

Article

A Parallel Solver for FSI Problems with Fictitious Domain Approach

Daniele Boffi ^{1,2}, Fabio Credali ^{1,2,*} , Lucia Gastaldi ³  and Simone Scacchi ⁴

¹ Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia

² Dipartimento di Matematica “F. Casorati”, Università degli Studi di Pavia, Via Ferrata 5, 27100 Pavia, Italy

³ Dipartimento di Ingegneria Civile, Architettura, Territorio, Ambiente e di Matematica, Università degli Studi di Brescia, Via Branze 43, 25123 Brescia, Italy

⁴ Dipartimento di Matematica “F. Enriques”, Università degli Studi di Milano, Via Saldini 50, 20133 Milano, Italy

* Correspondence: fabio.credali@kaust.edu.sa

Abstract: We present and analyze a parallel solver for the solution of fluid structure interaction problems described by a fictitious domain approach. In particular, the fluid is modeled by the non-stationary incompressible Navier–Stokes equations, while the solid evolution is represented by the elasticity equations. The parallel implementation is based on the PETSc library and the solver has been tested in terms of robustness with respect to mesh refinement and weak scalability by running simulations on a Linux cluster.

Keywords: fluid–structure interactions; fictitious domain; preconditioners; parallel solver

MSC: 65N30; 65N12; 74F10; 65F08



Citation: Boffi, D.; Credali, F.; Gastaldi, L.; Scacchi, S. A Parallel Solver for FSI Problems with Fictitious Domain Approach. *Math. Comput. Appl.* **2023**, *28*, 59. <https://doi.org/10.3390/mca28020059>

Academic Editors: Hans Beushausen, Sebastian Skatulla and Suchuan Dong

Received: 24 January 2023

Revised: 4 March 2023

Accepted: 6 April 2023

Published: 10 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The analysis of fluid–structure interaction (FSI) problems is important for several applications in science and engineering; typical examples that we have in mind are, for instance, the study of fluid-dynamics of heart valves, particulate flows, and propagation of free surfaces. FSI problems are challenging both from the mathematical and computational point of view: the difficulties originate from the necessity of handling interactions between several objects and the presence of nonlinear terms in the governing equations. As a consequence, in past years, several approaches have been presented for the numerical modeling of FSI problems: among those, we mention the Arbitrary Lagrangian Eulerian formulation (ALE) [1–4], the unfitted Nitsche method [5], the level set formulation [6], and the fictitious domain approach [7,8]. It is important to notice that each method is effective for a selected class of problems, since there is no method that can be applied to all possible situations.

Our fictitious domain approach with distributed Lagrange multiplier was considered first in [9] as evolution of the immersed boundary method [10,11], originally introduced by C. Peskin during the Seventies for cardiac simulations of heart valves. This approach is based on the idea that the fluid domain is fictitiously extended also to the region occupied by the immersed body. In particular, the fluid is governed by the incompressible time dependent Navier–Stokes equations, while the structure can be characterized by either linear or nonlinear constitutive laws for viscous elastic materials. The fluid dynamics is studied on a fixed Eulerian mesh, while we resort to a Lagrangian description to represent the motion and the deformation of the immersed structure; the Lagrangian frame is built by introducing a reference domain which is mapped, at each time step, into the actual position of the body.

In order to get accurate results, simulations of FSI problems require, in general, huge computational resources, both in terms of time and memory. In this framework, the design

of robust parallel solvers is an important tool to perform, in a reasonable amount of time, computations involving a large number of time steps and fine space discretizations. Several works are focused on this task, mainly in the setting of ALE formulations [12–18] or by applying a variational transfer in order to couple fluid and solids [19,20]. To the best of our knowledge, preconditioners for the fictitious domain formulation with distributed Lagrange multiplier have been introduced only recently in [21].

Here, we continue the analysis of the parallel solver introduced in [21]: we focus our attention on the robustness with respect to mesh refinement, with different choices of time step, and to the weak scalability. We also describe some implementation issues. The finite element discretization is performed by choosing the (Q_2, P_1) element for velocities and pressures of the fluid and the Q_1 element for the structure variables; the time marching scheme is a first order semi-implicit finite difference algorithm. Moreover, the fluid–structure coupling matrix is assembled by exact computations over non-matching meshes as described in [22]; the computational costs of this procedure will be described. At each time step, the linear system arising from the discretization is solved by the GMRES method, combined with either a block-diagonal or a block-triangular preconditioner. Our parallel implementation is based on the PETSc library from Argonne National Laboratory [23,24] and on the parallel direct solver Mumps [25,26], which is used to invert the diagonal blocks in the block preconditioners.

After recalling some functional analysis notation, in Section 3, we present the mathematical model describing fluid structure interaction problems in the spirit of the fictitious domain approach. In Section 4, we describe the numerical method we implemented for our simulations and in Section 5 we introduce two possible choices of preconditioner for our parallel solver. Finally, in Section 6 we present some numerical tests aiming at assessing the robustness with respect to mesh refinement and the weak scalability.

2. Notation

We recall some useful functional analysis notation [27]. Let us consider an open and bounded domain D . The space of square integrable functions is denoted by $L^2(D)$, with scalar product $(\cdot, \cdot)_D$. In particular, $L_0^2(D)$ is the subspace of functions with null mean over D . We denote Sobolev spaces by $W^{s,q}(D)$: with $s \in \mathbb{R}$ referring to the differentiability and $q \in [1, \infty]$ to the summability exponent. When $q = 2$, we adopt the classical notation $H^s(D) = W^{s,2}(D)$. In addition, $H_0^1(D) \subset H^1(D)$ is the space of functions with zero trace on the boundary ∂D . For vector valued spaces the dimension is explicitly indicated.

3. Continuous Formulation

We simulate fluid–structure interaction problems characterized by a visco-elastic incompressible solid body immersed in a viscous incompressible fluid. We denote by Ω_t^f and Ω_t^s the two regions in \mathbb{R}^d (with $d = 2, 3$) occupied by the fluid and the structure, respectively, at the time instant t ; the interface between these two regions is denoted by Γ_t . The evolution of such a system takes place inside Ω , that is the union of Ω_t^f and Ω_t^s : this new domain is independent of time and we assume that it is connected and bounded with Lipschitz continuous boundary $\partial\Omega$. It is worth mentioning that, even if we are going to consider only thick solids, also the evolution of thin structures can be treated by our mathematical model.

The dynamics of the fluid is studied by considering an Eulerian description, associated with the variable \mathbf{x} . On the other hand, the evolution of the immersed body is modeled by a Lagrangian description: we introduce the solid reference domain \mathcal{B} , associated with the variable \mathbf{s} , so that the deformation can be represented by the map $\mathbf{X} : \mathcal{B} \rightarrow \Omega_t^s$. This means that $\mathbf{x} \in \Omega_t^s$ is the image at the time t of a certain point $\mathbf{s} \in \mathcal{B}$ and the motion of the structure is represented by the kinematic equation

$$\mathbf{u}_s(\mathbf{x}, t) = \frac{\partial \mathbf{X}}{\partial t}(\mathbf{s}, t) \quad \text{for } \mathbf{x} = \mathbf{X}(\mathbf{s}, t), \quad (1)$$

where the material velocity is denoted by \mathbf{u}_s . The deformation gradient $\nabla_s \mathbf{X}$ is denoted by \mathbb{F} and $J(\mathbf{s}, t) = \det \mathbb{F}(\mathbf{s}, t)$. Since we are assuming that the solid body is incompressible, the determinant J is constant in time.

In our model, we consider a Newtonian fluid with density ρ_f and viscosity $\nu_f > 0$, so that the Cauchy stress tensor can be written as

$$\sigma_f = -p_f \mathbb{I} + \nu_f \underline{\varepsilon}(\mathbf{u}_f), \tag{2}$$

where \mathbf{u}_f denotes the velocity of the fluid and p_f its pressure, and where \mathbb{I} denotes the identity tensor. In particular, the symbol $\underline{\varepsilon}(\cdot)$ refers to the symmetric gradient $\underline{\varepsilon}(\mathbf{v}) = (\nabla \mathbf{v} + \nabla \mathbf{v}^\top)/2$. Therefore, the dynamics in Ω_t^f is governed by the incompressible Navier–Stokes equations

$$\begin{aligned} \rho_f \left(\frac{\partial \mathbf{u}_f}{\partial t} + \mathbf{u}_f \cdot \nabla \mathbf{u}_f \right) &= \operatorname{div} \sigma_f \\ \operatorname{div} \mathbf{u}_f &= 0. \end{aligned} \tag{3}$$

For the solid, we consider a viscous-hyperelastic material with density ρ_s and viscosity $\nu_s > 0$; for this type of materials, the Cauchy stress tensor σ_s can be seen as the sum of two contributions: a viscous part, similar to the one of the fluid

$$\sigma_s^v = -p_s \mathbb{I} + \nu_s \underline{\varepsilon}(\mathbf{u}_s), \tag{4}$$

and an elastic part, which can be written, moving from Eulerian to Lagrangian setting, in terms of the Piola–Kirchhoff stress tensor \mathbb{P}

$$\mathbb{P}(\mathbb{F}(\mathbf{s}, t)) = J(\mathbf{s}, t) \sigma_s^e(\mathbf{x}, t) \mathbb{F}(\mathbf{s}, t)^{-\top} \quad \text{for } \mathbf{x} = \mathbf{X}(\mathbf{s}, t). \tag{5}$$

In particular, hyperelastic materials are characterized by a positive energy density $W(\mathbb{F})$, which is related with \mathbb{P} since $\mathbb{P}(\mathbb{F}) = \partial W / \partial \mathbb{F}$. Consequently, the elastic potential energy of the solid body can be expressed as

$$E(\mathbf{X}(t)) = \int_B W(\mathbb{F}(\mathbf{s}, t)) \, ds. \tag{6}$$

Finally, the system is described by the following equations in strong form

$$\begin{aligned} \rho_f \left(\frac{\partial \mathbf{u}_f}{\partial t} + \mathbf{u}_f \cdot \nabla \mathbf{u}_f \right) &= \operatorname{div} \sigma_f && \text{in } \Omega_t^f \\ \operatorname{div} \mathbf{u}_f &= 0 && \text{in } \Omega_t^f \\ \rho_s \frac{\partial^2 \mathbf{X}}{\partial t^2} &= \operatorname{div}_s (J \sigma_s^v \mathbb{F}^{-\top} + \mathbb{P}(\mathbb{F})) && \text{in } \mathcal{B} \\ \operatorname{div} \mathbf{u}_s &= 0 && \text{in } \Omega_t^s \end{aligned} \tag{7}$$

and completed by two transmission conditions to enforce continuity of velocity and stress along the interface Γ_t

$$\begin{aligned} \mathbf{u}_f &= \mathbf{u}_s && \text{on } \Gamma_t \\ \sigma_f \mathbf{n}_f &= -(\sigma_s^v + J^{-1} \mathbb{P} \mathbb{F}^\top) \mathbf{n}_s && \text{on } \Gamma_t, \end{aligned} \tag{8}$$

where \mathbf{n}_f and \mathbf{n}_s denote the outer normals to Ω_t^f and Ω_t^s , respectively. Moreover, we consider the following initial and boundary conditions

$$\begin{aligned}
 \mathbf{u}_f(0) &= \mathbf{u}_{f,0} && \text{in } \Omega_0^f \\
 \mathbf{u}_s(0) &= \mathbf{u}_{s,0} && \text{in } \Omega_0^s \\
 \mathbf{X}(0) &= \mathbf{X}_0 && \text{in } \mathcal{B} \\
 \mathbf{u}_f &= 0 && \text{on } \partial\Omega.
 \end{aligned}
 \tag{9}$$

The idea of the fictitious domain approach is to extend the first two equations in (7) to the whole domain Ω so that all the involved variables are defined on a domain which is independent of time. Consequently, following [9], we introduce two new unknowns

$$\mathbf{u} = \begin{cases} \mathbf{u}_f & \text{in } \Omega_t^f \\ \mathbf{u}_s & \text{in } \Omega_t^s \end{cases} \quad p = \begin{cases} p_f & \text{in } \Omega_t^f \\ p_s & \text{in } \Omega_t^s. \end{cases}
 \tag{10}$$

In this new setting, (1) becomes a constraint on \mathbf{u} , since we have to impose that

$$\mathbf{u}(\mathbf{X}(\mathbf{s}, t), t) = \frac{\partial \mathbf{X}}{\partial t}(\mathbf{s}, t) \quad \text{for } \mathbf{s} \in \mathcal{B}.
 \tag{11}$$

This condition can be weakly enforced by employing a distributed Lagrange multiplier. To this end, we set $\Lambda = (H^1(\mathcal{B})^d)'$, the dual space of $H^1(\mathcal{B})^d$, and denote by $\langle \cdot, \cdot \rangle$ the duality pairing between Λ and $H^1(\mathcal{B})^d$. Notice that, for $\mathbf{Y} \in H^1(\mathcal{B})^d$, we have the following property

$$\langle \boldsymbol{\mu}, \mathbf{Y} \rangle = 0 \quad \forall \boldsymbol{\mu} \in \Lambda \implies \mathbf{Y} = 0.
 \tag{12}$$

At this point, following [9,28], the equations in (7), endowed with conditions (8) and (9), can be written in variational form.

Problem 1. For given $\mathbf{u}_0 \in H_0^1(\Omega)^d$ and $\mathbf{X}_0 \in W^{1,\infty}(\mathcal{B})$, find $\mathbf{u}(t) \in H_0^1(\Omega)^d$, $p(t) \in L_0^2(\Omega)$, $\mathbf{X}(t) \in H^1(\mathcal{B})^d$, and $\boldsymbol{\lambda}(t) \in \Lambda$ such that for almost all $t \in (0, T)$:

$$\begin{aligned}
 \rho_f \left(\frac{\partial}{\partial t} \mathbf{u}(t), \mathbf{v} \right)_{\Omega} + b(\mathbf{u}(t), \mathbf{u}(t), \mathbf{v}) + a(\mathbf{u}(t), \mathbf{v}) \\
 - (\operatorname{div} \mathbf{v}, p(t))_{\Omega} + \langle \boldsymbol{\lambda}(t), \mathbf{v}(\mathbf{X}(\cdot, t)) \rangle &= 0 \quad \forall \mathbf{v} \in H_0^1(\Omega)^d \\
 (\operatorname{div} \mathbf{u}(t), q)_{\Omega} &= 0 \quad \forall q \in L_0^2(\Omega) \\
 (\rho_s - \rho_f) \left(\frac{\partial^2 \mathbf{X}}{\partial t^2}(t), \mathbf{Y} \right)_{\mathcal{B}} + (\mathbb{P}(\mathbb{F}(t)), \nabla_s \mathbf{Y})_{\mathcal{B}} - \langle \boldsymbol{\lambda}(t), \mathbf{Y} \rangle &= 0 \quad \forall \mathbf{Y} \in H^1(\mathcal{B})^d \\
 \langle \boldsymbol{\mu}, \mathbf{u}(\mathbf{X}(\cdot, t), t) - \frac{\partial \mathbf{X}}{\partial t}(t) \rangle &= 0 \quad \forall \boldsymbol{\mu} \in \Lambda \\
 \mathbf{u}(\mathbf{x}, 0) &= \mathbf{u}_0(\mathbf{x}) \quad \text{in } \Omega \\
 \mathbf{X}(\mathbf{s}, 0) &= \mathbf{X}_0(\mathbf{s}) \quad \text{in } \mathcal{B}.
 \end{aligned}$$

In particular,

$$\begin{aligned}
 a(\mathbf{u}, \mathbf{v}) &= \nu (\boldsymbol{\varepsilon}(\mathbf{u}), \boldsymbol{\varepsilon}(\mathbf{v}))_{\Omega} \\
 b(\mathbf{u}, \mathbf{v}, \mathbf{w}) &= \frac{\rho_f}{2} ((\mathbf{u} \cdot \nabla \mathbf{v}, \mathbf{w})_{\Omega} - (\mathbf{u} \cdot \nabla \mathbf{w}, \mathbf{v})_{\Omega}).
 \end{aligned}
 \tag{13}$$

Moreover, ν is the extended viscosity with value ν_f in Ω_t^f and ν_s in Ω_t^s . For our numerical tests, we are going to consider $\nu_f = \nu_s$ since it is a reasonable assumption for biological models [29].

For our simulations, we consider a simplified version of the problem: we drop the convective term of the Navier–Stokes equations and we assume that fluid and solid materials have the same density, i.e., $\rho_s = \rho_f$. We focus on this case since it is interesting to see how the solver behaves when this assumption is combined with a semi-implicit time advancing scheme in the setting of the fictitious domain approach. This is actually the

critical situation when the added mass effect can cause instabilities. For instance, in [30], a simplified one-dimensional setting is considered for which non implicit schemes are proved to be unconditionally unstable when applied to FSI problems modeled by ALE if fluid and solid have the same densities $\rho_s = \rho_f$. This phenomenon appears regardless of the discrete parameters. In order to alleviate such critical behavior, an appropriate treatment of the transmission conditions has been investigated, for instance, in [31,32]). Therefore, the problem we are going to simulate reads as follows.

Problem 2. Given $\mathbf{u}_0 \in H_0^1(\Omega)^d$ and $\mathbf{X}_0 \in W^{1,\infty}(\mathcal{B})$, find $\mathbf{u}(t) \in H_0^1(\Omega)^d$, $p(t) \in L_0^2(\Omega)$, $\mathbf{X}(t) \in H^1(\mathcal{B})^d$, and $\lambda(t) \in \Lambda$ such that for almost all $t \in (0, T)$:

$$\begin{aligned} \rho_f \left(\frac{\partial}{\partial t} \mathbf{u}(t), \mathbf{v} \right)_{\Omega} + a(\mathbf{u}(t), \mathbf{v}) & - (\operatorname{div} \mathbf{v}, p(t))_{\Omega} + \langle \lambda(t), \mathbf{v}(\mathbf{X}(\cdot, t)) \rangle = 0 & \forall \mathbf{v} \in H_0^1(\Omega)^d \\ (\operatorname{div} \mathbf{u}(t), q)_{\Omega} = 0 & \forall q \in L_0^2(\Omega) \\ (\mathbb{P}(\mathbb{F}(t)), \nabla_s \mathbf{Y})_{\mathcal{B}} - \langle \lambda(t), \mathbf{Y} \rangle = 0 & \forall \mathbf{Y} \in H^1(\mathcal{B})^d \\ \langle \boldsymbol{\mu}, \mathbf{u}(\mathbf{X}(\cdot, t), t) - \frac{\partial \mathbf{X}}{\partial t}(t) \rangle = 0 & \forall \boldsymbol{\mu} \in \Lambda \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) & \text{in } \Omega \\ \mathbf{X}(\mathbf{s}, 0) = \mathbf{X}_0(\mathbf{s}) & \text{in } \mathcal{B}. \end{aligned}$$

4. Discrete Formulation

Before discussing the discrete formulation, we remark that, from now on, we will focus on two-dimensional problems ($d = 2$).

The time semi-discretization of Problem 2 is based on the Backward Euler scheme. The time interval $[0, T]$ is uniformly partitioned into N parts with size $\Delta t = T/N$. We denote the subdivision nodes by $t_n = n\Delta t$. For a generic function g depending on time, setting $g^n = g(t_n)$, the time derivative is approximated as

$$\frac{\partial g}{\partial t}(t_{n+1}) \approx \frac{g^{n+1} - g^n}{\Delta t}. \tag{14}$$

Moreover, the nonlinear coupling terms $\langle \lambda(t), \mathbf{v}(\mathbf{X}(\cdot, t)) \rangle$ and $\langle \boldsymbol{\mu}, \mathbf{u}(\mathbf{X}(\cdot, t), t) \rangle$ are semi-implicitly treated by considering the position of the structure at the previous time step as $\langle \lambda^{n+1}, \mathbf{v}(\mathbf{X}^n) \rangle$ and $\langle \boldsymbol{\mu}, \mathbf{u}^{n+1}(\mathbf{X}^n) \rangle$.

For the discretization in space, we work with quadrilateral meshes for both fluid and solid. For the fluid, we consider a partition \mathcal{T}_h^{Ω} of Ω with meshsize h_{Ω} and two finite element spaces $\mathbf{V}_h \subset H_0^1(\Omega)^d$ and $Q_h \subset L_0^2(\Omega)$ for velocity and pressure, respectively, satisfying the inf-sup condition for the Stokes problem. In particular, we work with the $(\mathcal{Q}_2, \mathcal{P}_1)$ pair, which is one of the most popular Stokes elements, making use of continuous piecewise quadratic velocities and discontinuous piecewise linear pressures.

For the solid domain, we choose a partition $\mathcal{T}_h^{\mathcal{B}}$ of \mathcal{B} with meshsize $h_{\mathcal{B}}$, independent of \mathcal{T}_h^{Ω} . We then consider two finite dimensional spaces $\mathbf{S}_h \subset H^1(\mathcal{B})^d$ and $\Lambda_h \subset \Lambda$. We assume that $\mathbf{S}_h = \Lambda_h$ and we approximate both the mapping \mathbf{X} and the Lagrange multiplier λ with piecewise bilinear elements on quadrilaterals. Other stable combinations of finite element spaces for our class of problems have been studied in [33], both from the theoretical and the numerical point of view.

We notice that, since Λ_h is included in $L^2(\mathcal{B})^d$, at a discrete level, the duality pairing can be replaced by the scalar product in $L^2(\mathcal{B})^d$

$$\langle \boldsymbol{\mu}_h, \mathbf{Y}_h \rangle = (\boldsymbol{\mu}_h, \mathbf{Y}_h)_{\mathcal{B}} \quad \forall \boldsymbol{\mu}_h \in \Lambda_h, \forall \mathbf{Y}_h \in \mathbf{S}_h. \tag{15}$$

Therefore, we get the following fully discrete problem.

Problem 3. Given $\mathbf{u}_{0,h} \in \mathbf{V}_h$ and $\mathbf{X}_{0,h} \in \mathbf{S}_h$, for all $n = 1, \dots, N$ find $\mathbf{u}_h^n \in \mathbf{V}_h$, $p_h^n \in Q_h$, $\mathbf{X}_h^n \in \mathbf{S}_h$, and $\lambda_h^n \in \Lambda_h$ fulfilling:

$$\begin{aligned} \rho_f \left(\frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^n}{\Delta t}, \mathbf{v}_h \right)_{\Omega} + a(\mathbf{u}_h^{n+1}, \mathbf{v}_h) \\ - \left(\operatorname{div} \mathbf{v}_h, p_h^{n+1} \right)_{\Omega} + \left(\lambda_h^{n+1}, \mathbf{v}_h(\mathbf{X}_h^n) \right)_{\mathcal{B}} = 0 \quad \forall \mathbf{v}_h \in \mathbf{V}_h \\ \left(\operatorname{div} \mathbf{u}_h^{n+1}, q_h \right)_{\Omega} = 0 \quad \forall q_h \in Q_h \\ \left(\mathbb{P}(\mathbb{F}_h^{n+1}), \nabla_s \mathbf{Y}_h \right)_{\mathcal{B}} - \left(\lambda_h^{n+1}, \mathbf{Y}_h \right)_{\mathcal{B}} = 0 \quad \forall \mathbf{Y}_h \in \mathbf{S}_h \\ \left(\mu_h, \mathbf{u}_h^n(\mathbf{X}_h^n) - \frac{\mathbf{X}_h^{n+1} - \mathbf{X}_h^n}{\Delta t} \right)_{\mathcal{B}} = 0 \quad \forall \mu_h \in \Lambda_h \\ \mathbf{u}_h^0 = \mathbf{u}_{0,h}, \quad \mathbf{X}_h^0 = \mathbf{X}_{0,h}. \end{aligned}$$

Assuming for simplicity $\mathbb{P}(\mathbb{F}) = \kappa \mathbb{F}$, Problem 3 can be represented in matrix form as

$$\left[\begin{array}{cc|cc} A_f & -B^T & 0 & C_f(\mathbf{X}_h^n)^T \\ -B & 0 & 0 & 0 \\ \hline 0 & 0 & A_s & -C_s^T \\ C_f(\mathbf{X}_h^n) & 0 & -\frac{1}{\Delta t} C_s & 0 \end{array} \right] \begin{bmatrix} \mathbf{u}_h^{n+1} \\ p_h^{n+1} \\ \mathbf{X}_h^{n+1} \\ \lambda_h^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_1 \\ 0 \\ 0 \\ \mathbf{g}_2 \end{bmatrix}, \tag{16}$$

with

$$\begin{aligned} A_f &= \frac{\rho_f}{\Delta t} M_f + K_f \\ (M_f)_{ij} &= \left(\boldsymbol{\phi}_j, \boldsymbol{\phi}_i \right)_{\Omega}, \quad (K_f)_{ij} = a(\boldsymbol{\phi}_j, \boldsymbol{\phi}_i) \\ B_{ki} &= \left(\operatorname{div} \boldsymbol{\phi}_i, \psi_k \right)_{\Omega} \\ (A_s)_{ij} &= \kappa \left(\nabla_s \chi_j, \nabla_s \chi_i \right)_{\mathcal{B}} \\ (C_f(\mathbf{X}_h^n))_{\ell j} &= \left(\chi_{\ell}, \boldsymbol{\phi}_j(\mathbf{X}_h^n) \right)_{\mathcal{B}}, \quad (C_s)_{\ell j} = \left(\chi_{\ell}, \chi_j \right)_{\mathcal{B}} \\ \mathbf{g}_1 &= \frac{\rho_f}{\Delta t} M_f \mathbf{u}_h^n, \quad \mathbf{g}_2 = -\frac{1}{\Delta t} C_s \mathbf{X}_h^n. \end{aligned}$$

Here, $\boldsymbol{\phi}_i$ and ψ_k denote the basis functions of \mathbf{V}_h and Q_h respectively, while χ_j are the basis functions of the space defined on \mathcal{B} . We observe that, since $\mathbf{S}_h = \Lambda_h$, C_s is the mass matrix in \mathbf{S}_h or Λ_h .

We can see that the matrix in (16) splits into four blocks, defined as follows:

$$\begin{aligned} \mathcal{A}_{11} &= \begin{bmatrix} A_f & -B^T \\ -B & 0 \end{bmatrix} & \mathcal{A}_{12} &= \begin{bmatrix} 0 & C_f(\mathbf{X}_h^n)^T \\ 0 & 0 \end{bmatrix} \\ \mathcal{A}_{21} &= \begin{bmatrix} 0 & 0 \\ C_f(\mathbf{X}_h^n) & 0 \end{bmatrix} & \mathcal{A}_{22} &= \begin{bmatrix} A_s & -C_s^T \\ -\frac{1}{\Delta t} C_s & 0 \end{bmatrix} \end{aligned}$$

where \mathcal{A}_{11} is related to the fluid dynamic, \mathcal{A}_{22} to the solid evolution, while \mathcal{A}_{12} and \mathcal{A}_{21} contain the coupling term.

Particular attention has to be paid to the assembly of the coupling matrix $C_f(\mathbf{X}_h^n)$, since it involves the integration over \mathcal{B} of solid and mapped fluid basis functions. In order to compute these integrals, we need to know how each element E of $\mathcal{T}_h^{\mathcal{B}}$ is mapped into the fluid domain. We implement an exact quadrature rule by computing, at each time step, the intersection between the fluid mesh \mathcal{T}_h^{Ω} and a mapped solid element $\mathbf{X}_h^n(E)$. In order to detect all the intersections, each solid element is tested against all the fluid elements making use of a bounding box technique, which, in this particular case, is trivial since the

fluid is discretized with a Cartesian mesh of squares; then the intersections are explicitly computed by means of the Sutherland–Hodgman algorithm. For more details about the procedure in a similar situation, we refer to [22]. The implementation of this composite rule is quite involved and it is not straightforwardly parallelizable.

In general, when $\mathbb{P}(\mathbb{F})$ is nonlinear, we use a solver for nonlinear systems of equations such as the Newton iterator method.

5. Parallel Preconditioners

The design of an efficient parallel solver influences two aspects of the numerical method: first, the finite element matrices need to be assembled in parallel on each processor; second, the solution of the saddle point system arising from the discretization has to be solved saving computational resources, in terms of both memory and execution time. For this purpose, we implemented a Fortran90 code based on the library PETSc from Argonne National Laboratory [23,24]. Such library is built on the MPI standard and it offers advanced data structures and routines for the parallel solution of partial differential equations, from basic vector and matrix operations to more complex linear and nonlinear equation solvers. In our code, vectors and matrices are built and subassembled in parallel on each processor.

Our parallel solver adopts two possible choices of preconditioner:

- *block-diagonal preconditioner*

$$\begin{bmatrix} \mathcal{A}_{11} & 0 \\ 0 & \mathcal{A}_{22} \end{bmatrix}$$

- *block-triangular preconditioner*

$$\begin{bmatrix} \mathcal{A}_{11} & 0 \\ \mathcal{A}_{21} & \mathcal{A}_{22} \end{bmatrix}.$$

We solve the linear system making use of the parallel GMRES method combined with the action of our preconditioners, which consists of the exact inversion of the diagonal blocks performed by the parallel direct solver Mumps [25,26].

6. Numerical Tests

The proposed preconditioners have been widely studied in [21] in terms of robustness with respect to mesh refinement, strong scalability and refinement of the time step. In this work, after reporting new results in terms of optimality, we analyze the weak scalability of our solver. We focus on both linear and nonlinear models describing the solid material.

In the GMRES solver, we adopt as stopping criterion a 10^{-8} reduction of the Euclidean norm of the relative residual and a restart parameter of 200. In the case of the nonlinear model, the stopping criterion adopted for the Newton method is a 10^{-6} reduction of the Euclidean norm of the relative residual.

Our simulations were run on the Shaheen cluster at King Abdullah University of Science and Technology (KAUST, Saudi Arabia). It is a Cray XC40 cluster constituted by 6174 dual sockets computing nodes, based on 16 core Intel Haswell processors running at 2.3 GHz. Each node has 128 GB of DDR4 memory running at 2300 MHz.

6.1. Linear Solid Model

We consider a quarter of the elastic annulus $\{\mathbf{x} \in \mathbb{R}^2 : 0.3 \leq |\mathbf{x}| \leq 0.5\}$ included in $\Omega = [0, 1]^2$: in particular, the solid reference domain corresponds to the resting configuration of the body; that is

$$\mathcal{B} = \{\mathbf{s} = (s_1, s_2) \in \mathbb{R}^2 : s_1, s_2 \geq 0, 0.3 \leq |\mathbf{s}| \leq 0.5\}.$$

The dynamics of the system is generated by stretching the annulus and observing how internal forces bring it back to the resting condition. In this case, Ω_0^s coincides with the stretched annulus. Four snapshots of the evolution are shown in Figure 1.

The solid behavior is governed by a linear model; therefore, $\mathbb{P}(\mathbb{F}) = \kappa \mathbb{F}$, with $\kappa = 10$. We choose fluid and solid materials with the same density $\rho_f = \rho_s = 1$ and the same viscosity $\nu_f = \nu_s = 0.1$. We impose no slip conditions for the velocity on the upper and right edge of Ω , while on the other two edges, we allow the motion of both fluid and structure along the tangential direction. Finally, the following initial conditions are considered

$$\mathbf{u}(\mathbf{x}, 0) = 0, \quad \mathbf{X}(\mathbf{s}, 0) = \left(\frac{s_1}{1.4}, 1.4 s_2 \right).$$

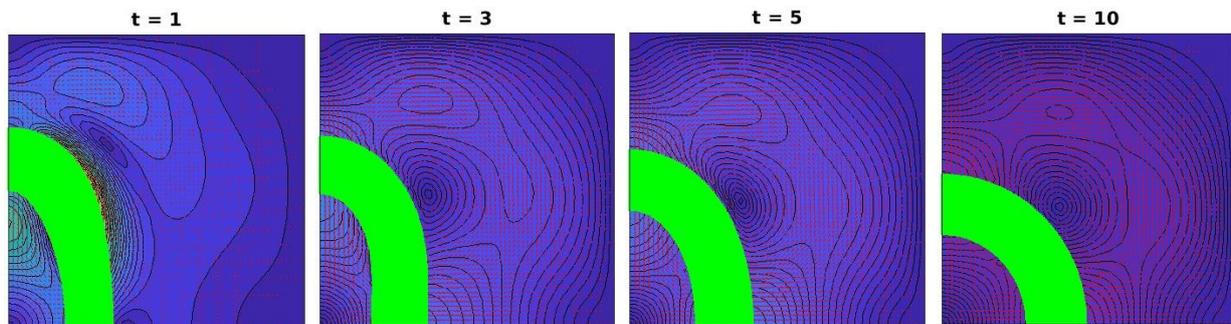


Figure 1. Four snapshots of the evolution of the structure with linear constitutive law.

In Table 1, we report the results for the optimality test, where the robustness of the solver is studied by refining the mesh and keeping fixed the number of processors. In particular, we set the time step to $\Delta t = 0.01$ and the final time of our simulation to $T = 2$. The number of processors used for the simulation is 32. The time T_{ass} needed to assemble the matrix of the problem increases moderately, while the time T_{coup} , needed for the assembly of the coupling matrix by computing the intersection between the involved meshes, exhibits a superlinear growth. In terms of preconditioners, we can see that block-diag is not robust with respect to mesh refinement since the number of GMRES iterations grows from 13 to 430; clearly, this phenomenon affects also the time T_{sol} we need to solve the system. On the other hand, block-tri is robust since the number of GMRES iterations remains bounded by 14 when the mesh is refined. Therefore, T_{sol} presents only a moderate growth and, for 1,074,054 dofs, it is 30 times smaller than the value we obtain for block-diag preconditioner.

Table 1. Refinements of the mesh in the linear solid model. The simulations are run on the Shaheen cluster. procs = number of processors; dofs = degrees of freedom; T_{ass} = CPU time to assemble the stiffness and mass matrices; T_{coup} = CPU time to assemble the coupling term; its = GMRES iterations; T_{sol} = CPU time to solve the linear system; T_{tot} = total simulation CPU time. The quantities T_{coup} , its and T_{sol} are averaged over the time steps. All CPU times are reported in seconds.

Linear Solid Model—Mesh Refinement Test								
procs = 32, T = 2, $\Delta t = 0.01$								
dofs	T_{ass} (s)	T_{coup} (s)	Block-diag			Block-tri		
			its	T_{sol} (s)	T_{tot} (s)	its	T_{sol} (s)	T_{tot} (s)
30,534	1.02×10^{-2}	9.98×10^{-2}	13	1.14×10^{-1}	42.01	7	6.93×10^{-2}	33.83
120,454	2.12×10^{-2}	1.09	31	8.30×10^{-1}	390.90	9	2.40×10^{-1}	266.17
269,766	9.20×10^{-2}	7.60	97	5.41	2.55×10^3	11	6.47×10^{-1}	1.65×10^3
478,470	1.31×10^{-1}	25.04	192	18.75	9.07×10^3	12	1.14	5.24×10^3
746,566	1.23×10^{-1}	85.32	422	67.92	3.07×10^4	13	2.17	1.75×10^4
1,074,054	1.81×10^{-1}	196.88	430	97.19	5.90×10^4	14	3.21	4.00×10^4

The weak scalability of the proposed parallel solver is analyzed in Table 2. Again, we choose $T = 2$ and $\Delta t = 0.01$. We perform six tests by doubling both the global number of dofs and the number of processors. Thanks to the resources provided by PETSc, the time

T_{ass} to assemble stiffness and mass matrices is perfectly scalable. On the other hand, the assembly procedure for the coupling matrix is much more complicated: in order to detect all the intersections between solid and fluid elements, the algorithm consists of two nested loops. For each solid element (outer loop), we check its position with respect to all the fluid elements (inner loop). In particular, only the outer loop is distributed over all the processors. Consequently, T_{coup} is not scalable since the number of fluid dofs, analyzed in serial, increases at each test. We now discuss the behavior of the two proposed preconditioners. It is evident that block-diag is not scalable since the number of GMRES iteration drastically increases as we increase dofs and procs, clearly affecting T_{sol} and T_{tot} . On the other hand, block-tri behaves well: even if it is not perfectly scalable, the number of iterations slightly increases from 8 to 18 and T_{sol} ranges from 2.24×10^{-1} s to 11.43 s.

Table 2. Weak scalability for the linear solid model. The simulations are run on the Shaheen cluster. Same format as Table 1.

Linear Solid Model—Weak Scalability Test									
T = 2, Δt = 0.01									
procs	dofs	T_{ass} (s)	T_{coup} (s)	Block-diag			Block-tri		
				its	T_{sol} (s)	T_{tot} (s)	its	T_{sol} (s)	T_{tot} (s)
4	68,070	8.55×10^{-2}	3.95	22	6.25×10^{-1}	933.43	8	2.24×10^{-1}	833.44
8	135,870	1.00×10^{-1}	5.23	38	2.16	1.48×10^3	9	4.41×10^{-1}	1.13×10^3
16	269,766	1.01×10^{-1}	8.77	111	10.23	3.80×10^3	11	9.70×10^{-1}	1.95×10^3
32	539,926	9.24×10^{-2}	59.27	706	108.05	2.50×10^4	18	2.91	1.24×10^4
64	1,074,054	1.90×10^{-1}	48.00	429	113.59	3.24×10^4	14	3.90	1.04×10^4
128	2,152,614	1.90×10^{-1}	98.63	-	-	-	18	11.43	2.20×10^4

6.2. Nonlinear Solid Model

For this test, we set again the fluid domain Ω to be the unit square; the immersed solid body is a bar represented, at resting configuration, by the rectangle $\mathcal{B} = \Omega_0^s = [0, 0.4] \times [0.45, 0.55]$. During the time interval $[0, 1]$, the structure is pulled down by a force applied at the middle point of the right edge. Therefore, when released, the solid body returns to its resting configuration by the action of internal forces. Four snapshots of the evolution are shown in Figure 2.

The energy density of the solid material is given by the potential strain energy function of an isotropic hyperelastic material; in particular, we have

$$W(\mathbb{F}) = (\gamma/2\eta) \exp(\eta[\text{tr}(\mathbb{F}^T \mathbb{F}) - 2]),$$

where $\text{tr}(\mathbb{F}^T \mathbb{F})$ denotes the trace of $\mathbb{F}^T \mathbb{F}$, while $\gamma = 1.333$ and $\eta = 9.242$. It can be proved that W is a strictly locally convex strain energy function, as discussed in [34–36].

Also for this test we assume that fluid and solid materials share the same density, equal to 1, and the same viscosity, equal to 0.2. The velocity is imposed to be zero at the boundary of Ω , while the following initial conditions are imposed

$$\mathbf{u}(\mathbf{x}, 0) = 0, \quad \mathbf{X}(\mathbf{s}, 0) = \mathbf{s}.$$

Results for the mesh refinement test are reported in Table 3: we consider the evolution of the system during the time interval $[0, 2]$, with time step $\Delta t = 0.002$. The number of processors for the simulations is set to 64, while the number of dofs increases from 21,222 to 741,702. As for the linear case, T_{ass} increases moderately and T_{coup} follows a superlinear growth. Both preconditioners are robust with respect to mesh refinement: the number of Newton iterations is 2 for each test and the average number of GMRES iterations per nonlinear iteration is bounded by 15 for block-diag and by 10 for block-tri. This behavior of block-diag is in contrast with the results we obtained for the linear solid model: this is due to the finer time step chosen for this simulation.

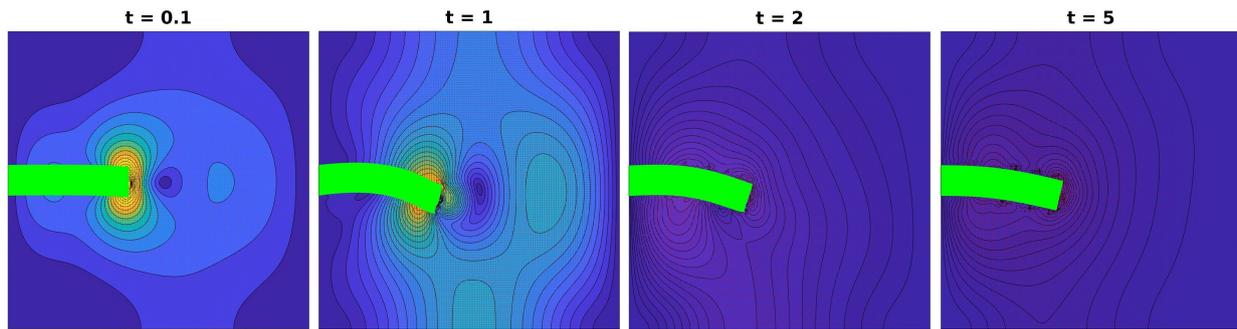


Figure 2. Four snapshots of the evolution of the structure with nonlinear constitutive law.

Table 3. Refining of the mesh in the nonlinear solid model. The simulations are run on the Shaheen cluster. procs = number of processors; dofs = degrees of freedom; T_{ass} = CPU time to assemble the stiffness and mass matrices; T_{coup} = CPU time to assemble the coupling term; nit = Newton iterations; its = GMRES iterations to solve the Jacobian system; T_{sol} = CPU time to solve the Jacobian system; T_{tot} = total simulation CPU time. The quantities T_{coup} and nit are averaged over the time steps, whereas the quantities its and T_{sol} are averaged over the Newton iterations and the time steps. All CPU times are reported in seconds.

Nonlinear Solid Model—Mesh Refinement Test											
procs = 64, T = 2, $\Delta t = 0.002$											
dofs	T_{ass} (s)	T_{coup} (s)	Block-diag				Block-tri				
			nit	its	T_{sol} (s)	T_{tot} (s)	nit	its	T_{sol} (s)	T_{tot} (s)	
21,222	4.04×10^{-3}	3.89×10^{-2}	2	11	4.29×10^{-1}	9.35×10^2	2	8	3.93×10^{-1}	8.64×10^2	
83,398	1.68×10^{-2}	3.60×10^{-1}	2	12	1.67	4.06×10^3	2	8	1.57	3.86×10^3	
186,534	3.80×10^{-2}	1.57	2	14	4.23	1.16×10^4	2	9	4.00	1.11×10^4	
330,630	6.68×10^{-2}	4.77	2	14	7.71	2.49×10^4	2	10	7.07	2.37×10^4	
515,686	1.05×10^{-1}	11.40	2	15	13.03	4.92×10^4	2	10	11.48	4.58×10^4	
741,702	1.52×10^{-1}	23.23	2	15	18.58	8.49×10^4	2	10	16.63	7.98×10^4	

In order to study the weak scalability, we choose $T = 0.1$ and $\Delta t = 0.002$. The results, reported in Table 4, are similar to the results obtained for the linear case. As before, T_{coup} is not scalable due to the algorithm we implemented for the assembling of the coupling term. Even if it is not perfectly scalable, block-tri performs pretty well since the average number of linear iterations per nonlinear iteration increases only from 15 to 19. On the other hand, the good behavior of block-diag registered in Table 3 is not confirmed: the average number of linear iterations reaches 101, showing a lack of weak scalability, as already seen in Table 2.

Table 4. Weak scalability for the nonlinear solid model. The simulations are run on the Shaheen cluster. Same format as Table 3.

Nonlinear Solid Model—Weak Scalability Test											
T = 0.1, $\Delta t = 0.002$											
procs	dofs	T_{ass} (s)	T_{coup} (s)	Block-diag				Block-tri			
				nit	its	T_{sol} (s)	T_{tot} (s)	nit	its	T_{sol} (s)	T_{tot} (s)
4	83,398	1.01×10^{-1}	2.21	3	23	6.76	448.65	3	15	5.50	386.13
8	156,910	1.59×10^{-1}	3.77	3	38	15.49	963.03	3	16	8.87	627.84
16	330,630	1.62×10^{-1}	8.92	3	49	36.58	2.28×10^3	3	17	17.84	1.34×10^3
32	741,702	2.60×10^{-1}	25.18	3	67	123.99	7.46×10^3	3	18	48.85	3.70×10^3
64	1,316,614	2.61×10^{-1}	69.12	3	101	328.18	1.99×10^4	3	19	97.28	8.38×10^3

7. Conclusions

We analyzed two preconditioners, block-diagonal and block-triangular, for saddle point systems originating from the finite element discretization of fluid–structure interaction problems with fictitious domain approach. We have focused only on the case where the fluid and solid domains have the same densities, which, based on previous studies, should be the most challenging to solve. In particular, the analysis has been conducted by studying the robustness with respect to mesh refinement and weak scalability, applying the parallel solver to both linear and nonlinear problems.

Only block-triangular appears to be robust in terms of mesh refinement for linear and nonlinear problems; on the other hand, block-diagonal works well when the time step is very small.

Moreover, by studying the weak scalability, we can notice two further limitations of the proposed method, which will be the subject of future studies. First, the time to assemble the coupling matrix is not scalable: it is based on two nested loops, related to solid and fluid elements, respectively; but only the external one is conducted in parallel over the processors. In order to improve this procedure, one may subdivide the involved meshes into clusters or, alternatively, one may keep track of the position of the solid body at the previous time instant. Second, since the action of the preconditioners consists of the exact inversion of two matrices, the time for solving the linear system slightly increases when the mesh is refined. Some preliminary results have shown that, in contrast with the fluid block \mathcal{A}_{11} , the solid block \mathcal{A}_{22} does not behave well when an inexact inversion is applied: our future studies will be primarily focused on overcoming this problem.

As concerns the modeling approach, several extensions are possible. First of all, we have neglected the convective term in the Navier–Stokes equation; second, we have considered an isotropic nonlinear constitutive law for the structure instead of anisotropic variants and, finally, we have focused only on two dimensional problems.

Author Contributions: Conceptualization, D.B. and L.G.; Methodology, S.S.; Software, F.C. and S.S.; Validation, F.C. and S.S.; Resources, D.B.; Writing—Original draft, F.C. and S.S.; Writing—Review & editing, F.C., D.B., L.G. and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors are member of GNCS/INdAM research group. D.B., F.C. and L.G. are partially supported by IMATI/CNR. Moreover, D.B. and L.G. are partially supported by PRIN/MIUR.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hirt, C.W.; Amsden, A.A.; Cook, J. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *J. Comput. Phys.* **1974**, *14*, 227–253. [[CrossRef](#)]
2. Donéa, J.; Fasoli-Stella, P.; Giuliani, S. Lagrangian and Eulerian finite element techniques for transient fluid–structure interaction problems. *Struct. Mech. React. Technol.* **1977**. Available online: <https://repository.lib.ncsu.edu/bitstream/handle/1840.20/27448/B1-2.pdf?sequence=1> (accessed on 23 January 2023).
3. Donea, J.; Huerta, A.; Ponthot, J.P.; Rodríguez-Ferran, A. Arbitrary Lagrangian–Eulerian Methods. *Encycl. Comput. Mech.* **2004**. [[CrossRef](#)]
4. Hughes, T.J.; Liu, W.K.; Zimmermann, T.K. Lagrangian–Eulerian finite element formulation for incompressible viscous flows. *Comput. Methods Appl. Mech. Eng.* **1981**, *29*, 329–349. [[CrossRef](#)]
5. Burman, E.; Fernández, M.A. An unfitted Nitsche method for incompressible fluid–structure interaction using overlapping meshes. *Comput. Methods Appl. Mech. Eng.* **2014**, *279*, 497–514. [[CrossRef](#)]
6. Chang, Y.C.; Hou, T.; Merriman, B.; Osher, S. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *J. Comput. Phys.* **1996**, *124*, 449–464. [[CrossRef](#)]

7. Glowinski, R.; Pan, T.W.; Periaux, J. A Lagrange multiplier/fictitious domain method for the numerical simulation of incompressible viscous flow around moving rigid bodies: (I) case where the rigid body motions are known a priori. *C. R. Acad. Sci. Ser. I Math.* **1997**, *324*, 361–369. [[CrossRef](#)]
8. Glowinski, R.; Pan, T.W.; Hesla, T.I.; Joseph, D.D.; Periaux, J. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow. *J. Comput. Phys.* **2001**, *169*, 363–426. [[CrossRef](#)]
9. Boffi, D.; Cavallini, N.; Gastaldi, L. The Finite Element Immersed Boundary Method with Distributed Lagrange Multiplier. *SIAM J. Numer. Anal.* **2015**, *53*, 2584–2604. [[CrossRef](#)]
10. Peskin, C.S. The immersed boundary method. *Acta Numer.* **2002**, *11*, 479–517. [[CrossRef](#)]
11. Boffi, D.; Gastaldi, L.; Heltai, L.; Peskin, C.S. On the hyper-elastic formulation of the immersed boundary method. *Comput. Methods Appl. Mech. Eng.* **2008**, *197*, 2210–2231. [[CrossRef](#)]
12. Balzani, D.; Deparis, S.; Fausten, S.; Forti, D.; Heinlein, A.; Klawonn, A.; Quarteroni, A.; Rheinbach, O.; Schroeder, J. Numerical modeling of fluid–structure interaction in arteries with anisotropic polyconvex hyperelastic and anisotropic viscoelastic material models at finite strains. *Int. J. Numer. Methods Biomed. Eng.* **2016**, *32*, e02756. [[CrossRef](#)] [[PubMed](#)]
13. Crosetto, P.; Deparis, S.; Fourestey, G.; Quarteroni, A. Parallel algorithms for fluid–structure interaction problems in haemodynamics. *SIAM J. Sci. Comput.* **2011**, *33*, 1598–1622. [[CrossRef](#)]
14. Deparis, S.; Forti, D.; Grandperrin, G.; Quarteroni, A. FaCSI: A block parallel preconditioner for fluid–structure interaction in hemodynamics. *J. Comput. Phys.* **2016**, *327*, 700–718. [[CrossRef](#)]
15. Barker, A.T.; Cai, X.C. Scalable parallel methods for monolithic coupling in fluid–structure interaction with application to blood flow modeling. *J. Comput. Phys.* **2010**, *229*, 642–659. [[CrossRef](#)]
16. 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. [[CrossRef](#)]
17. Heinlein, A.; Klawonn, A.; Rheinbach, O. A Parallel Implementation of a Two-Level Overlapping Schwarz Method with Energy-Minimizing Coarse Space Based on Trilinos. *SIAM J. Sci. Comput.* **2016**, *38*, C713–C747. [[CrossRef](#)]
18. Jodlbauer, D.; Langer, U.; Wick, T. Parallel block–preconditioned monolithic solvers for fluid–structure interaction problems. *Int. J. Numer. Methods Eng.* **2019**, *117*, 623–643. [[CrossRef](#)]
19. Krause, R.; Zulian, P. A parallel approach to the variational transfer of discrete fields between arbitrarily distributed unstructured finite element meshes. *SIAM J. Sci. Comput.* **2016**, *38*, C307–C333. [[CrossRef](#)]
20. Nestola, M.G.C.; Becsek, B.; Zolfaghari, H.; Zulian, P.; De Marinis, D.; Krause, R.; Obrist, D. An immersed boundary method for fluid–structure interaction based on variational transfer. *J. Comput. Phys.* **2019**, *398*, 108884. [[CrossRef](#)]
21. Boffi, D.; Credali, F.; Gastaldi, L.; Scacchi, S. A parallel solver for fluid structure interaction problems with Lagrange multiplier. *arXiv* **2022**, arXiv:2212.13410.
22. Boffi, D.; Credali, F.; Gastaldi, L. On the interface matrix for fluid–structure interaction problems with fictitious domain approach. *Comput. Methods Appl. Mech. Eng.* **2022**, *401*, 115650. [[CrossRef](#)]
23. Balay, S.; Abhyankar, S.; Adams, M.F.; Brown, J.; Brune, P.; Buschelman, K.; Dalcin, L.; Eijkhout, V.; Gropp, W.D.; Kaushik, D.; et al. PETSc Web Page. 2018. Available online: <http://www.mcs.anl.gov/petsc> (accessed on 3 December 2022).
24. Balay, S.; Abhyankar, S.; Adams, M.F.; Brown, J.; Brune, P.; Buschelman, K.; Dalcin, L.; Eijkhout, V.; Gropp, W.D.; Kaushik, D.; et al. *PETSc Users Manual*; Technical Report ANL-95/11—Revision 3.9; Argonne National Laboratory: Lemont, IL, USA, 2018.
25. Amestoy, P.R.; Duff, I.S.; L’Excellent, J.Y.; Koster, J. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.* **2001**, *23*, 15–41. [[CrossRef](#)]
26. Amestoy, P.R.; Guermouche, A.; L’Excellent, J.Y.; Pralet, S. Hybrid scheduling for the parallel solution of linear systems. *Parallel Comput.* **2006**, *32*, 136–156. [[CrossRef](#)]
27. Lions, J.L.; Magenes, E. *Non-Homogeneous Boundary Value Problems and Applications: Vol. 1*; Springer Science & Business Media: Berlin, Germany, 2012; Volume 181.
28. Boffi, D.; Gastaldi, L. A fictitious domain approach with Lagrange multiplier for fluid–structure interactions. *Numer. Math.* **2017**, *135*, 711–732. [[CrossRef](#)]
29. Peskin, C.S.; McQueen, D.M. A three-dimensional computational method for blood flow in the heart I. Immersed elastic fibers in a viscous incompressible fluid. *J. Comput. Phys.* **1989**, *81*, 372–405. [[CrossRef](#)]
30. Causin, P.; Gerbeau, J.; Nobile, F. Added-mass effect in the design of partitioned algorithms for fluid–structure problems. *Comput. Methods Appl. Mech. Eng.* **2005**, *194*, 4506–4527. [[CrossRef](#)]
31. Deparis, S.; Fernández, M.A.; Formaggia, L. Acceleration of a fixed point algorithm for fluid–structure interaction using transpiration conditions. *ESAIM Math. Model. Numer. Anal.* **2003**, *37*, 601–616. [[CrossRef](#)]
32. Badia, S.; Nobile, F.; Vergara, C. Fluid–structure partitioned procedures based on Robin transmission conditions. *J. Comput. Phys.* **2008**, *227*, 7027–7051. [[CrossRef](#)]
33. Alshehri, N.; Boffi, D.; Gastaldi, L. Unfitted mixed finite element methods for elliptic interface problems. *arXiv* **2022**, arXiv:2211.03443.
34. Delfino, A.; Stergiopoulos, N.; Moore, J., Jr.; Meister, J.J. Residual strain effects on the stress field in a thick wall finite element model of the human carotid bifurcation. *J. Biomech.* **1997**, *30*, 777–786. [[CrossRef](#)] [[PubMed](#)]

35. Holzapfel, G.A.; Gasser, T.C.; Ogden, R.W. A new constitutive framework for arterial wall mechanics and a comparative study of material models. *J. Elast. Phys. Sci. Solids* **2000**, *61*, 1–48.
36. Ogden, R.W. *Non-Linear Elastic Deformations*; Ellis Horwood Series: Mathematics and Its Applications; Ellis Horwood Ltd.: Chichester, UK; Halsted Press [John Wiley & Sons, Inc.]: New York, NY, USA, 1984; 532p.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.