $\mathbf{A}^n$. Since $\mathbf{A}^n$ is defined by a pseudo-displacement, those derivatives are implicitly defined by $\hat{\mathbf{u}}^n$.

We approximate the time derivative of an arbitrary field $\hat{\kappa}$ defined on $\hat{\Omega} \times [0, T]$ by the k-step BDF,[51]

$$\frac{\partial\hat{\kappa}(t = t^n, \hat{\mathbf{x}})}{\partial t} \approx \partial_\tau \hat{\kappa}^n(\hat{\mathbf{x}}) := \frac{1}{\gamma\tau}\sum_{i=0}^{k} \alpha_i \hat{\kappa}^{n-i}(\hat{\mathbf{x}}), \tag{27}$$

with the coefficients normalized so that $\alpha_0 = 1$. Here, we restrict ourselves to $k \le 2$. The case $k = 1$ corresponds to the classical implicit Euler scheme ($\gamma = 1$, $\alpha_1 = -1$), while for $k = 2$ there holds $\gamma = \frac{2}{3}$, $\alpha_1 = -\frac{4}{3}$, $\alpha_2 = \frac{1}{3}$.

Therefore, at $t = t^n$, we will have

$$\frac{\partial \hat{\mathbf{u}}}{\partial t} \approx \partial_\tau \hat{\mathbf{u}}^n, \qquad \hat{\mathbf{w}}_f = \frac{\partial \mathrm{Ext}(\hat{\mathbf{u}}_s)}{\partial t} \approx \hat{\mathbf{w}}_f^n := \partial_\tau \mathrm{Ext}(\hat{\mathbf{u}}_s^n).$$

For the ALE time derivative of the velocity, we follow the identity $\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}(t, \mathbf{x}) = \frac{\partial\hat{\mathbf{v}}}{\partial t}(t, \hat{\mathbf{x}})$ and approximate, compare for example, References 5,36, and 13, at $\mathbf{x}^n = \mathbf{A}^n(\hat{\mathbf{x}})$, by moving back to the reference configuration:

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}(t = t^n, \mathbf{x}^n) \approx \mathrm{d}_\tau \mathbf{v}^n(\mathbf{x}^n) := \partial_\tau \hat{\mathbf{v}}^n(\hat{\mathbf{x}}) = \frac{1}{\gamma\tau}\sum_{i=0}^{k}\alpha_i \mathbf{v}^{n,i}(\mathbf{x}^n), \tag{28}$$

where we set $\mathbf{v}^{n,i} := \mathbf{v}^{n-i} \circ \mathbf{A}^{n-i} \circ (\mathbf{A}^n)^{-1}$.

After replacing the time derivatives in (21) by their approximates, we obtain an implicit time integration scheme:

Problem 4 (Fully implicit BDF time discretization of ALE weak form of the incompressible FSI problem). *Given* $\mathbf{g} \in \mathcal{V}_0'$, $\boldsymbol{\tau}^* \in H^{-1/2}(\Gamma_N)^d$, *and initial conditions* $(\mathbf{v}^0, \hat{\mathbf{u}}^0) \in \mathcal{V}_D^0 \times \hat{\mathcal{V}}^0$, *find* $(\mathbf{v}^n, p^n, \hat{\mathbf{u}}^0) \in \mathcal{V}_D^n \times \mathcal{Q}^0 \times \hat{\mathcal{V}}$ *for each time* t^n, $n = 1, \dots, N$, *such that*

$$\begin{aligned}
&\int_{\Omega^n} \rho\left(\mathrm{d}_{\tau\mathbf{v}^n} + (\mathrm{grad}\ \mathbf{v}^n)(\mathbf{v}^n - \mathbf{w}^n)\right)\cdot\boldsymbol{\phi}\ \mathrm{d}\mathbf{x} + \int_{\Omega^n} \boldsymbol{\sigma}^n : \epsilon(\boldsymbol{\phi})\ \mathrm{d}\mathbf{x} = \int_{\Omega^n} \mathbf{g}\cdot\boldsymbol{\phi}\ \mathrm{d}\mathbf{x} + \int_{\Gamma_N} \boldsymbol{\tau}^*\cdot\boldsymbol{\phi}\ \mathrm{d}s && \forall\boldsymbol{\phi} \in \mathcal{V}_0^n \\
&\int_{\Omega^n} \mathrm{div}(\mathbf{v}^n)\ q\ \mathrm{d}\mathbf{x} = 0 && \forall q \in \mathcal{Q}^n \qquad (29) \\
&\int_{\hat{\Omega}_f} \mu_A \mathrm{E}(\hat{\mathbf{u}}^n) : \mathrm{E}(\hat{\boldsymbol{\phi}})\ \mathrm{d}\hat{\mathbf{x}} + \int_{\hat{\Omega}_s} \left(\partial_\tau\hat{\mathbf{u}}^n - \hat{\mathbf{v}}^n\right)\cdot\hat{\boldsymbol{\phi}}\ \mathrm{d}\hat{\mathbf{x}} = 0 && \forall\hat{\boldsymbol{\phi}} \in \hat{\mathcal{V}},
\end{aligned}$$

with $\hat{\mathbf{w}}_f^n = \partial_\tau \mathrm{Ext}(\hat{\mathbf{u}}_s^n)$ *and* $\mathbf{w}_s^n = \mathbf{v}_s^n$; *the stress* $\boldsymbol{\sigma}^n = \boldsymbol{\sigma}(\mathbf{v}^n, \mathbf{u}^n, p^n)$ *specified in* (22) *and the initial data prescribed by* (23). *Here* $\mathcal{V}_D^n$ *and* $\mathcal{Q}^n$ *denote the corresponding spaces over domain* Ω^n, *which is implicitly defined by* (24) *and* (25)*. If a scheme of order* $k > 1$ *is used then first* $k-1$ *steps have to be done with lower order schemes so that* $(\mathbf{v}^i, \hat{\mathbf{u}}^i)$, $i = 1, \dots, k-1$ *are obtained.*

Note that the last equation of (29) effectively splits into two,

$$\hat{\mathbf{u}}_f^n = \text{Ext}(\hat{\mathbf{u}}_s^n), \qquad \partial_\tau \hat{\mathbf{u}}_s^n = \hat{\mathbf{v}}_s^n.$$

From the latter and the linearity of Ext it follows that

$$\hat{\mathbf{w}}_f^n = \partial_\tau \text{Ext}(\hat{\mathbf{u}}_s^n) = \text{Ext}(\partial_\tau \hat{\mathbf{u}}_s^n) = \text{Ext}(\hat{\mathbf{v}}_s^n). \tag{30}$$

3.1.1 | Semi-implicit scheme

Problem 4 needs be solved on every time step, it is nonlinear, and it is usually solved by the Newton's method[23,52,53] or a fixed-point method (without significant impact on stability, as demonstrated in Reference 54).

Here, we take a different approach inspired by Crosetto et al.,[5] Xu and Yang,[36] and Badia et al.,[34] and simplify the nonlinear problem by splitting the solid displacement into an explicitly predicted displacement and an implicit velocity dependent part. We exploit the structure of Problem 4 and in particular of the Cauchy stress definition (22) to expose a solution strategy based on fixed point iterations defined through a semi-implicit splitting of the BDF scheme defined in (27) for a simpler problem. The current pseudo-displacement and the current ALE velocity are affinely equivalent, and we expose this dependency together with the dependency on previously computed solutions $(\hat{\mathbf{u}}^{\#}, \hat{\mathbf{w}}^{\#})$ by expressing the currently unknown velocity and pseudo displacement $(\hat{\mathbf{u}}^{\square}, \hat{\mathbf{w}}^{\square})$ as

$$\hat{\mathbf{u}}^{\square} := \gamma\tau\hat{\mathbf{w}}^{\square} - \sum_{i=1}^{k} \alpha_i \hat{\mathbf{u}}^{n-i} = \gamma\tau\hat{\mathbf{w}}^{\square} + (\hat{\mathbf{u}}^{\#} - \gamma\tau\hat{\mathbf{w}}^{\#}), \tag{31}$$

where $\hat{\mathbf{w}}^{\#}$ is the most recently computed velocity (i.e., the velocity from the previous time step or from a previous fixed point iteration) and

$$\hat{\mathbf{u}}^{\#} := \gamma\tau\hat{\mathbf{w}}^{\#} - \sum_{i=1}^{k} \alpha_i \hat{\mathbf{u}}^{n-i} \tag{32}$$

represents an explicit approximation of the pseudo-displacement.

This splitting is based on an explicit prediction $\mathbf{u}^{\#}$ of the next displacement, to be used in the computation of a temporary $\mathbf{A}^{\#}$, and thus $\Omega^{\#}$, and in an explicit computation of nonlinear correction terms of the solid Cauchy stress. The remaining velocity dependent part is solved for implicitly, resulting in the following splitting of the global Cauchy stress

$$\boldsymbol{\sigma} = p\mathbf{I} + 2\eta\epsilon(\mathbf{v}^{\square}) + \begin{cases} 0 & \text{in } \Omega_f^{\#} \\ \mu_s\left[2\epsilon\left(\sum_{i=1}^{k}\alpha_i\mathbf{u}^{n,i}\right) - (\text{grad } \mathbf{u}_s^{\#})^T\text{grad } \mathbf{u}_s^{\#}\right] & \text{in } \Omega_s^{\#}, \end{cases} \tag{33}$$

where

$$\eta = \begin{cases} \eta_f & \text{in } \Omega_f^{\#} \\ \gamma\tau\mu_s & \text{in } \Omega_s^{\#}, \end{cases} \tag{34}$$

making the problem equivalent to a Stokes-like system on the domain $\Omega^{\#}$, with jumps in the viscosities across the interface Γ_i. We remark here that such jumps have a ratio of six to nine orders of magnitude. The resolution of the solid problem in terms of solid velocities is classical, but it is usually discarded from the computational point of view, due to the lack of efficient solvers for such high contrast coefficients.

The key contribution of this work comes from exploiting the robust preconditioner for Stokes-like systems with high-contrast jumps in the viscosities developed in References 37 and 40 also for the solution of FSI problems; see Section 4.

The final scheme, $\mathrm{GCsI}_k(M_1, \dots, M_S)$ (*geometry-convective semi-implicit*), which computes the solution on the next time step, is defined in Algorithm 1; its concept *loosely* resembles an S-stage predictor–corrector scheme. Here, k and $M_1, \dots, M_S$ are parameters of the scheme, with k corresponding to the order of the BDF formula (27). The algorithm proceeds in stages, according to prescribed operational mode M_s, where $s = 1, \dots, S$. We consider two modes: if $M_s = \mathcal{E}$, the scheme computes the geometry and convective terms in an explicit way, while for $M_s = \mathcal{I}$, it treats the convective term more implicitly (the details are provided later in this section). The simplest and cheapest scheme of this kind, a one-stage $\mathrm{GCsI}_k(\mathcal{E})$ method, coincides with the kth order GCE scheme of Reference 5, so our scheme can be considered a generalization of this approach.

Although schemes such as one-stage $\mathrm{GCsI}_k(\mathcal{I})$ or two-stage $\mathrm{GCsI}_k(\mathcal{E}, \mathcal{I})$ are more costly per time step than GCE, by adding more implicit stages to the scheme we stabilize the method, therefore allowing for significantly larger time steps as compared to the GCE scheme. This improves the overall performance of the solver—see Section 5, where we discuss the results of numerical experiments.

Algorithm 1. kth order, S–stage $\mathrm{GCsI}_k(M_1, \dots, M_S)$ scheme. The bilinear forms a_M, b and functionals $g_{\mathbf{v}M}$, and g_p are defined in (36)–(38), and (47), respectively

```
Data: $\hat{\mathbf{u}}^{n-i}, \hat{\mathbf{v}}^{n-i}, \hat{\mathbf{w}}^{n-i}, \quad i = 1, \dots, k$
Result: $\hat{\mathbf{u}}^{n}, \hat{\mathbf{v}}^{n}, \hat{\mathbf{w}}^{n}$
begin
    $\hat{\mathbf{w}}^{\#} := \hat{\mathbf{w}}^{n-1}$
    for $M$ in $(M_1, \dots, M_S)$ do
        $\hat{\mathbf{u}}^{\#} := \gamma\tau\hat{\mathbf{w}}^{\#} - \sum_{i=1}^{k} \alpha_i \hat{\mathbf{u}}^{n-k}$                ◁ Explicit step
        $\mathbf{A}^{\#} = \mathbf{I} + \hat{\mathbf{u}}^{\#}, \qquad \Omega^{\#} = \mathbf{A}^{\#}(\hat{\Omega})$                ◁ New geometry
        Find $\mathbf{v}^{\square} \in H^1(\Omega^{\#})$ and $p^{\square} \in L^2(\Omega^{\#})$ :                ◁ M-mode step
            $\begin{cases} a_M(\mathbf{v}^{\square}, \boldsymbol{\phi}) + b(\boldsymbol{\phi}, p^{\square}) = g_{\mathbf{v}M}(\boldsymbol{\phi}) & \forall \boldsymbol{\phi} \in H^1_D(\Omega^{\#}), \\ b(\mathbf{v}^{\square}, q) = g_p(q) & \forall q \in L^2(\Omega^{\#}). \end{cases} \qquad (35)$
        $\hat{\mathbf{w}}^{\square} := \mathrm{Ext}(\hat{\mathbf{v}}^{\square})$                ◁ Extension
        $\hat{\mathbf{w}}^{\#} := \hat{\mathbf{w}}^{\square}$
    end
    $\hat{\mathbf{v}}^{n} := \hat{\mathbf{v}}^{\square}, \qquad \hat{\mathbf{w}}^{n} := \hat{\mathbf{w}}^{\square}$
    $\hat{\mathbf{u}}^{n} := \gamma\tau\hat{\mathbf{w}}^{n} - \sum_{i=1}^{k} \alpha_i \hat{\mathbf{u}}^{n-i}$                ◁ Displacement recovery
end
```

The bilinear forms $a_M(\cdot, \cdot)$, $b(\cdot, \cdot)$ and functionals $g_M(\cdot)$ which appear in (35) are as follows:

$$a_M(\mathbf{v}^{\square}, \boldsymbol{\phi}) := \left(\rho\left(\mathrm{d}_\tau \mathbf{v}^{\square} + (\mathrm{grad}\ \mathbf{v}^{\star}_M)\mathbf{v}^{\circ}_M\right),\ \boldsymbol{\phi}\right)_{\Omega^{\#}} + \left(\eta\epsilon(\mathbf{v}^{\square}), \epsilon(\boldsymbol{\phi})\right)_{\Omega^{\#}}, \tag{36}$$

$$b(\mathbf{v}^{\square}, q) := \left(\nabla \cdot \mathbf{v}^{\square}, q\right)_{\Omega^{\#}}, \tag{37}$$

$$g_{\mathbf{v}M}(\boldsymbol{\phi}) := (\mathbf{g} \cdot \boldsymbol{\phi})_{\Omega^{\#}} + (\boldsymbol{\tau}^* \cdot \boldsymbol{\phi})_{\Gamma^{\#}_N} - \mu_s \left(2\epsilon\left(\sum_{i=1}^{k} \alpha_i \mathbf{u}^{n,i} \right) - (\mathrm{grad}\ \mathbf{u}^{\#}_s)^T \mathrm{grad}\ \mathbf{u}^{\#}_s, \boldsymbol{\phi} \right)_{\Omega^{\#}_s}, \tag{38}$$

where $\mathbf{v}^{\circ}_M$, $\mathbf{v}^{\star}_M$ are prescribed below. Let us note in passing that $\mathrm{d}_{\tau\mathbf{v}^{\square}}$ involves velocities $\mathbf{v}^{\square}, \mathbf{v}^{n-1}, \dots, \mathbf{v}^{n-k}$.

- If $M = \mathcal{E}$, we define $\mathbf{v}_M^\circ$ using explicit extrapolation, as in Reference 55: depending on the order of the scheme k,

$$\begin{aligned}
\mathbf{v}_M^\circ &:= \mathbf{v}^{n,1} - \mathbf{w}^{\#} \qquad \text{for } k = 1, \\
\mathbf{v}_M^\circ &:= 2(\mathbf{v}^{n,1} - \mathbf{w}^{\#}) - (\mathbf{v}^{n,2} - \mathbf{w}^{n,1}) \qquad \text{for } k = 2.
\end{aligned} \tag{39}$$

  For the convective velocity $\mathbf{v}_M^\star$ we use an explicit velocity

$$\begin{aligned}
\mathbf{v}_M^\star &:= \mathbf{v}^{n,1} \qquad \text{for } k = 1, \\
\mathbf{v}_M^\star &:= 2\mathbf{v}^{n,1} - \mathbf{v}^{n,2} \qquad \text{for } k = 2,
\end{aligned} \tag{40}$$

  which makes the bilinear part of the form $a_M(\cdot, \cdot)$ symmetric.
- For $M = \mathcal{I}$, we make the step a bit more implicit and set

$$\mathbf{v}_M^\circ := \mathbf{v}^{\#} - \mathbf{w}^{\#}, \tag{41}$$

  and turn to the semi-implicit advection,

$$\mathbf{v}_M^\star := \mathbf{v}^{\square}, \tag{42}$$

  to ensure a better stability of the scheme. For further discussion we refer to Reference 56 or 57. Since the velocity is updated after the corrector step, we use the (cheaper) explicit advection in the predictor. This provides a good balance between accuracy and efficiency.

The functional $g_p(\cdot)$ will be introduced in the following section, see (47).

3.1.2 | Volume-preserving correction

In our case, $\rho_s = \text{const}$ and mass conservation is equivalent to incompressibility, that is,

$$\operatorname{div}(\mathbf{v}_s) = -\frac{1}{\rho_s}\frac{\partial \rho_s}{\partial t} = 0. \tag{43}$$

If, for whatever reason, the solid density is perturbed (e.g., by a numerical approximation of the zero-divergence constraint), such perturbation is accumulated and maintained through time evolution, resulting in a solid volume that may change as a result of these errors.

We provide a dynamic and strongly consistent correction to the volume of the solid by introducing an additional term in (43) with the aim of restoring $\hat{J} = 1$ whenever the scheme moves away from it:

$$\frac{\partial \rho_s}{\partial t} = J\frac{\rho_s}{\eta_V}\left(\frac{\rho_s}{\rho_{s\,0}} - 1\right) = J\frac{\rho_s}{\eta_V}(J - 1), \tag{44}$$

so that the solution would approach the density $\rho_{s\,0}$ regardless of its starting point. The damping parameter η_V can be interpreted as a volumetric viscosity which controls the dynamic rate of density correction and leads to a modified volume constraint for the solid with volume-preserving correction,

$$\operatorname{div}(\mathbf{v}_s) = -\frac{J}{\eta_V}(J - 1). \tag{45}$$

Accordingly, we reformulate the weak form $(21)_2$ of the volume constraint in the solid by adding the respective right-hand side, expressed in the reference configuration as

$$\int_{\Omega_s} q(\operatorname{div}\mathbf{v})\mathrm{d}\mathbf{x} = -\int_{\hat{\Omega}_s} \frac{1}{\eta_V}(\det(\hat{\mathbf{F}}) - 1)\hat{q}\ \mathrm{d}\hat{\mathbf{x}} \quad \forall q \in L^2(\Omega). \tag{46}$$

In the actual time-discrete scheme outlined in Algorithm 1, the volume-preserving correction is treated in an explicit manner, so that the right-hand side in the weak form (35) is defined as

$$g_p(q) := \left(-\frac{1}{\eta_V}(\hat{J}^\# - 1), \hat{q}\right)_{\hat{\Omega}_s}, \tag{47}$$

where $\hat{J}^\# = \det(\hat{\mathbf{F}}^\#)$, and $\hat{\mathbf{F}}^\# = \mathbf{I} + \mathrm{Grad}\ \hat{\mathbf{u}}_s^\#$ is the most recent algorithmic approximation of the deformation gradient $\hat{\mathbf{F}}$.

3.2 | Spatial discretization

Let us now introduce the fully discrete approximation of the fluid–structure interaction problem. We consider triangulation $\mathcal{T}$ of domain $\hat{\Omega}$ with characteristic element size of h. In our case, triangulation $\mathcal{T}$ consists of quadrilateral (2D) or hexahedral (3D) elements. We consider a regular matching grid, that is, assume that the initial fluid–solid interface does not intersect with any element. With sets of polynomials $\mathcal{P}_p(K)$ of order p on each element K, we define the finite element spaces on triangulation $\mathcal{T}$

$$\begin{aligned} \hat{\mathbb{V}}_h &= \{v \in H^1(\hat{\Omega}) \ : \ v|_K \in \mathcal{P}_{p_1}(K) \quad \forall K \in \mathcal{T}\}^d, \\ \hat{\mathbb{Q}}_h &= \{q \in H^1(\hat{\Omega}) \ : \ q|_K \in \mathcal{P}_{p_2}(K) \quad \forall K \in \mathcal{T}\}. \end{aligned} \tag{48}$$

where p_1 and p_2 are the orders of finite elements for the velocity and pressure, respectively. We first discretize the solid displacement $\hat{\mathbf{u}}_s \in \hat{\mathbb{V}}_h$ and the pseudo-displacement $\hat{\mathbf{u}} \in \hat{\mathbb{V}}_h$ that defines discrete mapping $\mathbf{A}^\#$. We then define the triangulation of $\Omega^\#$ as a transformed triangulation $\mathcal{T}$ by mapping $\mathbf{A}^\#$. The finite element spaces on domain $\Omega^\#$ are defined as:

$$\begin{aligned} \mathbb{V}_h &= \{v \circ (\mathbf{A}^\#)^{-1} \ : \ v \in \hat{\mathbb{V}}_h\}^d, \\ \mathbb{Q}_h &= \{q \circ (\mathbf{A}^\#)^{-1} \ : \ q \in \hat{\mathbb{Q}}_h\}. \end{aligned} \tag{49}$$

We assume that $\mathbb{V}_h$ and $\mathbb{Q}_h$ satisfy the Ladyzhenskaya–Babuska–Brezzi condition.[58]

3.2.1 | Streamline stabilization

Problem (36) is of convection–diffusion type, so for flows with high Reynolds numbers the convection becomes dominating. Since in such cases a straightforward finite element discretization typically results in oscillatory solutions, some additional stabilization to the original form $a_j(\cdot, \cdot)$ is necessary. While that might not be an issue on the finest grid, it clearly occurs on the coarser ones, which turns out to spoil the performance of the preconditioner. Hence, stabilization is crucial for the linear solver described in Section 4. In our scheme, we simply used

$$a_j^{\mathrm{stab}}(\mathbf{v}, \boldsymbol{\phi}) = a_j(\mathbf{v}, \boldsymbol{\phi}) + \sum_{K \in \mathcal{T}} \int_K r\ (v^\circ \cdot \mathrm{grad}\ v)\ (v^\circ \cdot \mathrm{grad}\ \boldsymbol{\phi})\ \mathrm{d}\mathbf{x}; \tag{50}$$

another possibility would be to use, for example, the streamline-upwind Petrov–Galerkin (SUPG) scheme.[59] The stabilization parameter inside cell K is[59,60]

$$r = \frac{h_K}{2\|v^\circ\| p_1} \frac{\coth(\mathrm{Pe}_K) - 1}{\mathrm{Pe}_K},$$

where h_K is the diameter of cell $K \in \mathcal{T}$ and the Peclet number Pe_K computed with respect to the cell size is

$$\mathrm{Pe}_K = \|v^\circ\| \frac{h_K}{2\mu\ p_1}.$$

In cells where $\mathrm{Pe}_K < 1$ we set $r = 0$ to avoid problems with floating-point arithmetics. Note that for a sufficiently fine mesh $\mathrm{Pe}_K < 1$ and thus the stabilization term vanishes, therefore it does not affect the solution. However, we intend to use a multigrid preconditioner to solve the linear system, thus we also need non-oscillatory solution regardless how large the element size is.

4 | A MULTILEVEL, MATRIX-FREE PRECONDITIONER FOR THE LINEAR SYSTEM

On every time step, there are two computationally intensive parts inside the "**for**" loop in Algorithm 1: the computation of the extension of the velocity, $\mathrm{Ext}(\hat{\mathbf{v}}^{\square})$ on the fluid domain, and the solution of the discretized system (35) in order to determine the velocity and the pressure on the entire domain. To perform the former, we use a geometric multigrid preconditioned CG, which in our experiments worked just fine. The latter system is much more challenging and therefore we will focus on this subproblem in the present section.

The linear system (35) has a block structure of a generalized Oseen-type saddle point problem

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ p \end{bmatrix} = \begin{bmatrix} g_{\mathbf{v}} \\ g_p \end{bmatrix}, \tag{51}$$

where the square matrix A corresponds to a discretized convection–diffusion–reaction operator (50) with the viscosity coefficient which is discontinuous across the fluid–solid interface, while B corresponds to the discrete divergence operator. Since the number of unknowns in (51) is very large, compare Table 2, direct solution of this system is infeasible. On the other hand, the system is also challenging for iterative solvers: it is ill-conditioned due to both the fine mesh size and the high-contrast in the effective viscosities between the fluid and solid domains (η_f vs. $\gamma\tau\mu_s$), so it requires an efficient preconditioner, see Reference 23. Since the fluid and the solid typically have very different properties, the viscosity contrast plays a substantial role even when the time step τ is relatively small.

Many existing preconditioners for FSI problems in the monolithic formulation exploit the block structure of the problem.[35,36] Others may be based on the multigrid, compare References 21,23,61, and 62 or the domain decomposition method.[5,63] See References 64 and 8 and references therein for a broad survey of classical and recent developments in this field.

We base our implementation on the `deal.II` library,[65] taking advantage of the specific structure of the linear problem, and leveraging the properties of modern computer hardware, such as the availability of vectorized SIMD instructions and parallelism—by choosing to implement the preconditioner (and the solver) using the matrix-free approach,[42] which is very well supported in the `deal.II` library. The CPU cache efficiency is improved because the data is accessed in a more localized manner, reducing the number of cache misses and increasing the overall performance of the solver.[66] Furthermore, the solver's memory footprint is kept low, which means that it can deal with larger problems or function on computers with limited memory resources.

To this end, we adapt the multilevel preconditioner developed for a generalized stationary Stokes problem with discontinuous viscosity coefficient, proposed and analyzed in References 37 and 40. This preconditioner is not only robust with respect to the mesh size and the viscosity contrast, but also supports the matrix-free paradigm by design. Theoretical foundations for a very similar preconditioner have recently been laid in Reference 41. While matrix-free preconditioners have successfully been applied in various contexts, for example, phase-field fracture problems[67] or mantle convection simulations,[68] its application within FSI solver frameworks is much less common, and state-of-the-art FSI solvers are often matrix-based.[69] See also Reference 70 for a recent application of matrix-free methods in the case of finite-strain solids.

If the convection is treated explicitly, (51) already defines a generalized Stokes system, so the preconditioner of Reference 40 can be applied directly. In the case of implicit convection, however, the matrix A becomes nonsymmetric, so the preconditioner needs some adjustments. For the convenience of the reader, below we briefly recall the main ingredients of the preconditioner, already adapted to the nonsymmetric case.

Since the preconditioner is of multilevel type, we will assume that the underlying grid $\mathcal{T}$ is a result of a J-level uniform refinement of some coarse grid $\mathcal{T}_0$ aligned with the fluid–solid interface, resulting in a family of nested conforming triangulations of $\hat{\Omega}$:

$$\mathcal{T}_0 \subset \mathcal{T}_1 \subset \cdots \subset \mathcal{T}_J = \mathcal{T}. \tag{52}$$

These, in turn, generate a family of discrete problems defined on mesh $\mathcal{T}_j$, with corresponding block matrices

$$\mathcal{M}_j = \begin{bmatrix} A_j & B_j^T \\ B_j & 0 \end{bmatrix}. \tag{53}$$

Algorithm 2. One iteration of a multigrid V-cycle procedure for a system $\mathcal{M}_j y = F_j$ with initial guess x

```
Function $y = \mathrm{MGM}(\mathcal{M}_j, F_j, \mathcal{K}_j, m, x, j)$
    if $j = 0$ then
        Solve $\mathcal{M}_0 y = F_0$;                              ◁ Direct solve on the coarsest grid $\mathcal{T}_0$
        return $y$;
    end
    $x^0 = x$;
    for $s = 1$ to $m$ do
        $x^s = x^{s-1} + \mathcal{K}_j(F_j - \mathcal{M}_j x^{s-1})$;                          ◁ pre-smoothing
    end
    $r_{j-1} = R_j \left(F_j - \mathcal{M}_j x^m\right)$;                   ◁ restriction to the coarser grid
    $e_{j-1} = \mathrm{MGM}(\mathcal{M}_{j-1}, r_{j-1}, \mathcal{K}_{j-1}, m, 0, j-1)$;   ◁ coarse correction; recursive call
    $e_j = R^T_{j-1} e_H$;                                   ◁ prolongation from the coarser grid
    $y^0 = x^m + e_j$;
    for $s = 1$ to $m$ do
        $y^s = y^{s-1} + \mathcal{K}_j(F_j - \mathcal{M}_j y^{s-1})$;                         ◁ post-smoothing
    end
    $y = y^m$;
    return $y$
end
```

The preconditioner is formulated as n iterations of the V-cycle multigrid for $\mathcal{M}_J$ (see Algorithm 2), with m smoothing steps which use a customized block smoother on the jth level

$$\mathcal{K}_j = \begin{bmatrix} \hat{A}_j & B_j^T \\ B_j & B_j \hat{A}_j^{-1} B_j^T - \hat{S}_j \end{bmatrix}^{-1}. \tag{54}$$

In order to apply $\mathcal{K}_j$ to a vector, two solves with $\hat{A}_j$ and one with $\hat{S}_j$ are required. Both are implemented as matrix-free operators as follows:

$$\hat{A}^{-1} = \mathrm{Cheb}(A, \mathrm{diag}(A), k_A), \qquad \hat{S}^{-1} = \begin{cases} \mathrm{ChebMINRES}(n_S, k_S) & \text{when } A = A^T, \\ \mathrm{ChebBiCGStab}(n_S, k_S) & \text{otherwise,} \end{cases}$$

making it possible to formulate a multiplication of $\mathcal{K}_j$ by a vector as a matrix-free operation. Here, $\mathrm{Cheb}(M, D, k)$ denotes k iterations of the Chebyshev smoother[71] for M preconditioned with D,

$$\mathrm{Cheb}(M, D, k) = P_k(D^{-1}M)D^{-1}, \tag{55}$$

where P_k is a linearly transformed kth degree Chebyshev polynomial of the first kind which minimizes the supremum norm over the interval $[\frac{2}{25}\lambda_M, \frac{30}{25}\lambda_M]$, and λ_M is the largest eigenvalue of $D^{-1}M$; see Reference 40 for details.

By $\mathrm{ChebMINRES}(n_S, k_S)$ we denote the result of n_S iterations of the MINRES method applied to the system with matrix $S = B\hat{A}^{-1}B^T$, preconditioned with $\mathrm{Cheb}(S, \mathrm{diag}\big(B(\mathrm{diag}A)^{-1}B^T\big), k_S)$. This form of $\hat{S}^{-1}$ is thus used when solving (35) if the time-stepping procedure is in "$\mathcal{E}$" operational mode. The only difference between ChebMINRES and ChebBiCGStab is that in the MINRES iteration in the former is replaced with the BiCGStab method in the latter, which is suitable in "$\mathcal{I}$" working mode of the scheme. All these iterative methods take the zero vector as the initial guess.

While our solver employs efficient matrix-free operations for smoothing and solution transfers, the use of a direct solver as a coarse grid correction is not well-suited for parallel environments. Although direct solvers offer high accuracy, they often exhibit limited scalability in distributed computing settings, creating computational bottlenecks for large-scale parallel simulations. This impact on performance is particularly pronounced when the number of refinement levels is relatively low, resulting in a higher ratio between the number of coarse degrees of freedom and the total number of degrees of freedom.

5 | NUMERICAL RESULTS

In this section, we present the results of numerical experiments conducted to evaluate the performance and efficiency of the method on a set of benchmark problems. As mentioned in Section 4, the experimental framework was implemented using the `deal.II` library[65] and its matrix-free framework.[42] The stable Taylor–Hood finite element pair, which corresponds to choosing $p_1 = 2$ and $p_2 = 1$ in (48), was consistently used in all tests.

5.1 | Test problems

We evaluate the solver on test problems very closely resembling the well-known Turek–Hron two-dimensional benchmarks FSI2 and FSI3.[43] These benchmarks consist of an elastic beam attached to a cylindrical obstacle inside a channel, which interacts with the flow. To test the solver in 3D we use the benchmark problem FSI3D from Reference 11, which consists of an elastic plate attached to a cylinder (depicted in Figure 2). This geometry is obtained by extruding the geometry of 2D tests into the third dimension; the detailed geometrical settings are presented in Figure 2 together with dimensional parameters in Table 1.

Our experimental framework differs from the original benchmarks mentioned above in two ways. The most important departure is due to the choice of the model of the solid (see Section 2.3.2) which, in contrast to Turek and Hron setting, is assumed incompressible. To mark that our test problems involve an incompressible solid, we will refer to them as FSI2i, FSI3i, and FSI3Di, respectively. For this reason, our goal will not be to replicate the exact results of the original benchmarks, but rather to demonstrate the performance of our method in simulation of incompressible FSI problems. (To the best of our knowledge, there are no available results for Turek–Hron benchmark problems using an incompressible solid.)

Second, in contrast to the approach used in Reference 43, where the flow gradually accelerates, our simulation starts with an instantaneous acceleration of the flow at the initial time. This modification allows us to evaluate the solver's capabilities in handling a fully developed flow in a nearly undeformed configuration, and provides a fixed geometry for solver performance comparison—particularly when considering various time step sizes. However, it induces artificial pressure jumps during the first few time steps. From our observations, it can be seen that these pressure oscillations are damped within a few subsequent initial time steps and have a negligible impact on the final results.

Apart from that, all other settings are identical as in the benchmarks mentioned above.

In order to improve mass conservation and incompressibility of the solid, unless stated otherwise, we will apply the volume-preserving correction, see Section 3.1, with default parameter $\eta_V = 0.1$.

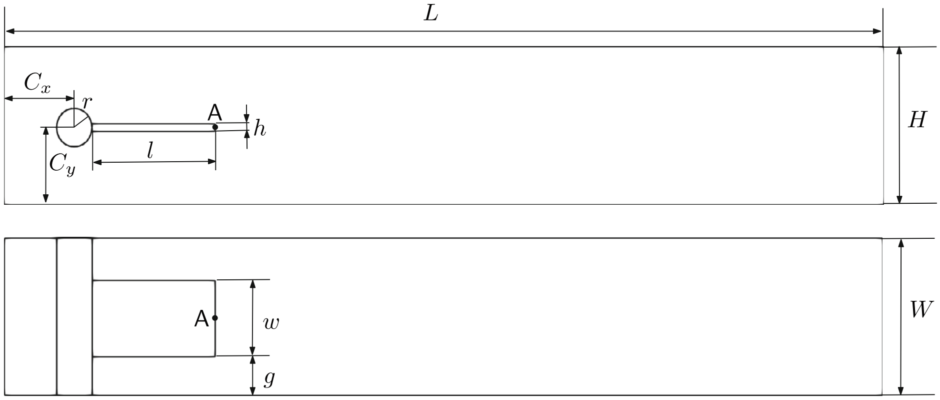

FIGURE 2 Geometry for test problems: $x - y$ view (top), used for experiments in two dimensions (FSI2i, FSI3i), and as a base for the extrusion in three dimensions (FSI3Di), see the $x - z$ view (bottom). The plotting point A is shown in both views.

TABLE 1 Geometry of test problems—List of dimensional parameters (all expressed in meters).

L	H	W	l	h	w	C_x	C_y	r	g
2.5	0.41	0.41	0.35	0.02	0.2	0.2	0.2	0.05	0.1

5.2 | Mesh and deformation handling

To discretize the reference spatial domain $\hat{\Omega}$, we start with a coarse grid $\mathcal{T}_0$ as illustrated in Figure 3, which then undergoes J rounds of uniform refinement, including adjustments to accommodate the curved boundary of the cylinder. This results in a family of refined grids $\mathcal{T}_0, \dots, \mathcal{T}_J$. The numbers of degrees of freedom corresponding to selected unknowns for specific J used in the experiments are summarized in Table 2. For brevity, in what follows we denote by N the total number of unknowns in the momentum equation.

To handle the mesh deformation, we employ a linear elasticity problem with a variable coefficient distribution to enhance the quality of the resulting computational grid. The coefficient distribution is as follows:

$$\mu_{\mathsf{A}}(\hat{x}, \hat{y}, \hat{z}) = 1 + 50 \exp(-800((\hat{x} - \mathsf{A}_{\hat{x}})^2 + (\hat{y} - \mathsf{A}_{\hat{y}})^2)), \tag{56}$$

where $\mathsf{A}_{\hat{x}}$ and $\mathsf{A}_{\hat{y}}$ are the coordinates of the point A, shown in Figure 2.

5.3 | Linear solver settings

The solution to (35) was obtained using the FGMRES iterative solver, preconditioned with the method outlined in Section 4, whose parameters are specified in Table 3. The iteration was terminated when the Euclidean norm of the residual dropped below the threshold value $\varepsilon = 10^{-6}$ in FSI2i and FSI3i testcases or $\varepsilon = 10^{-3}$ in FSI3Di. The initial guess for the FGMRES iteration was set equal to the most recent values of the velocity and pressure.

The extension (8) was computed by the CG iteration preconditioned with a classical matrix-free multigrid method with fourth order Chebyshev smoother based on the diagonal, compare Reference 42. There, the stopping criterion was always to reduce the Euclidean norm of the residual below 10^{-6}.

5.4 | Tests in 2D

In both FSI2i and FSI3i testcases, the cylinder, upper, and lower sides of the channel are considered as rigid walls and no-slip boundary conditions are imposed there. At the right end of the channel, a do-nothing boundary condition is

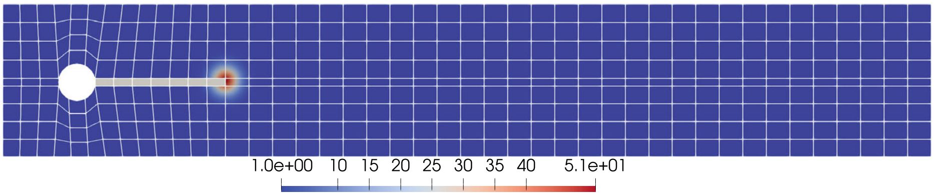

FIGURE 3 Coarse grid and distribution of the mesh stiffness parameter, see (56). Solid marked in gray.

TABLE 2 Number of degrees of freedom corresponding to grids in 2D or 3D obtained after J refinements of the coarse meshes (N = velocity + pressure).

	J	1	2	3	4	5	6	7	8
2D case	Velocity	12k	50K	198k	790k	3.15M	12.6M	50.4M	201M
	Pressure	1.6k	6k	25k	99k	395k	1.58M	6.30M	25.2M
	N	14k	56k	223k	889k	3.55M	14.2M	56.7M	227M
	Displacement	12k	50K	198k	790k	3.15M	12.6M	50.4M	201M
3D case	Velocity	533k	4.11M	32.2M	255M	–	–	–	–
	Pressure	24k	178k	1.37M	10.7M	–	–	–	–
	N	557k	4.27M	33.6M	266M	–	–	–	–
	Displacement	533k	4.11M	32.2M	255M	–	–	–	–

TABLE 3 Default parameters of the preconditioner described in Section 4.

Parameter		$\mathcal{E}$	$\mathcal{I}$
Order of the Chebyshev smoother defining $\hat{A}^{-1}$	k_A	4	6
Number m of outer MG smoothing steps	m	2	2
Number of outer MG iterations	n	1	1
Order of the Chebyshev smoother defining $\tilde{S}^{-1}$	k_S	1	1
Number of inner MINRES/BiCGStab iterations	n_S	1	1

TABLE 4 Material parameters.

Parameter	Units	FSI2i	FSI3i	FSI3Di
ρ_s	$\mathrm{kg\,m^{-3}}$	10^4	10^3	10^3
ρ_f	$\mathrm{kg\,m^{-3}}$	10^3	10^3	10^3
μ_s	$\mathrm{kg\,m^{-1}\,s^{-2}}$	0.5×10^6	2×10^6	2×10^6
η_f	$\mathrm{kg\,m^{-1}\,s^{-2}}$	1	1	1
V_{in}	$\mathrm{m\,s^{-1}}$	1	2	1.75
Re	–	100	200	175

assumed, while on the left side of the channel a parabolic inflow velocity profile is prescribed, compare Reference 43. The flow is defined by the average velocity at the inlet, V_{in}, and by the Reynolds number (computed with respect to the obstacle diameter). Here, we consider $V_{\mathrm{in}} = 1$ or $V_{\mathrm{in}} = 2$, corresponding to Re $= 100$ and Re $= 200$, respectively. The detailed test data is presented in Table 4.

We conduct FSI2i and FSI3i tests using the GCsI$_2(\mathcal{E}, \mathcal{I})$ scheme on a fixed spatial grid after $J = 5$ levels of refinement (see Table 2 for details on the number of degrees of freedom). By varying time step size τ we evaluate the stability and convergence (in time) of our numerical method. Additionally, for qualitative comparison, we run the FSI2i benchmark using a one-stage method, GCsI$_2(\mathcal{I})$. Let us note that the standard GCE scheme (which, in our notation, corresponds to the GCsI$_2(\mathcal{E})$ method) turned out unstable for time steps larger than $\tau = 0.00025$ (and thus for all time steps considered below), so we excluded this method from the following analysis.

In Tables 5 and 6 we report the amplitude, average displacement, and period of the last computed oscillation at point A, see Figure 2. Apparently, with τ decreasing, both GCsI$_2(\mathcal{E}, \mathcal{I})$ and GCsI$_2(\mathcal{I})$ schemes converge to similar results, with the former settling for larger time step values than the latter. Lower amplitude of oscillations of $u_y(\mathsf{A})$ in the case of GCsI$_2(\mathcal{I})$ suggests that the single-stage method introduces more artificial damping compared to the two-stage scheme.

As expected, the results obtained from FSI2i and FSI3i tests are close to FSI2 and FSI3 from Reference 43. However, they are not identical, since our solid model uses partially different constitutive laws.

The number of FGMRES iterations required to reduce the residual below prescribed threshold remains, as seen in Figure 4, roughly constant over time, with lower convergence rate observed during the initial stages of the simulation, which then improves in the steady oscillation phase. During this second phase, the solver performs no more than 15 iterations in $\mathcal{I}$ stage of the GCsI$_2(\mathcal{E}, \mathcal{I})$ scheme (and significantly less in $\mathcal{E}$ mode), confirming high efficiency of the preconditioner described in Section 4 even when the linear problem is nonsymmetric (see also Table 8).

5.5 | Tests in 3D

In the FSI3Di test case (illustrated in Figure 5), a non-slip condition is imposed on the long outer sides of the channel. On the right side of the domain a do-nothing outflow condition is prescribed, while the left side is subject to the prescribed inflow velocity profile Dirichlet boundary condition:

$$v_{\mathrm{in}}(x, y, z) = \frac{36V_{\mathrm{in}}}{H^2W^2} y(H - y)z(W - z), \tag{57}$$

TABLE 5 Comparison of displacements and frequency in FSI2i (incompressible material) test problem obtained with $\text{GCsI}_2(\mathcal{E},\mathcal{I})$ (top) and $\text{GCsI}_2(\mathcal{I})$ (bottom) schemes, for various time step sizes τ. Spatial mesh with $J = 5$ levels.

	τ	$u_x(A) \times 10^{-3}$	$u_y(A) \times 10^{-3}$	Frequency (u_y(A))
FSI2i, $\text{GCsI}_2(\mathcal{E},\mathcal{I})$	0.04	(No steady oscillations)		
	0.02	−14.83 ± 11.65	1.29 ± 76.38	1.96
	0.01	−15.09 ± 13.04	1.25 ± 80.26	2.00
	0.005	−14.85 ± 12.89	1.23 ± 80.77	2.00
	0.0025	−14.72 ± 12.55	1.23 ± 80.77	2.00
FSI2i, $\text{GCsI}_2(\mathcal{I})$	0.04	(No steady oscillations)		
	0.02	−13.87 ± 11.80	1.38 ± 76.45	1.92
	0.01	−14.26 ± 12.65	1.32 ± 79.42	1.96
	0.005	−14.53 ± 12.71	1.30 ± 80.51	1.98
	0.0025	−14.52 ± 12.52	1.23 ± 80.77	2.00
FSI2 (compressible)[43]	0.001	−14.58 ± 12.44	1.23 ± 80.6	2.0

Note: For qualitative comparison, in the last row we also report results of the FSI2 benchmark for a compressible material.

TABLE 6 Comparison of displacements and frequency in the FSI3i (incompressible) testcase with $J = 4$.

	τ	$u_x(A) \times 10^{-3}$	$u_y(A) \times 10^{-3}$	Frequency (u_y(A))
FSI3i, $\text{GCsI}_2(\mathcal{E},\mathcal{I})$	0.02	(No steady oscillations)		
	0.01	−2.85 ± 2.06	1.80 ± 32.19	5.38
	0.005	−2.79 ± 2.46	1.47 ± 34.38	5.47
	0.0025	−2.88 ± 2.69	1.45 ± 34.56	5.51
FSI3 (compressible)[43]	0.0005	−2.69 ± 2.53	1.48 ± 34.38	5.3

Note: For qualitative comparison, in the last row we also report results of the FSI3 benchmark for a compressible material.

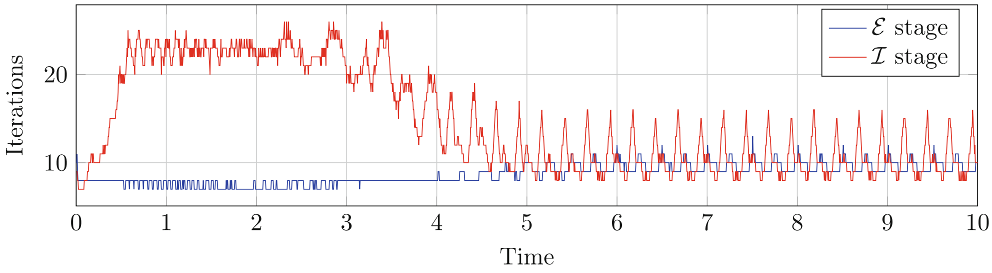

FIGURE 4 Number of FGMRES iterations per time step in 2D case (FSI2i), as a function of time. Results for $J = 5$ and $\tau = 0.005$.

which results in Re = 175 for $V_{\text{in}} = 1.75$ (calculated with respect to the cylinder diameter). For the handling of grid deformations, we use the linear elasticity problem with the same coefficient distribution as in 2D case, compare (56).

To generate the computational grids, we first extrude the 2D grid used in our earlier experiments along the third dimension, resulting in six elements in that direction. Then, the grid is uniformly refined J times to obtain a multilevel structure. We conduct the experiments dealing with four different levels of refinements, $J = 1, \dots, 4$ (see Table 2 for details on the number of degrees of freedom) and several time step sizes.

To assess the quality of the solution, we again record the oscillation frequency, the average, and the amplitude of the displacement of point A, see Table 7. The results indicate that, as the time step size decreases, the oscillation frequency, average and amplitude of displacement of point A, each settle at a certain value, supporting the expectation that the scheme is convergent as $\tau \to 0$.

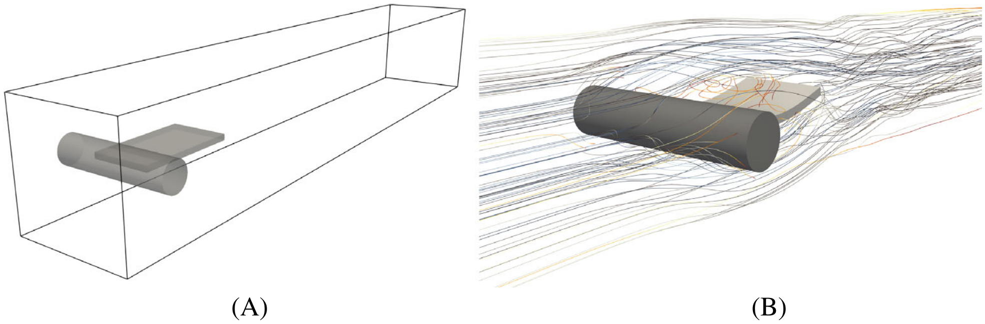

FIGURE 5 (A) Visualization of the geometry of the FSI3Di testcase. (B) Plot of the streamlines and of the deformed plate.

TABLE 7 Comparison of displacements and frequency in FSI3Di test with $J = 3$.

	τ	u_x(A) × 10^{-3}	u_y(A) × 10^{-3}	Frequency u_y(A))
FSI3i, GCsI$_2(\mathcal{E}, \mathcal{I})$	0.008	−2.39 ± 2.23	2.51 ± 28.47	5.63
	0.004	−2.39 ± 2.44	2.45 ± 29.20	5.73
	0.002	−2.39 ± 2.49	2.43 ± 29.24	5.77
lFSI3D (compressible)[11]	0.001	−2.143 ± 2.383	2.699 ± 25.594	5.60

Note: For qualitative comparison, in the last row we also report results of the FSI3D benchmark for a compressible material.

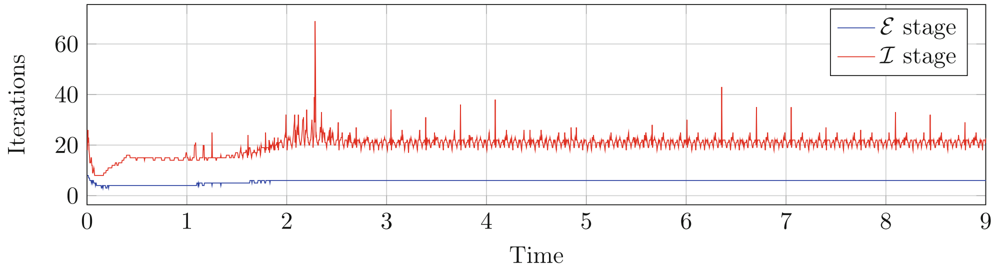

FIGURE 6 Number of FGMRES iterations per time step in 3D case (FSI3Di), as a function of time.

As in the 2D case, the results of the FSI3Di test are again slightly different from FSI3D,[11] due to different material laws for the solid.

From Figure 6 it follows that, after a short transition period, the number of FGMRES iterations required to converge does not exceed 6 during the $\mathcal{I}$ stage of the GCsI$_2(\mathcal{E}, \mathcal{I})$ scheme. During the $\mathcal{E}$ stage it remains below 25 for most of the steps; however, this number fluctuates and from time to time increases up to about 40.[*] This shows that the preconditioner works efficiently for the linear systems of equations being solved in each stage of the GCsI$_2(\mathcal{E}, \mathcal{I})$ scheme in 3D as well; see also Table 9.

5.6 | Influence of the volume-preserving correction on the quality of the solution

Here we investigate how effective is the volume-preserving correction, introduced in Section 3.1.2, in resolving certain issues related to the deformation of the beam. In Figure 7, we compare the results obtained for the FSI2i testcase with a relatively large time-step size, $\tau = 0.01$, when the volumetric damping is either weak, strong or absent. It is evident that without the volumetric correction ($\eta_V = \infty$) the maximum amount of horizontal displacement, $|u_x|$, at point A tends to slowly increase over time (so that the edge of the beam drifts away from A), whereas it remains stable for both $\eta_V = 0.1$ and $\eta_V = 0.02$. Moreover, the displacement graphs obtained for the latter values of η_V practically overlap.

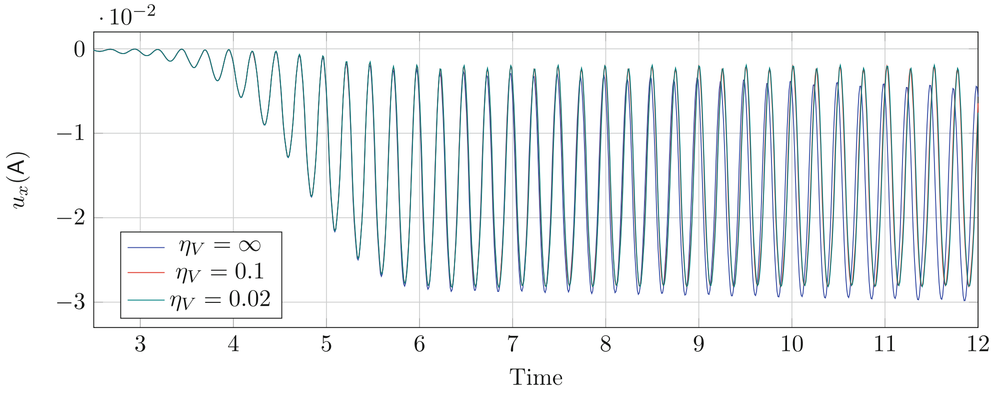

FIGURE 7 Influence of volume-preserving correction on the displacement of the point A versus time, for varying damping parameter η_V of the volume-preserving correction. (By convention, $\eta_V = \infty$ corresponds to no correction.) FSI2i test case, grid with $J = 5$ levels, $\mathrm{GCsI}_2(\mathcal{E}, \mathcal{I})$ scheme with timestep $\tau = 0.01$.

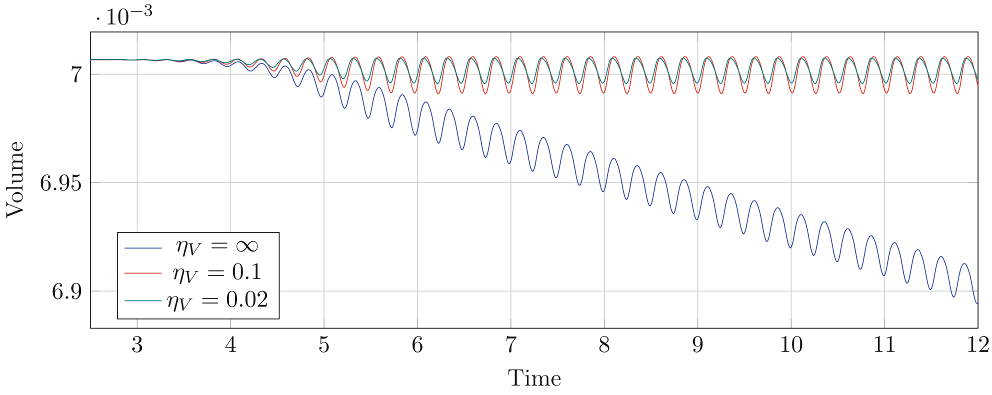

FIGURE 8 Volume of the solid versus time, for varying damping parameter η_V of the volume-preserving correction. (By convention, $\eta_V = \infty$ corresponds to no correction.) FSI2i test case, grid with $J = 5$ levels; $\mathrm{GCsI}_2(\mathcal{E}, \mathcal{I})$ scheme with time step $\tau = 0.01$.

The importance of the volume-preserving correction is even more pronounced when one compares the evolution of the volume of the solid in time, see Figure 8. Without stabilization, its volume is diminishing (a similar effect also occurs in the case of the first-order time integration scheme $\mathrm{GCsI}_1(\mathcal{E}, \mathcal{I})$ where the beam is gaining volume). The presence of the volume-preserving correction term resolves the problem with the incompressibility condition, thus paving the way to use large time steps in our scheme. Let us mention that, to some extent, the final result seems quite insensitive to the (small) value of the damping parameter η_V.

5.7 | Performance

In order to evaluate the performance of our solver, we plotted the time spent per degree of freedom versus the number of degrees of freedom for both the 2D and 3D cases, as seen in Figure 9. The tests were performed on four nodes, each equipped with two Intel Xeon 8160 @2.1GHz processors. The results demonstrate that our method is highly efficient, with the time spent per degree of freedom decreasing as the number of degrees of freedom increases. This trend can be attributed to the relatively inefficient coarse grid solver becoming a smaller fraction of the overall timing as the problem size increases. We also note that the number of iterations of the FGMRES linear solver required for convergence is roughly independent of the problem size, as demonstrated in Tables 8 and 9.

We compared its efficiency with results reported in the literature,[11(tab. 9)] as shown in Table 10. Notwithstanding the fact that Reference 11 used a different discretization scheme and compressible constitutive equations, the comparison clearly shows that matrix-free algorithms can outperform matrix-based ones in terms of throughput. This is especially true in 3D simulations.

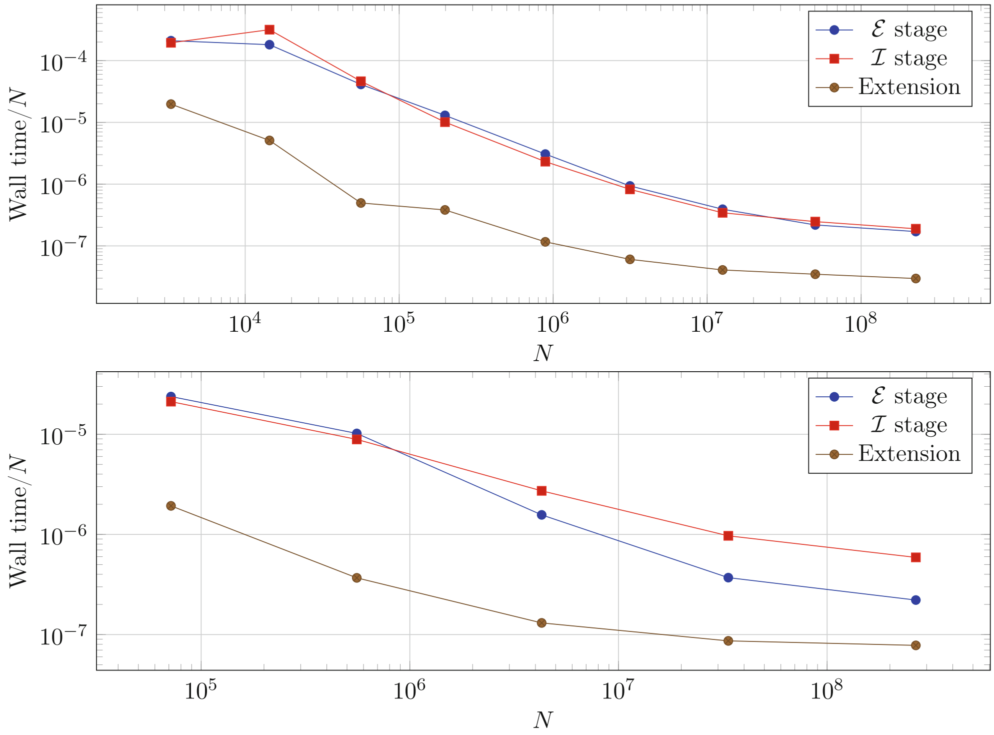

FIGURE 9 Efficiency of the solver, as a function of the total number N of degrees of freedom in the momentum equation. FSI2i problem (top) and FSI3Di (bottom).

TABLE 8 Number of FGMRES iterations required for convergence ($\varepsilon = 10^{-6}$) at the first time step $\tau = 0.004$ in 2D tests during both stages $\{\mathcal{E}, \mathcal{I}\}$ of the GCsI$_2(\mathcal{E}, \mathcal{I})$ scheme and for varying number of levels J.

	Stage ↓	***J* →**	**3**	**4**	**5**	**6**	**7**	**8**
FSI2i	$\mathcal{E}$		8	9	9	10	10	10
	$\mathcal{I}$		15	13	11	10	9	9
FSI3i	$\mathcal{E}$		11	11	12	12	12	12
	$\mathcal{I}$		26	28	23	20	19	18

TABLE 9 Number of FGMRES iterations required for convergence ($\varepsilon = 10^{-3}$) at the first time step $\tau = 0.005$ in the 3D test during both stages $\{\mathcal{E}, \mathcal{I}\}$ of the GCsI$_2(\mathcal{E}, \mathcal{I})$ scheme and for varying number of levels J.

	Stage ↓	***J* →**	**1**	**2**	**3**	**4**
FSI3Di	$\mathcal{E}$		9	10	8	8
	$\mathcal{I}$		9	15	18	15

Notably, as the problem size increases, our solver's efficiency improves significantly, making it well-suited for large-scale simulations. When solving problems with lower numbers of degrees of freedom, the impact of a suboptimal coarse solver degrades significantly the performances, which are quickly recovered when the problem size increases, and the relative cost of the coarse solver becomes more and more negligible.

The comparison highlights the advantage of the matrix-free implementation in terms of computational speed and scalability, reinforcing its potential for addressing complex FSI problems efficiently.

TABLE 10 Comparison of diverse strategies from literature, reported in Reference 11 (tab. 9).

	Approach	DoFs	τ	Cores	$\frac{\text{DoFs}}{\text{Second}}$	$\frac{\text{DoFs}}{\text{Second} \times \text{Core}}$	Source
2D	GMRES	16M	0.01	64	65,040 s^{-1}	1016 s^{-1}	72
	Red. Newton-MG	1.28M	0.002	16	27,943 s^{-1}	436 s^{-1}	11
	$\text{GCsI}_2(\mathcal{E}, \mathcal{I})$	1.68M	0.005	192	357,978 s^{-1}	1864 s^{-1}	Figure 9
	$\text{GCsI}_2(\mathcal{E}, \mathcal{I})$	428M	0.005	192	6,259,254 s^{-1}	32,600 s^{-1}	Figure 9
3D	GMRES	14M	0.001	256	1223 s^{-1}	4 s^{-1}	72
	Red. Newton-MG	3.53M	0.002	32	14,413 s^{-1}	450 s^{-1}	11
	$\text{GCsI}_2(\mathcal{E}, \mathcal{I})$	8.38M	0.004	192	439,318 s^{-1}	2288 s^{-1}	Figure 9
	$\text{GCsI}_2(\mathcal{E}, \mathcal{I})$	521M	0.004	192	2,067,736 s^{-1}	10,769 s^{-1}	Figure 9

Note: We highlight the impact of the degrees of freedom, time-step size, and the number of cores employed on the "degrees of freedom per second computed for each time step," and the corresponding value normalized with respect to the number of cores. Higher ratios signify improved performance.

6 | CONCLUSIONS

In this article, we presented a numerical method for solving FSI problems with an incompressible, hyperelastic Mooney–Rivlin solid. For a fully-coupled finite element discretization, we have designed a semi-implicit kth order BDF-based time integration scheme, $\text{GCsI}_k(\cdot)$, augmented with a correction term aimed at improving the volume preservation. The method uses the ALE formulation to track the moving parts.

The scheme, which resembles an S-stage predictor–corrector method, possesses improved stability properties, as compared to the GCE scheme introduced in Reference 5. This makes it possible to use much larger time steps while still capturing the dynamics of the interaction between the fluid and the solid. Although each time step is roughly S times more expensive than the corresponding step of the GCE method, numerical experiments demonstrate that already for $S = 2$, the $\text{GCsI}_2(\mathcal{E}, \mathcal{I})$ scheme is stable enough so that the gains in the efficiency, due to taking larger time steps, easily outweigh the increased cost of a single step.

The splitting between the explicit and implicit parts of the integration step has been designed in such a way that, in the latter, a solution to a generalized nonsymmetric Stokes-like problem needs to be computed. This is challenging, because the problem's condition number is adversely affected not only by the number of the unknowns, but also by a very large contrast in the coefficients of the underlying Stokes-like PDE. To address this issue, we adapted a preconditioning method from Reference 40 and proved its robustness and efficiency in this type of application. While the number of preconditioned FGMRES iterations fluctuated in time, it was always bounded by a reasonably moderate constant, regardless of the problem size.

The method described above has been implemented in a matrix-free fashion using the `deal.II` library and performed very well on classical FSI benchmarks in 2D and 3D, with as many as 250M spatial degrees of freedom, being solved in parallel on four computing nodes, each equipped with two Intel Xeon 8160 @2.1GHz CPU processors. The throughput of the method was compared to the throughput of other methods found in the literature,[11] and the results are summarized in Table 10, clearly showing that our solver has a significantly higher throughput than matrix-based methods, thanks to the efficiency of our preconditioner, and to the matrix-free implementation.

We believe that this approach has the potential for further improvement—that we plan to investigate in future research—for example through the introduction of a more efficient coarse solver, which is one of the bottlenecks in the current implementation, as shown by the efficiency results.

ACKNOWLEDGMENTS

Michał Wichrowski has been partially supported by the National Science Centre (NCN) in Poland through Grant No. 2015/19/N/ST8/03924. Stanisław Stupkiewicz has been partially supported by the EffectFact project (No. 101008140) funded within the H2020 Programme, MSC Action RISE-2022. Luca Heltai was partially supported by the grant MUR PRIN 2022 No. 2022WKWZA8 "Immersed methods for multiscale and multiphysics problems (IMMEDIATE)". Open Access funding enabled and organized by Projekt DEAL.

DATA AVAILABILITY STATEMENT

Data available on request from the authors.

ENDNOTE

[*] The unusual peak occurring at $t = 2.24$ is caused by restarting the simulation from a checkpoint—in that case, the initial guess for the FGMRES was not the previous solution but just the zero vector.

ORCID

Michał Wichrowski https://orcid.org/0000-0002-3644-3917
Stanisław Stupkiewicz https://orcid.org/0000-0002-4592-3576

REFERENCES

1. Taylor G. Analysis of the swimming of long and narrow animals. *Proc R Soc Lond A Math Phys Sci*. 1952;214(1117):158-183.
2. Fernández MÁ, Moubachir M. A Newton method using exact Jacobians for solving fluid-structure coupling. *Comput Struct*. 2005;83(2):127-142.
3. Badia S, Quaini A, Quarteroni A. Splitting methods based on algebraic factorization for fluid-structure interaction. *SIAM J Sci Comput*. 2008;30(4):1778-1805.
4. Degroote J, Bathe K, Vierendeels J. Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction. *Comput Struct*. 2009;87(11-12):793-801.
5. Crosetto P, Deparis S, Fourestey G, Quarteroni A. Parallel algorithms for fluid-structure interaction problems in haemodynamics. *SIAM J Sci Comput*. 2011;33(4):1598-1622.
6. Crosetto P, Reymond P, Deparis S, Kontaxakis D, Stergiopulos N, Quarteroni A. Fluid-structure interaction simulation of aortic blood flow. *Comput Fluids*. 2011;43(1):46-57.
7. Benra F-K, Josef Dohmen H, Pei J, Schuster S, Wan B. A comparison of one-way and two-way coupling methods for numerical analysis of fluid-structure interactions. *J Appl Math*. 2011;2011:853560.
8. Bucelli M, Dede L, Quarteroni A, Vergara C. Partitioned and monolithic algorithms for the numerical solution of cardiac fluid-structure interaction. *Commun Comput Phys*. 2022;32(5):1217-1256.
9. van Brummelen EH. Partitioned iterative solution methods for fluid-structure interaction. *Int J Numer Methods Fluids*. 2011;65(1-3):3-27.
10. Clevenger TC, Heister T. Comparison between algebraic and matrix-free geometric multigrid for a stokes problem on adaptive meshes with variable viscosity. *Numer Linear Algebra Appl*. 2021;28(5):e2375.
11. Failer L, Richter T. A parallel Newton multigrid framework for monolithic fluid-structure interactions. *J Sci Comput*. 2020;82(2):1-27.
12. Bazilevs Y, Takizawa K, Tezduyar T. *Computational Fluid-Structure Interaction: Methods and Applications*. John Wiley & Sons; 2013.
13. Richter T. *Fluid-Structure Interactions: Models, Analysis and Finite Elements*. Vol 118. Springer; 2017.
14. Donea J, Huerta A, Ponthot J-P, Rodríguez-Ferran A. Arbitrary Lagrangian–Eulerian methods. In: Stein E, Borst R, Hughes TJR, eds. *Encyclopedia of Computational Mechanics Second Edition*. Wiley; 2017:1-23.
15. Ryzhakov PB, Rossi R, Idelsohn SR, Onate E. A monolithic Lagrangian approach for fluid-structure interaction problems. *Comput Mech*. 2010;46:883-899.
16. Dunne T. An Eulerian approach to fluid-structure interaction and goal-oriented mesh adaptation. *Int J Numer Methods Fluids*. 2006;51(9-10):1017-1039.
17. Glowinski R, Pan T-W, Periaux J. A fictitious domain method for external incompressible viscous flow modeled by Navier–Stokes equations. *Comput Methods Appl Mech Eng*. 1994;112(1-4):133-148.
18. Boffi D, Gastaldi L, Heltai L. *A Distributed Lagrange Formulation of the Finite Element Immersed Boundary Method for Fluids Interacting with Compressible Solids*. Springer International Publishing; 2018:1-21.
19. Peskin CS. The immersed boundary method. *Acta Numer*. 2002;11(1):479-517.
20. Boffi D, Gastaldi L, Heltai L, Peskin C. On the hyper-elastic formulation of the immersed boundary method. *Comput Methods Appl Mech Eng*. 2008;197(25-28):2210-2231.
21. Hron J, Turek S. A monolithic FEM/multigrid solver for an ALE formulation of fluid-structure interaction with applications in biomechanics. In: Bungartz HJ, Schäfer M, eds. *Fluid-Structure Interaction*. Springer; 2006:146-170.
22. Wick T. Variational-monolithic ALE fluid-structure interaction: comparison of computational cost and mesh regularity using different mesh motion techniques. In: Bock H, Phu H, Rannacher R, Schlöder J, eds. *Modeling, Simulation and Optimization of Complex Processes HPSC 2015*. Springer; 2017:261-275.
23. Richter T. A monolithic geometric multigrid solver for fluid-structure interactions in ALE formulation. *Int J Numer Methods Eng*. 2015;104:372-390.
24. Burman E, Fernández MA. Explicit strategies for incompressible fluid-structure interaction problems: Nitsche type mortaring versus robin-robin coupling. *Int J Numer Methods Eng*. 2013;97(10):739-758.
25. Causin P, Gerbeau J-F, Nobile F. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Comput Methods Appl Mech Eng*. 2005;194(42):4506-4527.
26. Küttler U, Wall W. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Comput Mech*. 2008;43(1):61-72.

27. Landajuela M, Vidrascu M, Chapelle D, Fernández MA. Coupling schemes for the FSI forward prediction challenge: comparative study and validation. *Int J Numer Method Biomed Eng*. 2016;33(4):e2813.
28. Förster C, Wall WA, Ramm E. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Comput Methods Appl Mech Eng*. 2007;196(7):1278-1293.
29. van Brummelen EH. Added mass effects of compressible and incompressible flows in fluid-structure interaction. *J Appl Mech*. 2009;76(2):021206-021207.
30. Gerbeau J-F, Vidrascu M, Frey P. Fluid-structure interaction in blood flows on geometries based on medical imaging. *Comput Struct*. 2005;83(2):155-165.
31. Wall WA, Rabczuk T. Fluid-structure interaction in lower airways of CT-based lung geometries. *Int J Numer Methods Fluids*. 2008;57(5):653-675.
32. Jodlbauer D, Langer U, Wick T. Parallel block-preconditioned monolithic solvers for fluid-structure interaction problems. *Int J Numer Methods Eng*. 2019;117(6):623-643.
33. Murea C, Sy S. Updated Lagrangian/Arbitrary Lagrangian–Eulerian framework for interaction between a compressible neo-Hookean structure and an incompressible fluid. *Int J Numer Methods Eng*. 2017;109(8):1067-1084.
34. Badia S, Quaini A, Quarteroni A. Modular vs. non-modular preconditioners for fluid-structure systems with large added-mass effect. *Comput Methods Appl Mech Eng*. 2008;197(49-50):4216-4232.
35. Yang K, Sun P, Wang L, Xu J, Zhang L. Modeling and numerical studies for fluid-structure interaction involving an elastic rotor. *Comput Methods Appl Mech Eng*. 2016;311:788-814.
36. Xu J, Yang K. Well-posedness and robust preconditioners for discretized fluid-structure interaction systems. *Comput Methods Appl Mech Eng*. 2015;292:69-91.
37. Wichrowski M. *Fluid-Structure Interaction Problems: Velocity-Based Formulation and Monolithic Computational Methods*. PhD Thesis. Institute of Fundamental Technological Research, Polish Academy of Sciences; 2021. https://www.ippt.pan.pl/repository/doktoraty/open/2021wichrowski_m_doktorat.pdf
38. Holzapfel G. Biomechanics of soft tissue. In: Lemaitre J, ed. *The Handbook of Materials Behavior Models*. Vol 3. Academic Press; 2001:1049-1063.
39. Olshanskii MA, Peters J, Reusken A. Uniform preconditioners for a parameter dependent saddle point problem with application to generalized stokes interface equations. *Numer Math*. 2006;105(1):159-191.
40. Wichrowski M, Krzyzanowski P. A matrix-free multilevel preconditioner for the generalized stokes problem with discontinuous viscosity. *J Comput Sci*. 2022;63:101804.
41. Jodlbauer D, Langer U, Wick T, Zulehner W. Matrix-free monolithic multigrid methods for Stokes and generalized Stokes problems; 2022. https://arxiv.org/abs/2205.15770
42. Kronbichler M, Kormann K. A generic interface for parallel cell-based finite element operator application. *Comput Fluids*. 2012;63:135-147.
43. Turek S, Hron J. *Proposal for Numerical Benchmarking of Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow*. Springer; 2006.
44. Helenbrook B. Mesh deformation using the biharmonic operator. *Int J Numer Methods Eng*. 2003;56(7):1007-1021.
45. Wick T. Fluid-structure interactions using different mesh motion techniques. *Comput Struct*. 2011;89(13-14):1456-1467.
46. Ogden RW. Large deformation isotropic elasticity—on the correlation of theory and experiment for incompressible rubberlike solids. *Proc R Soc Lond A Math Phys Eng Sci*. 1972;326:565-584.
47. Bigoni D. *Nonlinear Solid Mechanics: Bifurcation Theory and Material Instability*. Cambridge University Press; 2012.
48. Le Tallec P, Mouro J. Fluid structure interaction with large structural displacements. *Comput Methods Appl Mech Eng*. 2001;190(24-25):3039-3067.
49. Melbø H, Kvamsdal T. Goal oriented error estimators for stokes equations based on variationally consistent postprocessing. *Comput Methods Appl Mech Eng*. 2003;192(5-6):613-633.
50. van Brummelen EH, van der Zee KG, Garg VV, Prudhomme S. Flux evaluation in primal and dual boundary-coupled problems. *J Appl Mech*. 2012;79(1):010904.
51. Cash JR. On the integration of stiff systems of ODEs using extended backward differentiation formulae. *Numer Math*. 1980;34(3):235-246.
52. Heil M, Hazel AL, Boyle J. Solvers for large-displacement fluid-structure interaction problems: segregated versus monolithic approaches. *Comput Mech*. 2008;43(1):91-101.
53. Langer U, Yang H. Numerical simulation of fluid-structure interaction problems with hyperelastic models: a monolithic approach. *Math Comput Simul*. 2018;145:186-208.
54. Lozovskiy A, Olshanskii MA, Salamatova V, Vassilevski YV. An unconditionally stable semi-implicit FSI finite element method. *Comput Methods Appl Mech Eng*. 2015;297:437-454.
55. van Kan JJIM. A second-order accurate pressure-correction scheme for viscous incompressible flow. *SIAM J Sci Stat Comput*. 1986;7(3):870-891.
56. Dong S, Shen J. An unconditionally stable rotational velocity-correction scheme for incompressible flows. *J Comput Phys*. 2010;229(19):7013-7029.
57. Turek S. A comparative study of time-stepping techniques for the incompressible Navier–Stokes equations: from fully implicit non-linear schemes to semi-implicit projection methods. *Int J Numer Methods Fluids*. 1996;22(10):987-1011.
58. Boffi D, Brezzi F, Fortin M. *Mixed Finite Element Methods and Applications*. Vol 44. Springer; 2013.

59. Brooks AN, Hughes T. Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Comput Methods Appl Mech Eng*. 1982;32(1-3):199-259.
60. John V, Knobloch P. On discontinuity-capturing methods for convection-diffusion equations. In: de Castro AB, Gómez D, Quintela P, Salgado P, eds. *Numerical Mathematics and Advanced Applications*. Springer; 2006:336-344.
61. Gee MW, Küttler U, Wall WA. Truly monolithic algebraic multigrid for fluid-structure interaction. *Int J Numer Methods Eng*. 2011;85(8):987-1016.
62. van Brummelen EH, van der Zee KG, de Borst R. Space/time multigrid for a fluid-structure-interaction problem. *Appl Numer Math*. 2008;58(12):1951-1971.
63. Wu Y, Cai X-C. A fully implicit domain decomposition based ALE framework for three-dimensional fluid-structure interaction with application in blood flow computation. *J Comput Phys*. 2014;258:524-537.
64. Langer U, Yang H. Recent development of robust monolithic fluid-structure interaction solvers. In: Frei S, Holm B, Richter T, Wick T, Yang H, eds. *Fluid-Structure Interactions. Modeling, Adaptive Discretization and Solvers*. Radon Series on Computational and Applied Mathematics. Vol 20. De Gruyter; 2016:169-192.
65. Arndt A, Bangerth W, Davydov D, et al. The deal.II finite element library: design, features, and insights. *Comput Math Appl*. 2021;81:407-422.
66. Kronbichler M, Sashko D, Munch P. Enhancing data locality of the conjugate gradient method for high-order matrix-free finite-element implementations. *Int J High Perform Comput Appl*. 2022;37:61-81.
67. Jodlbauer D, Langer U, Wick T. Parallel matrix-free higher-order finite element solvers for phase-field fracture problems. *Math Comput Appl*. 2020;25(3):40.
68. Kronbichler M, Heister T, Bangerth W. High accuracy mantle convection simulation through modern numerical methods. *Geophys J Int*. 2012;191(1):12-29.
69. Africa PC, Fumagalli I, Bucelli M, Zingaro A, Dede L, Quarteroni A. lifex-cfd: an open-source computational fluid dynamics solver for cardiovascular applications; 2023. https://arxiv.org/abs/2304.12032
70. Davydov D, Pelteret J-P, Arndt D, Kronbichler M, Steinmann P. A matrix-free approach for finite-strain hyperelastic problems using geometric multigrid. *Int J Numer Methods Eng*. 2020;121(13):2874-2895.
71. Adams M, Brezina M, Hu J, Tuminaro R. Parallel multigrid smoothing: polynomial versus gauss–Seidel. *J Comput Phys*. 2003;188(2):593-610.
72. Aulisa E, Bna S, Bornia G. A monolithic ALE Newton–Krylov solver with multigrid-Richardson–Schwarz preconditioning for incompressible fluid-structure interaction. *Comput Fluids*. 2018;174:213-228.

How to cite this article: Wichrowski M, Krzyżanowski P, Heltai L, Stupkiewicz S. Exploiting high-contrast Stokes preconditioners to efficiently solve incompressible fluid–structure interaction problems. *Int J Numer Methods Eng*. 2023;1-25. doi: 10.1002/nme.7350