

ON THE INTERFACE MATRIX FOR FLUID-STRUCTURE INTERACTION PROBLEMS WITH FICTITIOUS DOMAIN APPROACH

DANIELE BOFFI, FABIO CREDALI, AND LUCIA GASTALDI

ABSTRACT. We study a recent formulation for fluid-structure interaction problems based on the use of a distributed Lagrange multiplier in the spirit of the fictitious domain approach. In this paper, we focus our attention on a crucial computational aspect regarding the interface matrix for the finite element discretization: it involves integration of functions supported on two different meshes. Several numerical tests show that accurate computation of the interface matrix has to be performed in order to ensure the optimal convergence of the method.

Keywords: fluid-structure interaction, fictitious domain, non-matching grids, mesh intersection.

1. INTRODUCTION

The study of fluid-structure interaction problems, developed mainly from the second half of the last century, embraces various scientific fields, such as Mathematics, Physics and Engineering.

Methods for FSI problems can be divided into two large families.

Boundary-fitted approaches use a mesh for the fluid domain that deforms around a Lagrangian mesh for the structure; these two meshes match on the shared interface. A popular scheme of this category makes use of an arbitrary Lagrangian-Eulerian (ALE) coordinate system [21, 15, 16, 22]. The main advantage is that the kinematic constraints are satisfied by construction, but, on the other hand, strong distortions of the mesh could be obtained.

For this reason, the so-called non-boundary-fitted methods were subsequently introduced. In these cases a discretization of the solid structure is immersed in the fluid mesh. The disadvantage that arises is a reduction in accuracy near the interface between fluid and structure. This family includes several methods, developed to solve problems of various kinds since there is no method that is optimal for all situations. Among them, we have the level set formulation [14] for incompressible, immiscible Navier-Stokes equations separated by a free surface and the Nitsche-XFEM [1, 13], introduced to study thin-walled elastic structures immersed in an incompressible fluid.

The immersed boundary method [26, 25] is both a mathematical formulation and a numerical scheme. Mathematically, Eulerian and Lagrangian variables are linked making use of interaction equations involving Dirac delta functions. The numerical scheme consists in

2020 *Mathematics Subject Classification.* 65N30, 65N12, 74F10.

defining the Eulerian variables on a fixed Cartesian mesh and the Lagrangian variables on the reference domain. Finally, the fictitious domain method [19, 20] was originally introduced to study the particulate flow: indeed, the fluid domain is artificially extended to include also particles, which are considered as solid bodies. With this methodology it is not necessary to generate a new mesh at each time step: the solid body is represented making use of a reference configuration, which is mapped at each time step to the actual position.

To develop our model, we started from a formulation based on the immersed boundary method [10] and we moved towards the fictitious domain approach. In the case of the original finite element immersed boundary method, the evolution of the structure is governed by an ordinary differential equation that involves the position of the solid and the velocity of the fluid. Starting from here, we then introduced a Lagrange multiplier with the effect that the movement of the structure is managed through a bilinear form that contains the multiplier. As part of our research, several theoretical aspects have been addressed over the years: in [7], we showed the well-posedness of both continuous and discrete problems, in [9] the existence and the uniqueness of the solution were proved in the case of the linearized problem. These results are summarized in a unified setting in [8]. On the other hand, numerical aspects have been studied only from the point of view of time discretization in [11]. We are dealing with a monolithic scheme, based on solving the coupled system without separating the equations for fluid and structure and characterized by the fact that the solid and the fluid meshes are totally independent of each other. From the computational point of view, the assembling of the interface matrix is based on integrations over the solid mesh involving also the fluid basis functions. This operation can be carried out in two different ways: the first one is based on the computation of the intersection between the two meshes, with the aim of implementing a composite quadrature rule, on the other hand, the second one works directly on the solid elements without involving preliminary geometric computations.

The computation of intersection between immersed meshes has been addressed in several papers only from the computational point of view as a support for the simulation of fluid-structure interactions or contact mechanics without emphasizing why such a procedure is necessary. For instance, in [23], the goal is to develop an efficient parallel approach for the variational transfer of information between two meshes, using the computation of the intersection as starting point; in [17], the Sutherland-Hodgman algorithm for clipping polygons is used in support of Galerkin projection for conservative interpolation of discrete fields; finally, in [12], fast algorithms for the intersection of simplicial elements of different dimensions in the three-dimensional space are presented as a support for XFEM and Mortar methods.

Our aim is to show how this expensive geometric procedure is necessary in order to reach the optimal convergence rate of the method. Indeed, we are going to show that skipping the computation of the mesh intersection, we cannot reach the optimal convergence even if we increase the precision of the used quadrature rule.

The structure of this article is the following one. In Section 2, we summarize the theoretical background of our fluid-structure interaction problems starting from the Navier-Stokes equation and emphasizing the fictitious domain framework. In particular, we focus our attention on the case of a thick structure. Next, we briefly recall the time semi-discretization

and its connection with stationary saddle-point problems that can be studied numerically making use of the finite element method. In Section 5, we describe how the coupling matrix can be assembled with the two methods mentioned above, paying also attention on algorithmic aspects. Finally, in Section 6, we report several numerical tests showing how the choice of the procedure for assembling the interface matrix affects the convergence of our method.

2. FICTITIOUS DOMAIN APPROACH: PROBLEM SETTING

Let us consider the case of a fluid-structure interaction where a solid elastic body is immersed in a two or three dimensional fluid. We denote by Ω_t^f and Ω_t^s the time dependent regions in \mathbb{R}^d (with $d = 2, 3$) occupied respectively by the fluid and the solid at the time instant t . In particular, both cases of a solid of codimension zero and one are allowed in our model, even if in this paper we are going to consider only the first case. We denote by Ω the union between Ω_t^f and Ω_t^s such that it is not dependent on time, assuming also that the solid interface $\partial\Omega_t^s$ cannot touch the boundary of $\partial\Omega$, i.e. $\partial\Omega \cap \partial\Omega_t^s = \emptyset$. Moreover, the solid is represented making use of a fixed reference domain \mathcal{B} which is mapped at each time instant to the actual domain Ω_t^s through a mapping $\mathbf{X} : \mathcal{B} \rightarrow \Omega_t^s$.

The mapping \mathbf{X} represents the motion of the solid, while the fluid configuration is represented by the velocity \mathbf{u}^f and by the pressure p^f . The solid unknown is time dependent and is defined making use of a Lagrangian variable $\mathbf{s} \in \mathcal{B}$, while the fluid unknowns are functions of time and of the Eulerian variable \mathbf{x} . In this setting, each point $\mathbf{x} \in \Omega_t^s$ can be represented in terms of the Lagrangian variable

$$\mathbf{x} = \mathbf{X}(\mathbf{s}, t)$$

and the motion of the solid body is expressed by the kinematic equation

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

On the other hand, using the derivative with respect to \mathbf{s} , we can introduce the deformation gradient $\mathbb{F} = \nabla_{\mathbf{s}} \mathbf{X}$ and its determinant $|\mathbb{F}|$; since we are going to consider the case of incompressible solid materials, we have that $|\mathbb{F}|$ is constant in time and when the reference domain \mathcal{B} corresponds to the initial position Ω_0^s of the structure, we get $|\mathbb{F}| = 1$.

For the fluid, we denote by ρ_f the density and by $\boldsymbol{\sigma}^f$ the Cauchy stress tensor defined, using the viscosity $\nu_f > 0$ and the symmetric gradient $\underline{\boldsymbol{\varepsilon}}$, as

$$\boldsymbol{\sigma}^f = -p^f \mathbb{I} + \nu_f \underline{\boldsymbol{\varepsilon}}(\mathbf{u}^f);$$

these objects are involved in the incompressible Navier-Stokes equations which describe the fluid dynamic in Ω_t^f as follows

$$(2) \quad \begin{aligned} \rho_f \left(\frac{\partial \mathbf{u}^f}{\partial t} + \mathbf{u}^f \cdot \nabla \mathbf{u}^f \right) &= \text{div } \boldsymbol{\sigma}^f \\ \text{div } \mathbf{u}^f &= 0. \end{aligned}$$

For the solid, we assume that the material is incompressible but also viscoelastic, hence we describe it with a Cauchy stress tensor made of two contributions $\boldsymbol{\sigma}^s = \boldsymbol{\sigma}_f^s + \boldsymbol{\sigma}_s^s$: in details, $\boldsymbol{\sigma}_f^s$ is a term analogous to the fluid stress $\boldsymbol{\sigma}^f$ but defined using the pressure p^s which is the Lagrange multiplier associated with the incompressibility condition, while $\boldsymbol{\sigma}_s^s$ depends on the Piola-Kirchhoff elasticity stress tensor \mathbb{P} via the Piola transformation; we have respectively

$$\begin{aligned}\boldsymbol{\sigma}_f^s &= -p^s \mathbb{I} + \nu_s \underline{\boldsymbol{\epsilon}}(\mathbf{u}^s) \\ \boldsymbol{\sigma}_s^s &= |\mathbb{F}|^{-1} \mathbb{P} \mathbb{F}^T\end{aligned}$$

where we denote by $\nu_s > 0$ the body viscosity. Furthermore, a potential energy density $W(\mathbb{F}, \mathbf{s}, t)$ can be used for modeling the elastic part of the stress; we have that

$$\mathbb{P}(\mathbb{F}, \mathbf{s}, t) = \frac{\partial W}{\partial \mathbb{F}}(\mathbb{F}, \mathbf{s}, t).$$

At this point, we have all the ingredients we need to express the equations representing the solid behavior, indeed denoting by ρ_s the solid density, we can write

$$(3) \quad \begin{aligned}\rho_s \frac{\partial^2 \mathbf{X}}{\partial t^2} &= \operatorname{div}_{\mathbf{s}} (|\mathbb{F}| \boldsymbol{\sigma}_f^s \mathbb{F}^{-T} + \mathbb{P}(\mathbb{F})) && \text{in } \mathcal{B} \\ \operatorname{div} \mathbf{u}^s &= 0 && \text{in } \Omega_t^s.\end{aligned}$$

To complete the model we need to be careful with respect to the behavior of our system on the interface Γ_t and prescribe initial and boundary conditions. In particular, we need to enforce continuity of velocity and Cauchy stress, hence we impose the following transmission conditions

$$(4) \quad \begin{aligned}\mathbf{u}^f &= \mathbf{u}^s && \text{on } \partial\Omega_t^s \\ \boldsymbol{\sigma}_f \mathbf{n}_f &= -(\boldsymbol{\sigma}_f^s + |\mathbb{F}|^{-1} \mathbb{P} \mathbb{F}^T) \mathbf{n}_s && \text{on } \partial\Omega_t^s\end{aligned}$$

where \mathbf{n}_s and \mathbf{n}_f denote the outer unit normal vectors to Ω_t^s and Ω_t^f , respectively. Finally, we choose the following conditions

$$(5) \quad \begin{aligned}\mathbf{u}^f(0) &= \mathbf{u}_0^f && \text{in } \Omega_0^f \\ \mathbf{u}^s(0) &= \mathbf{u}_0^s && \text{in } \Omega_0^s \\ \mathbf{X}(0) &= \mathbf{X}_0 && \text{in } \mathcal{B} \\ \mathbf{u}^f &= 0 && \text{on } \partial\Omega.\end{aligned}$$

We now recall some standard notations in functional analysis [24] we are going to use in the following. Given a domain D , we denote by $L^2(D)$ the space of square integrable functions on D , while for the standard Sobolev spaces we use $W^{s,q}(D)$, where $s \in \mathbb{R}$ indicates the differentiability and $q \in [1, \infty]$ the integrability exponent. When $q = 2$, the usual notation $H^s(D)$ is used. Furthermore, $L_0^2(D)$ is the subspace of zero mean valued functions, while $H_0^1(D)$ is the subset of $H^1(D)$ with zero trace on the boundary of D . The norm in $H^s(D)$ is denoted by $\|\cdot\|_{s,D}$ and $(\cdot, \cdot)_D$ stands for the scalar product of $L^2(D)$. The indication of the domain will be omitted when no confusion arises; in particular we will omit Ω , while we will indicate explicitly when quantities are defined on \mathcal{B} .

Once we have defined the model for both fluid and structure, we can introduce the fictitious domain approach basically extending the fluid variables also in the region occupied by the solid. Hence, we call \mathbf{u} and p velocity and pressure defined on the whole Ω with

$$(6) \quad \mathbf{u} = \begin{cases} \mathbf{u}^f & \text{in } \Omega_t^f \\ \mathbf{u}^s & \text{in } \Omega_t^s \end{cases}, \quad p = \begin{cases} p^f & \text{in } \Omega_t^f \\ p^s & \text{in } \Omega_t^s \end{cases}.$$

Moreover, we take into account that the material velocity of the solid is equal to the velocity of the fictitious fluid, that is

$$(7) \quad \frac{\partial \mathbf{X}}{\partial t}(\mathbf{s}, t) = \mathbf{u}(\mathbf{X}(\mathbf{s}, t), t) \quad \text{for } \mathbf{s} \in \mathcal{B};$$

this condition, analogous to (1), is now a constraint for the velocity \mathbf{u} , describing its behavior in Ω_t^s . In variational terms, condition (7) is going to be enforced introducing a distributed Lagrange multiplier. For this purpose, we consider a Hilbert space $\mathbf{\Lambda}$ and we define a continuous bilinear form $\mathbf{c} : \mathbf{\Lambda} \times (H^1(\mathcal{B}))^d \rightarrow \mathbb{R}$ satisfying the property

$$(8) \quad \mathbf{c}(\boldsymbol{\mu}, \mathbf{Z}) = 0 \quad \forall \boldsymbol{\mu} \in \mathbf{\Lambda} \quad \Rightarrow \quad \mathbf{Z} = 0.$$

There are several choices for the form \mathbf{c} ; in particular, we can choose \mathbf{c} to be the duality pairing between $(H^1(\mathcal{B}))^d$ and its dual $\mathbf{\Lambda} = ((H^1(\mathcal{B}))^d)'$

$$(9) \quad \mathbf{c}(\boldsymbol{\mu}, \mathbf{Y}) = \langle \boldsymbol{\mu}, \mathbf{Y} \rangle \quad \forall \boldsymbol{\mu} \in ((H^1(\mathcal{B}))^d)', \quad \forall \mathbf{Y} \in (H^1(\mathcal{B}))^d$$

or, alternatively, we can set $\mathbf{\Lambda} = (H^1(\mathcal{B}))^d$ and define \mathbf{c} as the scalar product

$$(10) \quad \mathbf{c}(\boldsymbol{\mu}, \mathbf{Y}) = (\boldsymbol{\mu}, \mathbf{Y})_{\mathcal{B}} + (\nabla_s \boldsymbol{\mu}, \nabla_s \mathbf{Y})_{\mathcal{B}} \quad \forall \boldsymbol{\mu}, \mathbf{Y} \in (H^1(\mathcal{B}))^d.$$

At the end, we can write the variational formulation of our problem; before doing that, we introduce the following notations

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \nu \underline{\boldsymbol{\varepsilon}}(\mathbf{u}) : \underline{\boldsymbol{\varepsilon}}(\mathbf{v}) \, d\mathbf{x} \quad \nu = \begin{cases} \nu_f & \text{in } \Omega_t^f \\ \nu_s & \text{in } \Omega_t^s \end{cases}$$

$$b(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \frac{\rho_f}{2} ((\mathbf{u} \cdot \nabla \mathbf{v}, \mathbf{w}) - (\mathbf{u} \cdot \nabla \mathbf{w}, \mathbf{v})) \quad \delta \rho = \rho_s - \rho_f.$$

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

$$\begin{aligned}
 (11a) \quad & \rho_f \frac{\partial}{\partial t} (\mathbf{u}(t), \mathbf{v}) + b(\mathbf{u}(t), \mathbf{u}(t), \mathbf{v}) + a(\mathbf{u}(t), \mathbf{v}) \\
 & - (\operatorname{div} \mathbf{v}, p(t)) + \mathbf{c}(\boldsymbol{\lambda}(t), \mathbf{v}(\mathbf{X}(t))) = 0 \quad \forall \mathbf{v} \in (H_0^1(\Omega))^d \\
 (11b) \quad & (\operatorname{div} \mathbf{u}(t), q) = 0 \quad \forall q \in L_0^2(\Omega) \\
 (11c) \quad & \delta \rho \left(\frac{\partial^2 \mathbf{X}}{\partial t^2}, \mathbf{Y} \right)_{\mathcal{B}} + (\mathbb{P}(\mathbb{F}(\mathbf{s}, t)), \nabla_{\mathbf{s}} \mathbf{Y})_{\mathcal{B}} - \mathbf{c}(\boldsymbol{\lambda}(t), \mathbf{Y}) = 0 \quad \forall \mathbf{Y} \in (H^1(\mathcal{B}))^d \\
 (11d) \quad & \mathbf{c} \left(\boldsymbol{\mu}, \mathbf{u}(\mathbf{X}(\cdot, t), t) - \frac{\partial \mathbf{X}}{\partial t}(t) \right) = 0 \quad \forall \boldsymbol{\mu} \in \boldsymbol{\Lambda} \\
 (11e) \quad & \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega \\
 (11f) \quad & \mathbf{X}(\mathbf{s}, 0) = \mathbf{X}_0(\mathbf{s}) \quad \forall \mathbf{s} \in \mathcal{B}
 \end{aligned}$$

In [9], it was shown that a linearized version of Problem 1 admits a unique solution. See also [8] for a review on the state of the art about this problem.

3. TIME ADVANCING SCHEME AND ASSOCIATED STATIONARY PROBLEM

In this section we recall the formulation of our problem when it is discretized in time via backward finite differences; in particular, we are going to see that at each time step, the solution of the problem is reduced to a stationary saddle point problem.

Let us consider a positive integer N and partition the time interval $[0, T]$ into N equal parts so that each node corresponds to $t_n = n\Delta t$, for $n = 0, \dots, N$, where $\Delta t = T/N$ is the time step size. Hence, the approximation of the derivatives reads as follows

$$(12) \quad \partial_t \mathbf{v}(t_{n+1}) \approx \frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\Delta t}, \quad \partial_{tt} \mathbf{v}(t_{n+1}) \approx \frac{\mathbf{v}^{n+1} - 2\mathbf{v}^n + \mathbf{v}^{n-1}}{\Delta t^2}.$$

As a consequence, the time discretized version of Problem 1 can be written in the following manner.

Problem 2. Given $\mathbf{u}_0 \in (H_0^1(\Omega))^d$ and $\mathbf{X}_0 \in (W^{1,\infty}(\mathcal{B}))^d$, for $n = 1, \dots, N$ find $(\mathbf{u}^n, p^n) \in (H_0^1(\Omega))^d \times L_0^2(\Omega)$, $\mathbf{X}^n \in (H^1(\mathcal{B}))^d$, and $\boldsymbol{\lambda}^n \in \boldsymbol{\Lambda}$, such that

$$(13a) \quad \rho_f \left(\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t}, \mathbf{v} \right) + b(\mathbf{u}^n, \mathbf{u}^{n+1}, \mathbf{v}) + a(\mathbf{u}^{n+1}, \mathbf{v}) - (\operatorname{div} \mathbf{v}, p^{n+1}) + \mathbf{c}(\boldsymbol{\lambda}^{n+1}, \mathbf{v}(\mathbf{X}^n)) = 0 \quad \forall \mathbf{v} \in (H_0^1(\Omega))^d$$

$$(13b) \quad (\operatorname{div} \mathbf{u}^{n+1}, q) = 0 \quad \forall q \in L_0^2(\Omega)$$

$$(13c) \quad \delta \rho \left(\frac{\mathbf{X}^{n+1} - 2\mathbf{X}^n + \mathbf{X}^{n-1}}{\Delta t^2}, \mathbf{Y} \right)_{\mathcal{B}} + (\mathbb{P}(\mathbb{F}^{n+1}), \nabla_s \mathbf{Y})_{\mathcal{B}} - \mathbf{c}(\boldsymbol{\lambda}^{n+1}, \mathbf{Y}) = 0 \quad \forall \mathbf{Y} \in (H^1(\mathcal{B}))^d$$

$$(13d) \quad \mathbf{c} \left(\boldsymbol{\mu}, \mathbf{u}^{n+1}(\mathbf{X}^n) - \frac{\mathbf{X}^{n+1} - \mathbf{X}^n}{\Delta t} \right) = 0 \quad \forall \boldsymbol{\mu} \in \boldsymbol{\Lambda}$$

where we can define \mathbf{X}^{-1} via an equation of the following type

$$\frac{\mathbf{X}^0 - \mathbf{X}^{-1}}{\Delta t} = \mathbf{u}_0^s \quad \text{in } \mathcal{B}.$$

As we were saying before, at each time step of Problem 2, we solve a saddle point problem which is independent of time: we refer to [7] for its analysis, in terms of inf-sup conditions and convergence.

In order to present the structure of the stationary problem we are going to solve, we first reduce our discussion to the case of a linear model of the Piola-Kirchhoff stress tensor, that we define as $\mathbb{P}(\mathbb{F}) = \kappa \mathbb{F} = \kappa \nabla_s \mathbf{X}$. Therefore, we get the following saddle point problem.

Problem 3. Let $\bar{\mathbf{X}} \in (W^{1,\infty}(\mathcal{B}))^d$ be invertible with Lipschitz inverse and $\bar{\mathbf{u}} \in (H_0^1(\Omega))^d$ such that $\operatorname{div} \bar{\mathbf{u}} = 0$. Given $\mathbf{f} \in (L^2(\Omega))^d$, $\mathbf{g} \in (L^2(\mathcal{B}))^d$ and $\mathbf{d} \in (L^2(\mathcal{B}))^d$, find $(\mathbf{u}, p) \in (H_0^1(\Omega))^d \times L_0^2(\Omega)$, $\mathbf{X} \in (H^1(\mathcal{B}))^d$ and $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$, such that

$$(14a) \quad \mathbf{a}_f(\mathbf{u}, \mathbf{v}) - (\operatorname{div} \mathbf{v}, p) + \mathbf{c}(\boldsymbol{\lambda}, \mathbf{v}(\bar{\mathbf{X}})) = (\mathbf{f}, \mathbf{v}) \quad \forall \mathbf{v} \in (H_0^1(\Omega))^d$$

$$(14b) \quad (\operatorname{div} \mathbf{u}, q) = 0 \quad \forall q \in L_0^2(\Omega)$$

$$(14c) \quad \mathbf{a}_s(\mathbf{X}, \mathbf{Y}) - \mathbf{c}(\boldsymbol{\lambda}, \mathbf{Y}) = (\mathbf{g}, \mathbf{Y}) \quad \forall \mathbf{Y} \in (H^1(\mathcal{B}))^d$$

$$(14d) \quad \mathbf{c}(\boldsymbol{\mu}, \mathbf{u}(\bar{\mathbf{X}}) - \mathbf{X}) = \mathbf{c}(\boldsymbol{\mu}, \mathbf{d}) \quad \forall \boldsymbol{\mu} \in \boldsymbol{\Lambda}$$

Particularly, the bilinear forms \mathbf{a}_f and \mathbf{a}_s are defined as

$$(15) \quad \begin{aligned} \mathbf{a}_f(\mathbf{u}, \mathbf{v}) &= \alpha(\mathbf{u}, \mathbf{v}) + a(\mathbf{u}, \mathbf{v}) + b(\bar{\mathbf{u}}, \mathbf{u}, \mathbf{v}) \quad \forall \mathbf{u}, \mathbf{v} \in (H_0^1(\Omega))^d \\ \mathbf{a}_s(\mathbf{X}, \mathbf{Y}) &= \beta(\mathbf{X}, \mathbf{Y})_{\mathcal{B}} + \gamma(\nabla_s \mathbf{X}, \nabla_s \mathbf{Y})_{\mathcal{B}} \quad \forall \mathbf{X}, \mathbf{Y} \in (H^1(\mathcal{B}))^d, \end{aligned}$$

where $\bar{\mathbf{X}} = \mathbf{X}^n$, $\bar{\mathbf{u}} = \mathbf{u}^n$ in the nonlinear terms. Problem 3 is obtained from Problem 2 considering a single step and taking

$$\begin{aligned}\mathbf{u} &= \mathbf{u}^{n+1}, p = p^{n+1}, \mathbf{X} = \frac{\mathbf{X}^{n+1}}{\Delta t}, \boldsymbol{\lambda} = \boldsymbol{\lambda}^{n+1} \\ \mathbf{f} &= \frac{\rho_f}{\Delta t} \mathbf{u}^n \\ \mathbf{g} &= \frac{\delta \rho}{\Delta t^2} (2\mathbf{X}^n - \mathbf{X}^{n-1}) \\ \mathbf{d} &= -\frac{1}{\Delta t} \mathbf{X}^n \\ \alpha &= \frac{\rho_f}{\Delta t}, \beta = \frac{\delta \rho}{\Delta t}, \gamma = \kappa \Delta t.\end{aligned}$$

4. FINITE ELEMENT DISCRETIZATION OF THE STATIONARY PROBLEM

In order to discretize Problem 3 using finite elements, we need to consider four finite dimensional subspaces; in detail $\mathbf{V}_h \subset (H_0^1(\Omega))^d$, $Q_h \subset L_0^2(\Omega)$, $\mathbf{S}_h \subset (H^1(\mathcal{B}))^d$ and $\boldsymbol{\Lambda}_h \subset \boldsymbol{\Lambda}$, where the spaces for velocity and pressure, \mathbf{V}_h and Q_h have to satisfy the inf-sup conditions for the Stokes problem. In addition, we reduce the discussion to the case when the two spaces on the solid, \mathbf{S}_h and $\boldsymbol{\Lambda}_h$, coincide.

We observe that if \mathbf{c} is defined as the duality pairing between $(H^1(\mathcal{B}))^d$ and its dual, we can identify it with the scalar product in $(L^2(\mathcal{B}))^d$ providing that $\boldsymbol{\Lambda}_h$ is included in $(L^2(\mathcal{B}))^d$:

$$(16) \quad \mathbf{c}(\boldsymbol{\mu}, \mathbf{Y}) = (\boldsymbol{\mu}, \mathbf{Y})_{\mathcal{B}} \quad \forall \boldsymbol{\mu} \in \boldsymbol{\Lambda}_h, \forall \mathbf{Y} \in \mathbf{S}_h.$$

On the other hand, if \mathbf{c} is defined as the inner product of $(H^1(\mathcal{B}))^d$, no modification is necessary.

These finite dimensional spaces are defined making use of two meshes for the spatial discretization of the domains Ω and \mathcal{B} : we denote by \mathcal{T}_h the spatial discretization of Ω characterized by the size $h_{\mathcal{T}}$, while \mathcal{S}_h is the mesh we use for \mathcal{B} , with size $h_{\mathcal{S}}$.

In this setting, we are now able to state the finite element version of Problem 3 whose well-posedness has been discussed in [7] and [8].

Problem 4. Let $\bar{\mathbf{X}} \in (W^{1,\infty}(\mathcal{B}))^d$ be invertible with Lipschitz inverse and $\bar{\mathbf{u}} \in (H_0^1(\Omega))^d$ such that $\operatorname{div} \bar{\mathbf{u}} = 0$. Given $\mathbf{f} \in (L^2(\Omega))^d$, $\mathbf{g} \in (L^2(\mathcal{B}))^d$ and $\mathbf{d} \in (L^2(\mathcal{B}))^d$, find $(\mathbf{u}_h, p_h) \in \mathbf{V}_h \times Q_h$, $\mathbf{X}_h \in \mathbf{S}_h$ and $\boldsymbol{\lambda}_h \in \boldsymbol{\Lambda}_h$, such that

$$(17a) \quad \mathbf{a}_f(\mathbf{u}_h, \mathbf{v}_h) - (\operatorname{div} \mathbf{v}_h, p_h) + \mathbf{c}(\boldsymbol{\lambda}_h, \mathbf{v}_h(\bar{\mathbf{X}})) = (\mathbf{f}, \mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathbf{V}_h$$

$$(17b) \quad (\operatorname{div} \mathbf{u}_h, q_h) = 0 \quad \forall q_h \in Q_h$$

$$(17c) \quad \mathbf{a}_s(\mathbf{X}_h, \mathbf{Y}_h) - \mathbf{c}(\boldsymbol{\lambda}_h, \mathbf{Y}_h) = (\mathbf{g}, \mathbf{Y}_h) \quad \forall \mathbf{Y}_h \in \mathbf{S}_h$$

$$(17d) \quad \mathbf{c}(\boldsymbol{\mu}_h, \mathbf{u}_h(\bar{\mathbf{X}}) - \mathbf{X}_h) = \mathbf{c}(\boldsymbol{\mu}_h, \mathbf{d}) \quad \forall \boldsymbol{\mu}_h \in \boldsymbol{\Lambda}_h.$$

Moreover, this can be rewritten in matrix form as

$$(18) \quad \left[\begin{array}{cc|cc} A_f & B^\top & 0 & C_f^\top \\ B & 0 & 0 & 0 \\ \hline 0 & 0 & A_s & -C_s^\top \\ C_f & 0 & -C_s & 0 \end{array} \right] \begin{bmatrix} \mathbf{u} \\ p \\ \mathbf{X} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \\ \mathbf{g} \\ \mathbf{d} \end{bmatrix},$$

where the blocks are defined using the basis functions of \mathbf{V}_h , Q_h , \mathbf{S}_h and $\boldsymbol{\Lambda}_h$: indeed, denoting the basis functions respectively with $\boldsymbol{\phi}$, ψ , $\boldsymbol{\chi}$ and $\boldsymbol{\zeta}$, we get

$$(A_f)_{ij} = \mathbf{a}_f(\boldsymbol{\phi}_j, \boldsymbol{\phi}_i)$$

$$B_{ki} = -(\operatorname{div} \boldsymbol{\phi}_i, \psi_k)$$

$$(A_s)_{ij} = \mathbf{a}_s(\boldsymbol{\chi}_j, \boldsymbol{\chi}_i)$$

$$(C_f)_{\ell j} = \mathbf{c}(\boldsymbol{\zeta}_\ell, \boldsymbol{\phi}_j(\bar{\mathbf{X}}))$$

$$(C_s)_{\ell j} = \mathbf{c}(\boldsymbol{\zeta}_\ell, \boldsymbol{\chi}_j).$$

5. ASSEMBLING THE INTERFACE MATRIX C_f

The C_f block of the stiffness matrix of the finite element discretization (18) plays a crucial role in our method because it represents the interaction between the fluid and the structure. In particular, since we use a fictitious domain approach, the fluid domain is extended in order to include also the region occupied by the solid, hence there is an overlap of the fluid mesh with the solid mesh mapped into fluid domain.

The matrix C_f is associated with the bilinear form $\mathbf{c}(\boldsymbol{\mu}_h, \mathbf{v}_h(\bar{\mathbf{X}}))$, which can be the scalar product of $(L^2(\mathcal{B}))^d$ or $(H^1(\mathcal{B}))^d$. Therefore, we have to compute an integral over the solid reference domain \mathcal{B} that involves $\boldsymbol{\mu}_h \in \boldsymbol{\Lambda}_h$, defined on \mathcal{B} , and $\mathbf{v}_h \in \mathbf{V}_h$ defined on the whole domain Ω .

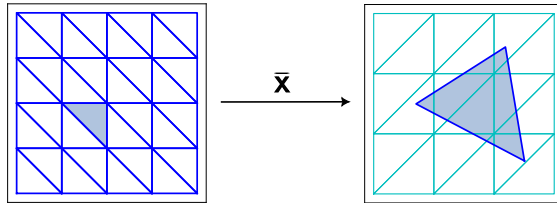


FIGURE 1. Mapping of a solid element into the fluid mesh.

In Figure 1, we show a portion of the meshes \mathcal{S}_h of \mathcal{B} and \mathcal{T}_h of Ω . In the picture we highlight in gray a particular triangle $T \in \mathcal{S}_h$ and the corresponding element $\bar{\mathbf{X}}(T)$ immersed

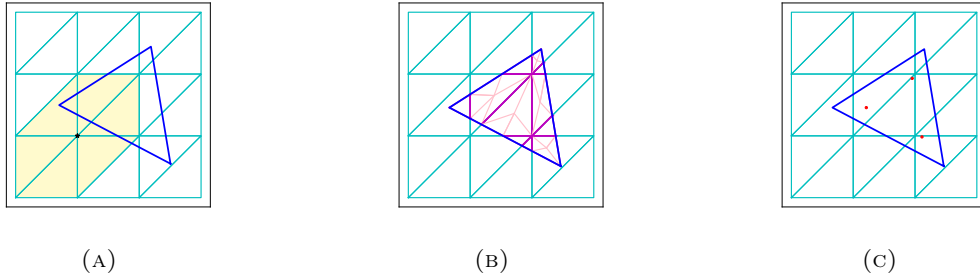


FIGURE 2. A graphical example for the two approaches proposed for the assembling of the interface matrix. The cyan mesh is related to the velocity while the big triangle is a mapped element in its actual position. (a) The support of the fluid basis function associated with the black-starred point is represented in yellow. In particular, it is clear that only a portion of the support is included in the solid element. (b) The mapped solid triangle is partitioned into sub-polygons: each of them is related to a single velocity element. If a polygon is not already a triangle, it is triangulated (pink lines). (c) Mapped nodes for a Gauss quadrature rule with order two.

in Ω . Basically, to map the mesh \mathcal{S}_h of \mathcal{B} into Ω , we simply apply the transformation $\bar{\mathbf{X}}$ to the nodes, keeping linear the edges connecting them. As a consequence, how to manage the geometrical aspects becomes really important because we have to integrate over \mathcal{S}_h the velocity shape functions defined on \mathcal{T}_h , combined with $\bar{\mathbf{X}}$. In doing that, we have to take into account the position occupied by $\bar{\mathbf{X}}(\mathcal{S}_h)$, see Figure 2a for an example of mismatch between the supports of fluid and mapped solid basis functions: indeed, the support of the fluid basis function under consideration, indicated in yellow, only partially matches the blue mapped solid triangle. We also notice that this situation does not concern the assembling of C_s , also defined through \mathbf{c} , but involving the spaces \mathbf{S}_h and $\mathbf{\Lambda}_h$ both defined on \mathcal{B} .

In this section, we restrict our discussion to the case $d = 2$ with triangular meshes in order to present two different approaches for the assembling procedure of C_f : the first approach depends on the computation of the intersection between the velocity mesh and the mesh of \mathcal{B} mapped into Ω^s , while the second one skips these geometric computations.

At this point, we introduce the discrete version of the two choices for the bilinear form \mathbf{c} with respect to the mesh \mathcal{S}_h ; if \mathbf{c} is defined like in (16), we have

$$(19) \quad \mathbf{c}_h(\boldsymbol{\mu}_h, \mathbf{v}_h(\bar{\mathbf{X}})) = \int_{\mathcal{S}_h} \boldsymbol{\mu}_h \cdot \mathbf{v}_h(\bar{\mathbf{X}}) \, ds = \sum_{T_s \in \mathcal{S}_h} \int_{T_s} \boldsymbol{\mu}_h \cdot \mathbf{v}_h(\bar{\mathbf{X}}) \, ds$$

otherwise, if the definition is the one in (10), we can write

$$\begin{aligned}
 \mathbf{c}_h(\boldsymbol{\mu}_h, \mathbf{v}_h(\bar{\mathbf{X}})) &= \int_{\mathcal{S}_h} \boldsymbol{\mu}_h \cdot \mathbf{v}_h(\bar{\mathbf{X}}) + \nabla_{\mathbf{s}} \boldsymbol{\mu}_h : \nabla_{\mathbf{s}} \mathbf{v}_h(\bar{\mathbf{X}}) \, \mathrm{d}\mathbf{s} \\
 (20) \qquad \qquad \qquad &= \sum_{T_s \in \mathcal{S}_h} \int_{T_s} \boldsymbol{\mu}_h \cdot \mathbf{v}_h(\bar{\mathbf{X}}) + \nabla_{\mathbf{s}} \boldsymbol{\mu}_h : \nabla_{\mathbf{s}} \mathbf{v}_h(\bar{\mathbf{X}}) \, \mathrm{d}\mathbf{s}.
 \end{aligned}$$

In the next subsections, we expand these two definitions, denoting by $\{(\mathbf{p}_k, \omega_k)\}_{k=1}^K$ nodes and weights respectively of the quadrature rule under consideration; $|T|$ denotes the area of a triangle T . We remark that $\nabla_{\mathbf{s}} \mathbf{v}_h(\bar{\mathbf{X}})$ is the gradient of a composite function, indeed we have to compute the gradient with respect to \mathbf{s} of the velocity \mathbf{v}_h applied to the map $\bar{\mathbf{X}}$, defined on \mathcal{B} .

5.1. Assembling with mesh intersection. If we want to perform an exact computation of the integrals in (19) and (20), we need to use a composite rule on the intersection of the fluid and solid meshes.

The aim of computing the intersection is to obtain a new triangulation for the structure, finer than the original one, such that each new element is immersed in (i.e. interacts with) a single fluid element: in this way, all the basis functions involved for the integration (both solid and fluid) are supported in the structure element under consideration. With this procedure, we can obtain an accurate computation of the interaction because, thanks to the composite quadrature rule defined, we can take fully into account the contribution given by the objects involved.

In particular, the computation of the intersection has to be made testing each actual solid element $\bar{\mathbf{X}}(T_s)$ with each fluid element $T_f \in \mathcal{T}_h$; the intersection could be still a triangle or a general polygon: in the first case, we have already a new element of the finer triangulation while, in the latter one, we compute a sub-triangulation connecting the barycenter with the vertices of the polygon. In particular, we denote by $\{P_j\}_{j=1}^J$ the set of the resulting polygons for each T_s and by $\{T_i\}_{i=1}^{I_j}$ the elements of the new triangulation of P_j . Therefore, the numerical version of the integrals in (19) is given by

$$\begin{aligned}
 \int_{T_s} \boldsymbol{\mu}_h \cdot \mathbf{v}_h(\bar{\mathbf{X}}) \, \mathrm{d}\mathbf{s} &= \sum_{j=1}^J \int_{P_j} \boldsymbol{\mu}_h \cdot \mathbf{v}_h(\bar{\mathbf{X}}) \, \mathrm{d}\mathbf{s} = \sum_{j=1}^J \sum_{i=1}^{I_j} \int_{T_i} \boldsymbol{\mu}_h \cdot \mathbf{v}_h(\bar{\mathbf{X}}) \, \mathrm{d}\mathbf{s} \\
 (21) \qquad \qquad \qquad &= \sum_{j=1}^J \sum_{i=1}^{I_j} |T_i| \sum_{k=1}^K \omega_k \boldsymbol{\mu}_h(\mathbf{p}_k) \cdot \mathbf{v}_h(\bar{\mathbf{X}}(\mathbf{p}_k)),
 \end{aligned}$$

while for (20), we have

$$\begin{aligned}
& \int_{T_s} \boldsymbol{\mu}_h \cdot \mathbf{v}_h(\bar{\mathbf{X}}) + \nabla_s \boldsymbol{\mu}_h : \nabla_s \mathbf{v}_h(\bar{\mathbf{X}}) \, ds \\
&= \sum_{j=1}^J \int_{P_j} \boldsymbol{\mu}_h \cdot \mathbf{v}_h(\bar{\mathbf{X}}) + \nabla_s \boldsymbol{\mu}_h : \nabla_s \mathbf{v}_h(\bar{\mathbf{X}}) \, ds \\
(22) \quad &= \sum_{j=1}^J \sum_{i=1}^{I_j} \int_{T_i} \boldsymbol{\mu}_h \cdot \mathbf{v}_h(\bar{\mathbf{X}}) + \nabla_s \boldsymbol{\mu}_h : \nabla_s \mathbf{v}_h(\bar{\mathbf{X}}) \, ds \\
&= \sum_{j=1}^J \sum_{i=1}^{I_j} |T_i| \sum_{k=1}^K \omega_k [\boldsymbol{\mu}_h(\mathbf{p}_k) \cdot \mathbf{v}_h(\bar{\mathbf{X}}(\mathbf{p}_k)) + \nabla_s \boldsymbol{\mu}_h(\mathbf{p}_k) : \nabla_s \mathbf{v}_h(\bar{\mathbf{X}}(\mathbf{p}_k))].
\end{aligned}$$

Algorithm 1 summarizes the steps of the procedure that is also graphically represented in Figure 2b where purple lines denote the polygons resulting from the intersection between the blue solid element and the cyan velocity triangles; pink lines denote the triangulation constructed using the barycenter.

Algorithm 1 Assembling C_f with mesh intersection

$\{T_f\}_{f=1,\dots,N_F}$ = elements of the fluid mesh

$\{T_s\}_{s=1,\dots,N_S}$ = elements of the reference solid mesh

Compute mesh intersection $\{T_f\}_f \cap \{\bar{\mathbf{X}}(T_s)\}_s$:

 each $\bar{\mathbf{X}}(T_s)$ is partitioned into $J \geq 1$ polygons $\{P_j\}_{j=1,\dots,J}$

 associated with J fluid elements $\{T_{f,j}\}_{j=1,\dots,J}$, i.e. $P_j \leftrightarrow T_{f,j}$

for $\{T_s\}_{s=1,\dots,N_S}$ **do**

for $\{P_j\}_{j=1,\dots,J}$ **do**

if P_j is not a triangle **then**

 | Triangulate P_j into $\{T_i\}_{i=1,\dots,I_j}$

end

else

 | $I_j = 1$ and $T_1 = P_j$

end

for $\{T_i\}_{i=1,\dots,I_j}$ **do**

 | Integrate using the basis functions related to T_s and $T_{f,j}$

 | Load contribution to the global matrix

end

end

end

5.2. Assembling without mesh intersection. A cheaper approach consists in directly integrating on each element of the structure discretization, therefore the first step consists in computing the quadrature nodes of the element under consideration accordingly to the chosen quadrature rule. At this point, the evaluation of the solid basis functions is a trivial operation.

On the other hand, for the evaluation of the shape functions related to the velocity, we must pay attention because the quadrature nodes could belong to different fluid elements: for this reason, it becomes necessary to understand whose fluid triangle each of the quadrature points belongs to. In the case of triangular meshes, this operation can be performed computing the barycentric coordinates of the nodes with respect to all fluid elements. Once this is done, at each quadrature point we can evaluate the fluid functions related to the element that contains it and then integrate. We emphasize that this operation introduces an additional source of error because we are evaluating the fluid basis functions at a quadrature node assuming that at the other ones they are not supported.

In this case, the quadrature over T_s for (19) is given by

$$(23) \quad \int_{T_s} \boldsymbol{\mu}_h \cdot \mathbf{v}_h(\bar{\mathbf{X}}) \, ds = |T_s| \sum_{k=1}^K \omega_k \boldsymbol{\mu}_h(\mathbf{p}_k) \cdot \mathbf{v}_h(\bar{\mathbf{X}}(\mathbf{p}_k));$$

conversely, for (20), we have

$$(24) \quad \begin{aligned} & \int_{T_s} \boldsymbol{\mu}_h \cdot \mathbf{v}_h(\bar{\mathbf{X}}) + \nabla_s \boldsymbol{\mu}_h : \nabla_s \mathbf{v}_h(\bar{\mathbf{X}}) \, ds \\ &= |T_s| \sum_{k=1}^K \omega_k [\boldsymbol{\mu}_h(\mathbf{p}_k) \cdot \mathbf{v}_h(\bar{\mathbf{X}}(\mathbf{p}_k)) + \nabla_s \boldsymbol{\mu}_h(\mathbf{p}_k) : \nabla_s \mathbf{v}_h(\bar{\mathbf{X}}(\mathbf{p}_k))]. \end{aligned}$$

A clarifying example is in Figure 2c, where to fix ideas, we represent the coupling without intersection is the case of a Gaussian quadrature rule with three nodes. Here, if we look at the three points, the basis functions of the related fluid elements are supported also in fluid triangles that we are not considering for this computation. All the steps are reported in Algorithm 2.

5.3. Other geometries and three-dimensional case. It is important to notice that the procedures just described for two dimensional simplices may be also valid for other geometries, such as quadrilateral meshes. Moreover, we can also extend the same ideas to the three dimensional case. For instance, if we consider simplicial meshes in the three-dimensional space, we have to deal with tetrahedra. From the theoretical point of view, our method still works and the two algorithms regarding the assembling techniques for the interface matrix do not change; on the other hand, the computation of the intersection of three dimensional polyhedra is a non-trivial operation and needs the support of specific geometric libraries. For the non intersection case, the use of barycentric coordinates for detecting where the quadrature nodes are placed in the fluid mesh is still a valid procedure.

Algorithm 2 Assembling the C_f matrix block without mesh intersection

$\{T_f\}_{f=1,\dots,N_F}$ = elements of the fluid mesh
 $\{T_s\}_{s=1,\dots,N_S}$ = elements of the reference solid mesh
for $\{T_s\}_{s=1,\dots,N_S}$ **do**
 Compute K quadrature nodes $\mathbf{p}_1, \dots, \mathbf{p}_K$ in T_s accordingly to the rule chosen
 Evaluate solid basis functions
 Find the fluid element containing each mapped quadrature point: $\bar{\mathbf{X}}(\mathbf{p}_k) \leftrightarrow T_{f,k}$ with
 $k = 1, \dots, K$
 for $\{T_{f,k}\}_{k=1,\dots,K}$ **do**
 Evaluate the fluid basis functions in $\bar{\mathbf{X}}(\mathbf{p}_k)$ and integrate over T_s
 Load contribution to the global matrix
 end
end

6. NUMERICAL TESTS

6.1. Model problem. To study how the two approaches presented above affect the convergence of our method, we perform several numerical tests on a two dimensional stationary problem, that turns out to be a simplified version of the problem presented in Problem 3.

Problem 5. Let $\bar{\mathbf{X}} \in (W^{1,\infty}(\mathcal{B}))^2$ be invertible with Lipschitz inverse. Given $\mathbf{f} \in (L^2(\Omega))^2$, $\mathbf{g} \in (L^2(\mathcal{B}))^2$ and $\mathbf{d} \in (L^2(\mathcal{B}))^2$, find $(\mathbf{u}, p) \in (H_0^1(\Omega))^2 \times L_0^2(\Omega)$, $\mathbf{X} \in (H^1(\mathcal{B}))^2$ and $\boldsymbol{\lambda} \in (H^1(\mathcal{B}))^2$, such that

$$(25a) \quad \mathbf{a}_f(\mathbf{u}, \mathbf{v}) - (\operatorname{div} \mathbf{v}, p) + \mathbf{c}(\boldsymbol{\lambda}, \mathbf{v}(\bar{\mathbf{X}})) = (\mathbf{f}, \mathbf{v}) \quad \forall \mathbf{v} \in (H_0^1(\Omega))^2$$

$$(25b) \quad (\operatorname{div} \mathbf{u}, q) = 0 \quad \forall q \in L_0^2(\Omega)$$

$$(25c) \quad \mathbf{a}_s(\mathbf{X}, \mathbf{Y}) - \mathbf{c}(\boldsymbol{\lambda}, \mathbf{Y}) = (\mathbf{g}, \mathbf{Y}) \quad \forall \mathbf{Y} \in (H^1(\mathcal{B}))^2$$

$$(25d) \quad \mathbf{c}(\boldsymbol{\mu}, \mathbf{u}(\bar{\mathbf{X}}) - \mathbf{X}) = \mathbf{c}(\boldsymbol{\mu}, \mathbf{d}) \quad \forall \boldsymbol{\mu} \in (H^1(\mathcal{B}))^2$$

where the bilinear forms are defined as

$$\mathbf{a}_f(\mathbf{u}, \mathbf{v}) = (\nabla \mathbf{u}, \nabla \mathbf{v})$$

$$\mathbf{a}_s(\mathbf{X}, \mathbf{Y}) = (\nabla_s \mathbf{X}, \nabla_s \mathbf{Y})_{\mathcal{B}}$$

$$\mathbf{c}(\boldsymbol{\mu}, \mathbf{Y}) = (\nabla_s \boldsymbol{\mu}, \nabla_s \mathbf{Y})_{\mathcal{B}} + (\boldsymbol{\mu}, \mathbf{Y})_{\mathcal{B}}$$

For each of our eight tests, we first select analytical solutions not belonging to the finite element spaces chosen for the approximation, and then we compute the right hand sides as follows

$$\mathbf{f} = -\Delta \mathbf{u} + \nabla p + \mathbf{c}(\boldsymbol{\lambda}, \mathbf{v})$$

$$\mathbf{g} = -\Delta \mathbf{X} - \boldsymbol{\lambda} + \mathbf{g}_{\partial \mathcal{B}}$$

$$\mathbf{d} = \mathbf{u}(\bar{\mathbf{X}}) - \mathbf{X}$$

where $\mathbf{g}_{\partial \mathcal{B}}$ is the boundary term we obtain integrating by parts the equation of the solid without considering the multiplier; for this reason it is related with the normal derivative

of \mathbf{X} on $\partial\mathcal{B}$

$$\mathbf{a}_s(\mathbf{X}, \mathbf{Y}) = (\nabla_s \mathbf{X}, \nabla_s \mathbf{Y})_{\mathcal{B}} = (-\Delta \mathbf{X}, \mathbf{Y})_{\mathcal{B}} + (\nabla_s \mathbf{X} \cdot \mathbf{n}_s, \mathbf{Y})_{\partial\mathcal{B}}.$$

For the definition of the datum \mathbf{f} , we emphasize again that the contribution given by the multiplier is weakly imposed only on the solid domain. Consequently, exploiting once again the immersion of the solid in the fluid, in order to assemble this term, we have still to use the mesh intersection to ensure an optimal computation of the datum.

The first six cases are characterized by the fact that $\overline{\mathbf{X}}$ is set to be the identity, hence the reference and the actual solid domain coincide; moreover in Tests 5 and 6 we consider the case of discontinuous pressure with the discontinuity matching both the fluid triangulation and the boundary of the structure and with non-matching interface, respectively. On the other hand, for the last two tests, we consider the case of a non trivial mapping $\overline{\mathbf{X}}$ with two different bodies. Each test is performed choosing the Stokes pair $\mathcal{P}_1 - iso - \mathcal{P}_2 / \mathcal{P}_1$ introduced by Bercovier and Pironneau in [2] and its enhanced version $\mathcal{P}_1 - iso - \mathcal{P}_2 / \mathcal{P}_1 + \mathcal{P}_0$ discussed in [4], while for the approximation of both the solid variables, we choose piecewise linear elements \mathcal{P}_1 .

6.2. Bercovier-Pironneau element. The Bercovier-Pironneau element is a popular choice in this field of study because it is considered as a cheaper version of the Hood-Taylor element: they share the same set of degrees of freedom, but the Bercovier-Pironneau involves linear shape functions also for the velocity discretization, hence, for this reason it is a low-order choice.

The peculiarity of a choice of this type concerns with the discretization of the domain and the need to work with two different meshes for velocity and pressure: indeed, each pressure element is a macro element containing the four velocity triangles obtained by connecting the middle points. Therefore, we can write \mathbf{V}_h and Q_h as

$$(26) \quad \begin{aligned} \mathbf{V}_h &= \{\mathbf{v} \in (H_0^1(\Omega))^2 : \mathbf{v}|_T \in (\mathcal{P}_1(T))^2 \quad \forall T \in \mathcal{T}_{h/2}\} \\ Q_h &= \{q \in L_0^2(\Omega) : q|_T \in \mathcal{P}_1(T) \quad \forall T \in \mathcal{T}_h\}. \end{aligned}$$

The enhanced version with discontinuous pressure has been introduced because it guarantees a local mass conservation with the average of the divergence equal to zero in each element; in this case, the pressure space is defined as

$$(27) \quad Q_h = \{q \in L_0^2(\Omega) : q = q_1 + q_0, q_1 \in H^1(\Omega), q_1|_T \in \mathcal{P}_1(T), q_0|_T \in \mathcal{P}_0(T) \quad \forall T \in \mathcal{T}_h\}.$$

The use of two different meshes for velocity and pressure has consequences also in the assembling of the interface matrix C_f : since it is defined from $\mathbf{c}(\boldsymbol{\lambda}_h, \mathbf{u}_h(\overline{\mathbf{X}}))$, we have to couple the solid mesh \mathcal{S}_h with the one related to the velocity $\mathcal{T}_{h/2}$.

6.3. Mesh generation. For the fluid domain discretization, we use uniform meshes. In particular, in the case of $\mathcal{P}_1 - iso - \mathcal{P}_2 / \mathcal{P}_1 + \mathcal{P}_0$ elements, we need to pay attention to corner triangles, i.e. triangles with two edges placed on the boundary, because they produce wrong approximations due to the sensitivity to the boundary conditions (the phenomenon is analyzed in [4]): this situation is corrected via diagonal exchange. On the other hand, for the structure, we work with both uniform and unstructured grids.

All the meshes, satisfying the shape regularity property, are generated on the unit square $[0, 1]^2$ and then mapped into Ω and \mathcal{B} . To obtain unstructured meshes, we used the Gmsh finite element mesh generator [18].

Some examples are presented in Figure 3, while in Table 1, we report the number of degrees of freedom of each unknown in dependence of the chosen discrete spaces and meshes.

DOFs \mathbf{u}_h	DOFs p_h	DOFs p_h^0	DOFs \mathbf{X}_h, λ_h	DOFs $\mathbf{X}_h^{uns}, \lambda_h^{uns}$
2,178	289	801	162	232
8,450	1,089	3,137	578	742
33,282	4,225	12,417	2,178	2,788
132,028	16,641	49,409	8,450	11,018
526,338	66,049	197,121	33,282	43,734
2,101,250	263,169	787,457	132,028	174,316

TABLE 1. Mesh parameters and degrees of freedom related to the chosen discrete spaces. The last column regards the solid discretization with unstructured mesh, while the others are related to uniform meshes. With p_h^0 , we denote the pressure approximation with $\mathcal{P}_1 + \mathcal{P}_0$ elements.

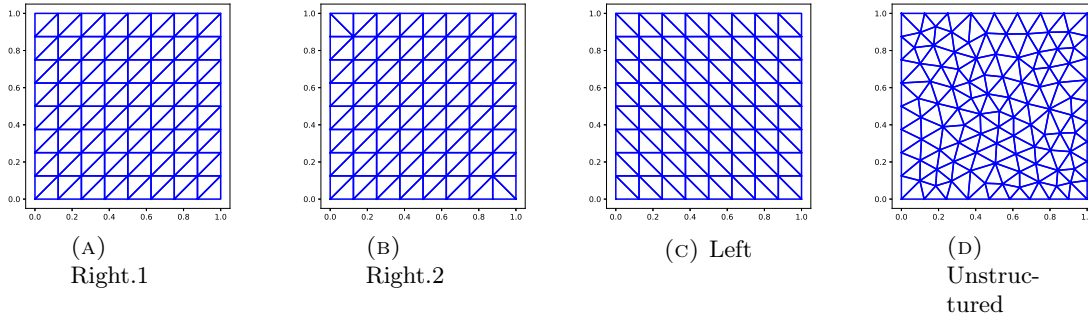


FIGURE 3. Some examples of meshes. (a) and (b) are used for the fluid discretization and, in particular, in (b) we can see the correction of the corner triangles we need when we work with the $\mathcal{P}_1 + \mathcal{P}_0$ approximation of the pressure. In (c) and (d), we have two meshes used for the structure. When we work with perfectly matching meshes, we use the type in (a) for both the domains.

6.4. Mesh intersection in Python. All the tests has been performed with Python 3.8 scripts written specifically for this purpose. We focus our attention on the libraries used for the geometric computations.

In Algorithms 1 and 2, we denoted by N_F and N_S the number of elements of the two meshes (fluid and solid respectively), hence the number of the intersections to be tested is $N_S \times N_F$, which may become large when working with very fine meshes. In order to avoid useless computations, we can check in advance if two elements have non-empty intersection making use of the bounding box technique: if the bounding boxes are disjoint, the intersection of the two elements under consideration is empty and therefore we can move on and examine another pair of elements. These preliminary tests can be done using a tree-search algorithm: in our Python code, this procedure is based on the `Rtree` library.

To actually compute the intersection between two triangles, the `Shapely.geometry` library is very useful because it can manage objects of type `polygon` that allow us to compute and represent the intersection in an easy way: it is enough to construct two `polygon` objects to represent the two triangles under consideration and then use the `intersection()` method to obtain the resulting polygon.

On the other hand, if we assemble C_f without intersection, we do not need particular tools: if we consider a solid triangle with its quadrature nodes, we can simply convert them with respect to the barycentric coordinates system related to each fluid element and understand in which one they are placed.

The geometric configurations of our tests are summarized in Figure 4: we can see the reciprocal position of fluid and solid meshes in the case of coarse spatial discretizations.

6.5. Quadrature rules for c_h . Now we focus our attention on the quadrature rules used for the assembling of the interface matrix: as mentioned above, we set \mathbf{c} as in (10). With the chosen finite element spaces, the basis functions both for the velocity and the Lagrange multiplier are piecewise linear and this implies that we need to compute integrals with second order precision for the L^2 contributions and with first order precision for the gradients terms. In particular, in order to have exact computations, we used Gauss quadrature rules with the required accuracy for both approaches. Furthermore, in order to better understand the behavior of the system when the interface matrix is assembled without mesh intersection, we implemented, only for this case, also a Gauss quadrature rule with third order precision.

For the sake of precision, we report the definition of these formulas on the reference triangle \hat{T} .

Definition 1 (Gauss rule, order 1). *A polynomial of degree one f can be integrated exactly by the following one-point formula*

$$(28) \quad \int_{\hat{T}} f(\mathbf{x}) \, d\mathbf{x} \approx |\hat{T}| f(\mathbf{x}_{\hat{T}})$$

where $\mathbf{x}_{\hat{T}}$ is the barycenter of \hat{T} .

Definition 2 (Gauss rule, order 2). *A polynomial of degree two f can be integrated exactly by the following three-points formula*

$$(29) \quad \int_{\hat{T}} f(\mathbf{x}) \, d\mathbf{x} \approx \frac{|\hat{T}|}{3} \sum_{k=1}^3 f(\mathbf{x}^{(k)})$$

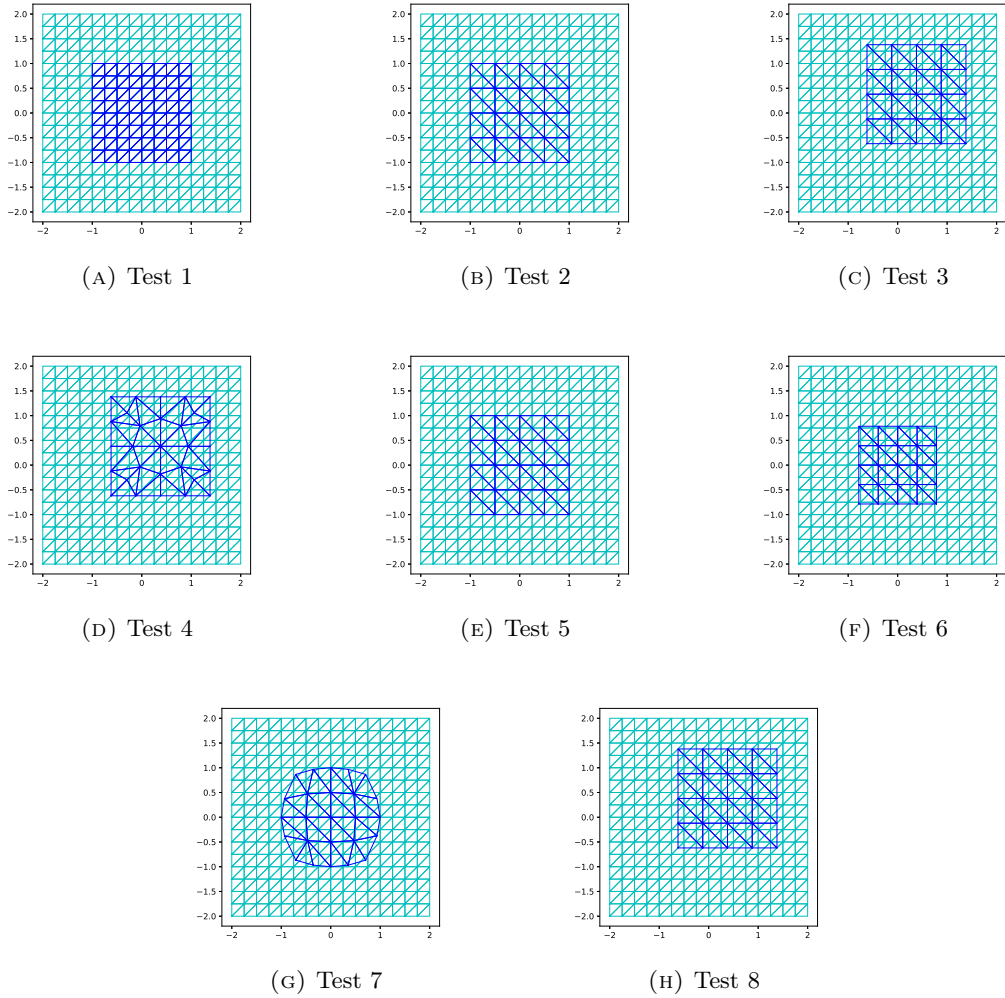


FIGURE 4. All the geometric configurations of our tests with coarse meshes. In particular, for the first six tests, we have that the reference and actual configuration of the solid coincide. We point out that the fluid mesh represented is the one related to the velocity. Moreover, each geometric configuration remains unchanged when we refine the two meshes.

where the quadrature nodes, represented in barycentric coordinates, are $\mathbf{x}^{(1)} = (2/3, 1/6, 1/6)$ and its permutation.

Definition 3 (Gauss rule, order 3). *A polynomial of degree three f can be integrated exactly by the following four-points formula*

$$(30) \quad \int_{\hat{T}} f(\mathbf{x}) d\mathbf{x} \approx |\hat{T}| \left(\frac{25}{48} \sum_{k=1}^3 f(\mathbf{x}^{(k)}) - \frac{9}{16} f(\mathbf{x}_{\hat{T}}) \right)$$

where the quadrature nodes, represented in barycentric coordinates, are $\mathbf{x}^{(1)} = (3/5, 1/5, 1/5)$ with its permutation and the barycenter $\mathbf{x}_{\hat{T}}$ of \hat{T} .

6.6. Test 1. Sanity check. Let us consider Problem 5 on the fluid square domain $\Omega = [-2, 2]^2$ and with $\mathcal{B} = \Omega^s = [-1, 1]^2$. This assumption implies that the initial datum $\bar{\mathbf{X}}$ is simply the identity, hence $\mathbf{d} = \mathbf{0}$. We compute the right hand sides \mathbf{f} , \mathbf{g} , and \mathbf{d} in order to obtain the following solutions:

$$(31) \quad \begin{aligned} \mathbf{u}(x, y) &= \text{curl} \left((4 - x^2)^2 (4 - y^2)^2 \right) \\ \mathbf{u}|_{\partial\Omega} &= \mathbf{0} \\ p(x, y) &= 150 \sin(x) \\ \mathbf{X}(x, y) &= \mathbf{u}(x, y) \\ \boldsymbol{\lambda}(x, y) &= (e^x, e^y). \end{aligned}$$

In particular, we discretize both domains using perfectly matching uniform right-oriented meshes (Figures 3a and 3b): we can see that the two methods for the assembling of the interface matrix are equivalent, indeed, the convergence rates confirm what expected. The errors and the convergence rates are collected in Tables 2, 3, 4, 5: in particular in the first two tables, we have the data related to the approximation with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1/\mathcal{P}_1/\mathcal{P}_1$ element, while in Table 4 and 5, we have the results related to the $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1 + \mathcal{P}_0/\mathcal{P}_1/\mathcal{P}_1$ approximation.

In Subsection 6.2, we discussed the main features of the Bercovier-Pironneau element, which is a low order Stokes pair. Therefore, we expect that the errors $\|p - p_h\|_0$, $\|\mathbf{u} - \mathbf{u}_h\|_1$ decay with order one, while $\|\mathbf{u} - \mathbf{u}_h\|_0$ decays with order two. Moreover, since for both displacement and multiplier we use \mathcal{P}_1 elements, we expect that they converge with order two in the L^2 norm and with order one in the H^1 norm. From the results, we can see that, in the case of the classical Bercovier-Pironneau element, we get a superconvergence for pressure and Lagrange multiplier.

6.7. Test 2. We consider the same problem as in Test 1, but changing the meshes: we discretize the fluid domain with a right-oriented uniform mesh (Figures 3a and 3b), while the solid domain is discretized with a left-oriented uniform mesh (Figure 3c).

In this test, we consider the case of meshes with coincident edges to make a first comparison between the methods in a situation where the intersection of the meshes does not have a complex geometry. From Figures 5 and 6, we can, however, already notice that the convergence rates of the method without intersection with quadrature rule of order two are lower than the convergence rates of the method with intersection. On the other hand, the increase of the quadrature order for the method without intersection produces in this

Errors and convergence rates for Test 1 • $\mathcal{P}_1 - \text{iso} - \mathcal{P}_2/\mathcal{P}_1/\mathcal{P}_1/\mathcal{P}_1$						
$h_{\mathcal{T}}$	$\ p - p_h\ _0$		$\ \mathbf{u} - \mathbf{u}_h\ _0$		$\ \mathbf{u} - \mathbf{u}_h\ _1$	
	Error	Rate	Error	Rate	Error	Rate
<i>Coupling with mesh intersection</i>						
1/4	2.102e-02	-	9.622e-03	-	7.684e-02	-
1/8	6.251e-03	1.75	2.408e-03	2.00	3.834e-02	1.00
1/16	1.977e-03	1.66	6.017e-04	2.00	1.915e-02	1.00
1/32	6.546e-04	1.59	1.504e-04	2.00	9.572e-03	1.00
1/64	2.233e-04	1.55	3.758e-05	2.00	4.785e-03	1.00
1/128	7.745e-05	1.53	9.394e-06	2.00	2.392e-03	1.00
<i>Coupling without mesh intersection, quad. rule of order 2</i>						
1/4	2.102e-02	-	9.622e-03	-	7.684e-02	-
1/8	6.251e-03	1.75	2.408e-03	2.00	3.834e-02	1.00
1/16	1.977e-03	1.66	6.017e-04	2.00	1.915e-02	1.00
1/32	6.546e-04	1.59	1.504e-04	2.00	9.572e-03	1.00
1/64	2.233e-04	1.55	3.758e-05	2.00	4.785e-03	1.00
1/128	7.745e-05	1.53	9.394e-06	2.00	2.392e-03	1.00
<i>Coupling without mesh intersection, quad. rule of order 3</i>						
1/4	2.102e-02	-	9.622e-03	-	7.684e-02	-
1/8	6.251e-03	1.75	2.408e-03	2.00	3.834e-02	1.00
1/16	1.977e-03	1.66	6.017e-04	2.00	1.915e-02	1.00
1/32	6.546e-04	1.59	1.504e-04	2.00	9.572e-03	1.00
1/64	2.233e-04	1.55	3.758e-05	2.00	4.785e-03	1.00
1/128	7.745e-05	1.53	9.394e-06	2.00	2.392e-03	1.00

TABLE 2. Errors and convergence rates for the fluid variables of Test 1 discretized with $\mathcal{P}_1 - \text{iso} - \mathcal{P}_2/\mathcal{P}_1/\mathcal{P}_1/\mathcal{P}_1$

case a significant improvement, almost reaching the good behavior of the method with intersection.

6.8. **Test 3.** For this third test, we use again the solutions chosen in (31). On the other hand, we set $\Omega = [-2, 2]^2$ and $\mathcal{B} = \Omega^s = [-0.62, 1.38]^2$ so that the interface of the solid does not match with the fluid mesh, discretizing them with uniform meshes as before. The results are reported in Figures 7 and 8. Thanks to the new positioning of the structure, without matching boundaries, we get clearer results. The only method capable of ensuring good convergence is the one that computes the intersection. In this case, the increase in precision of the quadrature rule used when the intersection is not calculated does not produce any improvement.

Errors and convergence rates for Test 1 • $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1/\mathcal{P}_1/\mathcal{P}_1$									
h_S	$\ \mathbf{X} - \mathbf{X}_h\ _{0,\mathcal{B}}$		$\ \mathbf{X} - \mathbf{X}_h\ _{1,\mathcal{B}}$		$\ \boldsymbol{\lambda} - \boldsymbol{\lambda}_h\ _{0,\mathcal{B}}$		$\ \boldsymbol{\lambda} - \boldsymbol{\lambda}_h\ _{1,\mathcal{B}}$		
	Error	Rate	Error	Rate	Error	Rate	Error	Rate	
<i>Coupling with mesh intersection</i>									
1/8	8.011e-03	-	4.972e-02	-	1.908e-01	-	4.166e-01	-	
1/16	2.011e-03	1.99	2.479e-02	1.00	4.786e-02	2.00	1.111e-01	1.91	
1/32	5.032e-04	2.00	1.239e-02	1.00	1.197e-02	2.00	2.963e-02	1.91	
1/64	1.258e-04	2.00	6.193e-03	1.00	2.991e-03	2.00	8.185e-03	1.86	
1/128	3.146e-05	2.00	3.096e-03	1.00	7.476e-04	2.00	2.524e-03	1.70	
1/256	7.864e-06	2.00	1.548e-03	1.00	1.869e-04	2.00	9.445e-04	1.42	
<i>Coupling without mesh intersection, quad. rule of order 2</i>									
1/8	8.011e-03	-	4.972e-02	-	1.908e-01	-	4.166e-01	-	
1/16	2.011e-03	1.99	2.479e-02	1.00	4.786e-02	2.00	1.111e-01	1.91	
1/32	5.032e-04	2.00	1.239e-02	1.00	1.197e-02	2.00	2.963e-02	1.91	
1/64	1.258e-04	2.00	6.193e-03	1.00	2.991e-03	2.00	8.185e-03	1.86	
1/128	3.146e-05	2.00	3.096e-03	1.00	7.476e-04	2.00	2.524e-03	1.70	
1/256	7.864e-06	2.00	1.548e-03	1.00	1.869e-04	2.00	9.445e-04	1.42	
<i>Coupling without mesh intersection, quad. rule of order 3</i>									
1/8	8.011e-03	-	4.972e-02	-	1.908e-01	-	4.166e-01	-	
1/16	2.011e-03	1.99	2.479e-02	1.00	4.786e-02	2.00	1.111e-01	1.91	
1/32	5.032e-04	2.00	1.239e-02	1.00	1.197e-02	2.00	2.963e-02	1.91	
1/64	1.258e-04	2.00	6.193e-03	1.00	2.991e-03	2.00	8.185e-03	1.86	
1/128	3.146e-05	2.00	3.096e-03	1.00	7.476e-04	2.00	2.524e-03	1.70	
1/256	7.864e-06	2.00	1.548e-03	1.00	1.869e-04	2.00	9.445e-04	1.42	

TABLE 3. Errors and convergence rates for the solid variables of Test 1 discretized with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1/\mathcal{P}_1/\mathcal{P}_1$

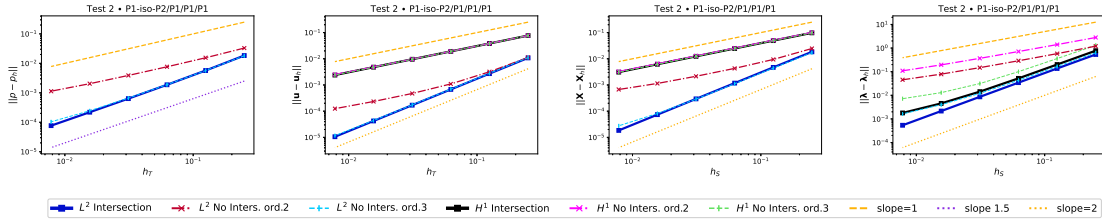


FIGURE 5. Convergence plots of Test 2 with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1/\mathcal{P}_1/\mathcal{P}_1$

Errors and convergence rates for Test 1 • $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1 + \mathcal{P}_0/\mathcal{P}_1/\mathcal{P}_1$						
$h_{\mathcal{T}}$	$\ p - p_h\ _0$		$\ \mathbf{u} - \mathbf{u}_h\ _0$		$\ \mathbf{u} - \mathbf{u}_h\ _1$	
	Error	Rate	Error	Rate	Error	Rate
<i>Coupling with mesh intersection</i>						
1/4	7.981e-02	-	1.043e-02	-	8.042e-02	-
1/8	3.939e-02	1.02	2.617e-03	1.99	4.017e-02	1.00
1/16	1.957e-02	1.01	6.549e-04	2.00	2.008e-02	1.00
1/32	9.749e-03	1.00	1.637e-04	2.00	1.004e-02	1.00
1/64	4.866e-03	1.00	4.093e-05	2.00	5.018e-03	1.00
1/128	2.431e-03	1.00	1.023e-05	2.00	2.509e-03	1.00
<i>Coupling without mesh intersection, quad. rule of order 2</i>						
1/4	7.981e-02	-	1.043e-02	-	8.042e-02	-
1/8	3.939e-02	1.02	2.617e-03	1.99	4.017e-02	1.00
1/16	1.957e-02	1.01	6.549e-04	2.00	2.008e-02	1.00
1/32	9.749e-03	1.00	1.637e-04	2.00	1.004e-02	1.00
1/64	4.866e-03	1.00	4.093e-05	2.00	5.018e-03	1.00
1/128	2.431e-03	1.00	1.023e-05	2.00	2.509e-03	1.00
<i>Coupling without mesh intersection, quad. rule of order 3</i>						
1/4	7.981e-02	-	1.043e-02	-	8.042e-02	-
1/8	3.939e-02	1.02	2.617e-03	1.99	4.017e-02	1.00
1/16	1.957e-02	1.01	6.549e-04	2.00	2.008e-02	1.00
1/32	9.749e-03	1.00	1.637e-04	2.00	1.004e-02	1.00
1/64	4.866e-03	1.00	4.093e-05	2.00	5.018e-03	1.00
1/128	2.431e-03	1.00	1.023e-05	2.00	2.509e-03	1.00

TABLE 4. Errors and convergence rates for the fluid variables of Test 1 discretized with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1 + \mathcal{P}_0/\mathcal{P}_1/\mathcal{P}_1$

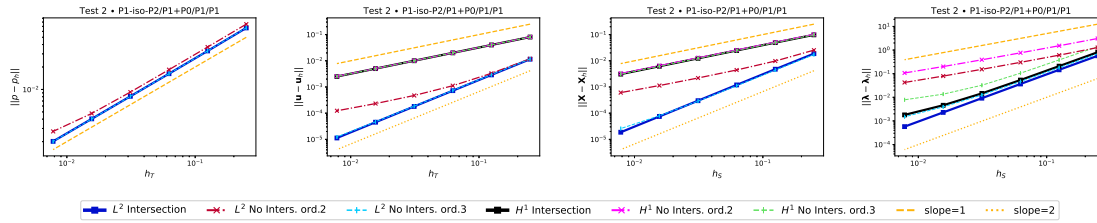


FIGURE 6. Convergence plots of Test 2 with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1 + \mathcal{P}_0/\mathcal{P}_1/\mathcal{P}_1$

Errors and convergence rates for Test 1 • $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1 + \mathcal{P}_0/\mathcal{P}_1/\mathcal{P}_1$									
h_S	$\ \mathbf{X} - \mathbf{X}_h\ _{0,\mathcal{B}}$		$\ \mathbf{X} - \mathbf{X}_h\ _{1,\mathcal{B}}$		$\ \boldsymbol{\lambda} - \boldsymbol{\lambda}_h\ _{0,\mathcal{B}}$		$\ \boldsymbol{\lambda} - \boldsymbol{\lambda}_h\ _{1,\mathcal{B}}$		
	Error	Rate	Error	Rate	Error	Rate	Error	Rate	
<i>Coupling with mesh intersection</i>									
1/8	8.854e-03	-	5.239e-02	-	2.300e-01	-	1.861e+00	-	
1/16	2.228e-03	1.99	2.616e-02	1.00	5.802e-02	1.99	9.338e-01	0.99	
1/32	5.578e-04	2.00	1.308e-02	1.00	1.453e-02	2.00	4.672e-01	1.00	
1/64	1.395e-04	2.00	6.539e-03	1.00	3.632e-03	2.00	2.336e-01	1.00	
1/128	3.487e-05	2.00	3.269e-03	1.00	9.078e-04	2.00	1.168e-01	1.00	
1/256	8.718e-06	2.00	1.635e-03	1.00	2.269e-04	2.00	5.840e-02	1.00	
<i>Coupling without mesh intersection, quad. rule of order 2</i>									
1/8	8.854e-03	-	5.239e-02	-	2.300e-01	-	1.861e+00	-	
1/16	2.228e-03	1.99	2.616e-02	1.00	5.802e-02	1.99	9.338e-01	0.99	
1/32	5.578e-04	2.00	1.308e-02	1.00	1.453e-02	2.00	4.672e-01	1.00	
1/64	1.395e-04	2.00	6.539e-03	1.00	3.632e-03	2.00	2.336e-01	1.00	
1/128	3.487e-05	2.00	3.269e-03	1.00	9.078e-04	2.00	1.168e-01	1.00	
1/256	8.718e-06	2.00	1.635e-03	1.00	2.269e-04	2.00	5.840e-02	1.00	
<i>Coupling without mesh intersection, quad. rule of order 3</i>									
1/8	8.854e-03	-	5.239e-02	-	2.300e-01	-	1.861e+00	-	
1/16	2.228e-03	1.99	2.616e-02	1.00	5.802e-02	1.99	9.338e-01	0.99	
1/32	5.578e-04	2.00	1.308e-02	1.00	1.453e-02	2.00	4.672e-01	1.00	
1/64	1.395e-04	2.00	6.539e-03	1.00	3.632e-03	2.00	2.336e-01	1.00	
1/128	3.487e-05	2.00	3.269e-03	1.00	9.078e-04	2.00	1.168e-01	1.00	
1/256	8.718e-06	2.00	1.635e-03	1.00	2.269e-04	2.00	5.840e-02	1.00	

TABLE 5. Errors and convergence rates for the solid variables of Test 1 discretized with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1 + \mathcal{P}_0/\mathcal{P}_1/\mathcal{P}_1$

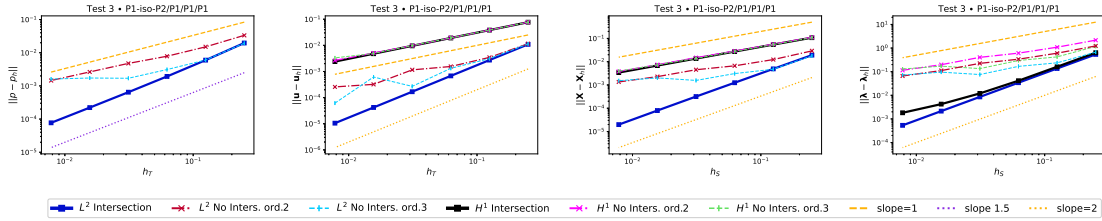


FIGURE 7. Convergence plots of Test 3 with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1/\mathcal{P}_1/\mathcal{P}_1$

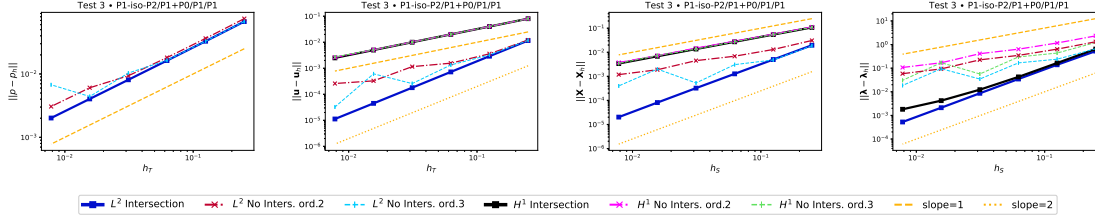


FIGURE 8. Convergence plots of Test 3 with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1 + \mathcal{P}_0/\mathcal{P}_1/\mathcal{P}_1$

6.9. **Test 4.** In the same setting of the previous test, we partition the solid domain making use of an unstructured grid (Figure 3d). The use of this type of mesh for the discretization of the immersed body places us in an even more general setting. Figures 9 and 10 show that the bad performances seen in the previous tests for the methods without intersection are amplified; in the case of the pressure, the non-intersection methods perform better when we use discontinuous elements. Furthermore, if in the previous tests $\|\lambda - \lambda_h\|_{1,\mathcal{B}}$ showed a convergence rate higher than 1.5, this does not happen now, settling around 1.

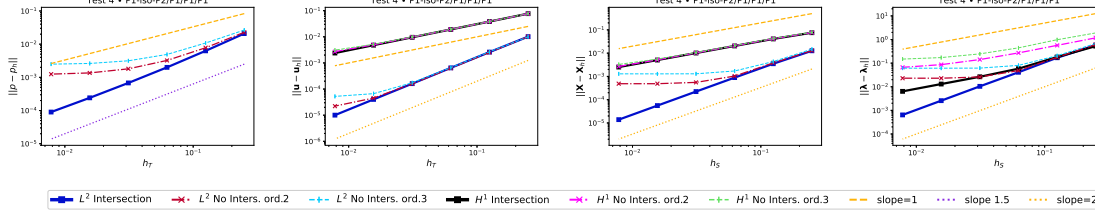


FIGURE 9. Convergence plots of Test 4 with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1/\mathcal{P}_1/\mathcal{P}_1$

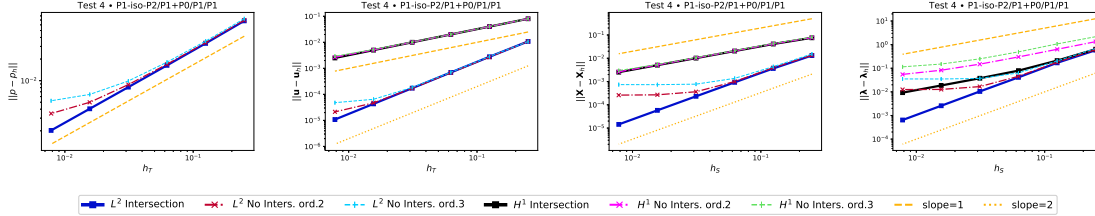


FIGURE 10. Convergence plots of Test 4 with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1 + \mathcal{P}_0/\mathcal{P}_1/\mathcal{P}_1$

6.10. **Test 5.** The configuration of this test is the same we chose for Test 1, but, in this case, we consider a pressure which is discontinuous on $\partial\Omega^s$. We assume that $\Omega = [-2, 2]^2$ and $\mathcal{B} = \Omega^s = [-1, 1]^2$: as before, for the fluid domain we choose a right-uniform mesh, while for the structure a left-uniform one. The fact that the reference and the actual solid

domain coincide implies that $\bar{\mathbf{X}} = \mathbf{I}_{\mathcal{B}}$. We compute the right hand sides \mathbf{f} , \mathbf{g} in order to obtain the following solutions:

$$(32) \quad \begin{aligned} \mathbf{u}(x, y) &= \text{curl} \left((4 - x^2)^2 (4 - y^2)^2 \right) \\ \mathbf{u}|_{\partial\Omega} &= \mathbf{0} \\ p(x, y) &= \begin{cases} 150 \sin(x) - \frac{50}{3} & \text{in } \Omega^f \\ 150 \sin(x) + 50 & \text{in } \Omega^s = \mathcal{B}. \end{cases} \\ \mathbf{X}(x, y) &= \mathbf{u}(x, y) \\ \boldsymbol{\lambda}(x, y) &= (e^x, e^y). \end{aligned}$$

Here, the choice of p implies that we have to introduce a new weak term in the definition of the right hand side \mathbf{f} taking into account the discontinuity. Indeed, we have

$$\int_{\Omega} p \operatorname{div} \mathbf{v} \, d\mathbf{x} = \int_{\Omega^s} \mathbf{v} \cdot \nabla p \, d\mathbf{x} - \int_{\partial\Omega^s} (\mathbf{v} \cdot \mathbf{n}_s) p \, dA + \int_{\Omega \setminus \Omega^s} \mathbf{v} \cdot \nabla p \, d\mathbf{x} - \int_{\partial\Omega^s} (\mathbf{v} \cdot \mathbf{n}_f) p \, dA;$$

therefore, in order to compute the boundary integrals combining $\partial\Omega^s$ with the basis for \mathbf{V}_h , we have to take care of the immersion of the solid boundary in the fluid mesh. We particularly emphasize that we are in the situation where the discontinuity of the pressure not only matches the interface, but also the velocity mesh. In this situation, we can see how the choice of element for pressure affects the convergence of all the variables. As expected and presented in [4], the enhanced space $\mathcal{P}_1 + \mathcal{P}_0$ allows us to reach the optimal convergence without presenting the Gibbs phenomenon that we get using the classical \mathcal{P}_1 pressure.

Regarding the computation of the intersection, we can see again that avoiding this procedure we do not obtain the optimal convergence rate. In particular, we can see that the behavior is similar to the one seen in the first test. The results are collected in Figures 11 and 12.

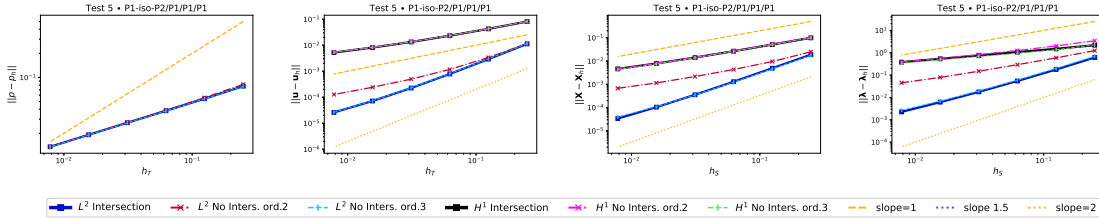


FIGURE 11. Convergence plots of Test 5 with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1/\mathcal{P}_1/\mathcal{P}_1$

6.11. **Test 6.** We consider again a discontinuous pressure on the boundary of the structure but in the case where the fluid triangulation does not match it: for this reason, we choose $\mathcal{B} = \Omega^s = [-\pi/4, \pi/4]^2$ and we discretize it with a uniform left mesh. Our choices for Ω and its discretization remain unchanged. We compute the right hand sides \mathbf{f} , \mathbf{g} , and \mathbf{d} in order

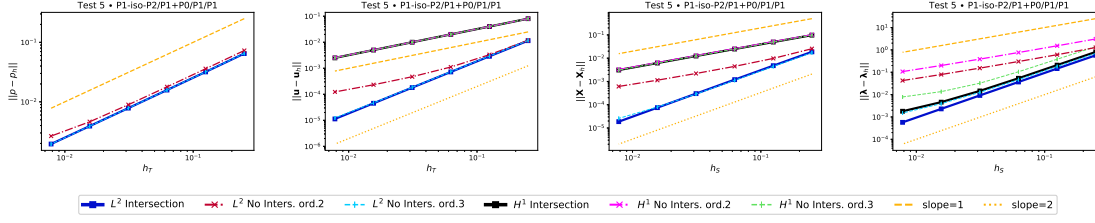


FIGURE 12. Convergence plots of Test 5 with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1 + \mathcal{P}_0/\mathcal{P}_1/\mathcal{P}_1$

to obtain the following solutions:

$$\begin{aligned}
 \mathbf{u}(x, y) &= \text{curl} \left((4 - x^2)^2 (4 - y^2)^2 \right) \\
 \mathbf{u}|_{\partial\Omega} &= \mathbf{0} \\
 (33) \quad p(x, y) &= \begin{cases} 150 \sin(x) + 50 \frac{\pi^2}{4} \left(\frac{\pi^2}{4} - 16 \right)^{-1} & \text{in } \Omega^f \\ 150 \sin(x) + 50 & \text{in } \Omega^s = \mathcal{B}. \end{cases} \\
 \mathbf{X}(x, y) &= \mathbf{u}(x, y) \\
 \lambda(x, y) &= (e^x, e^y).
 \end{aligned}$$

Due to the non matching boundary with respect to the singularity, also the $\mathcal{P}_1 + \mathcal{P}_0$ choice does not reach the optimal convergence rate presenting the Gibbs phenomenon. In this case, as we can see in Figures 13 and 14, the non-computation of the mesh intersection does not affect the fluid variables basically; on the other hand, for the solid ones, the behavior changes drastically.

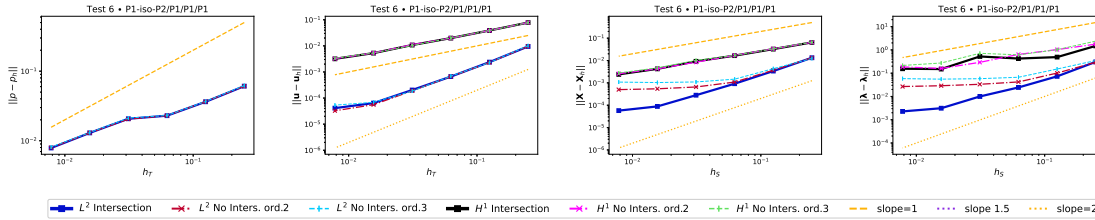


FIGURE 13. Convergence plots of Test 6 with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1/\mathcal{P}_1/\mathcal{P}_1$

6.12. Test 7, disk. At this point, we consider a situation where the initial map $\bar{\mathbf{X}}$ is not trivial: indeed, we consider as reference domain for the solid the square $\mathcal{B} = [-1, 1]^2$ (partitioned with a uniform left mesh), while the actual immersed body is the unit disk $\Omega^s = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\| \leq 1\}$; hence we have

$$\bar{\mathbf{X}}(x, y) = \left(x \sqrt{1 - \frac{y^2}{2}}, y \sqrt{1 - \frac{x^2}{2}} \right).$$

On the other hand, the domain Ω is still the same as before, and the computations are done in order to obtain as solutions the ones chosen in (31). In addition, since the map

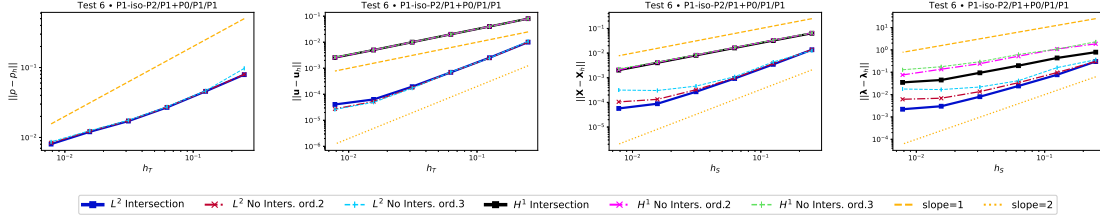


FIGURE 14. Convergence plots of Test 6 with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1 + \mathcal{P}_0/\mathcal{P}_1/\mathcal{P}_1$

$\bar{\mathbf{X}}$ is nontrivial, $\mathbf{d} = \mathbf{u}(\bar{\mathbf{X}}) - \mathbf{X} \neq \mathbf{0}$ and we have again to deal with the computation of $\mathbf{c}(\boldsymbol{\mu}, \mathbf{u}(\bar{\mathbf{X}}))$ intersecting the two meshes as we do in order to compute \mathbf{f} .

The substantial difference between this test and the previous ones is that the evaluation of the solid basis functions is done after mapping back the nodes on the reference domain \mathcal{B} . Also in this case, the behavior observed previously is confirmed with the intersection approach prevailing on the non-intersection one (see Figures 15 and 16).

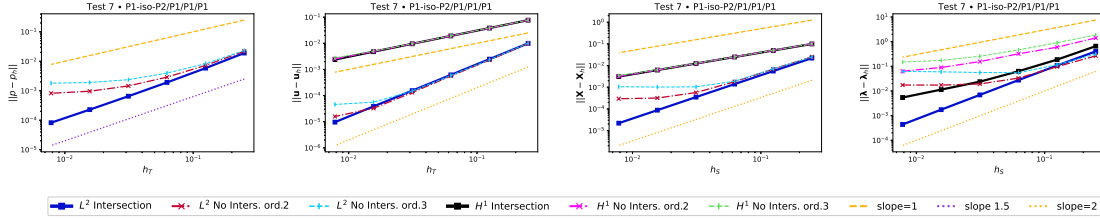


FIGURE 15. Convergence plots of Test 7 with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1/\mathcal{P}_1/\mathcal{P}_1$

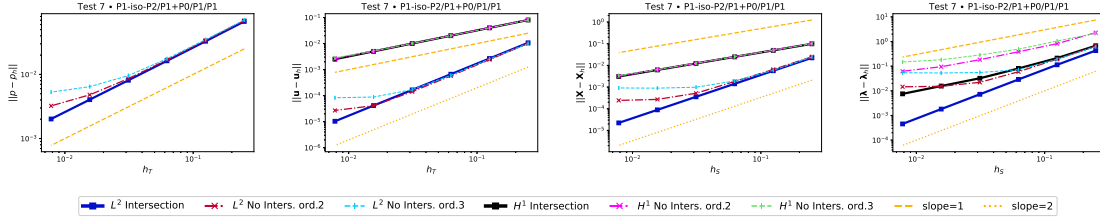


FIGURE 16. Convergence plots of Test 7 with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1 + \mathcal{P}_0/\mathcal{P}_1/\mathcal{P}_1$

6.13. Test 8, square. This test is a variation of Test 2 and 3: we consider the same geometry but in the case of a non trivial mapping. Indeed, the solid reference domain is chosen to be the unit square $\mathcal{B} = [0, 1]^2$, while the actual body is an another square, precisely $\Omega^s = [-0.62, 1.38]^2$. We use again uniform meshes (right and left oriented respectively) for both $\Omega = [-2, 2]^2$ and \mathcal{B} . The initial mapping $\bar{\mathbf{X}}$ is defined by

$$\bar{\mathbf{X}}(x, y) = (-0.62 + 2x, -0.62 + 2y),$$

and the solutions are those presented in (31).

The results, shown in Figures 17 and 18, are in agreement with Test 7.

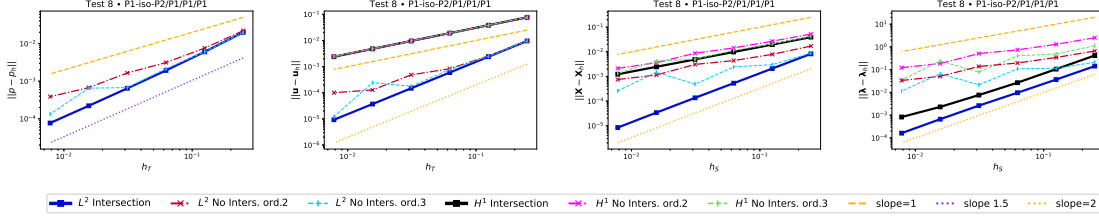


FIGURE 17. Convergence plots of Test 8 with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1/\mathcal{P}_1/\mathcal{P}_1$

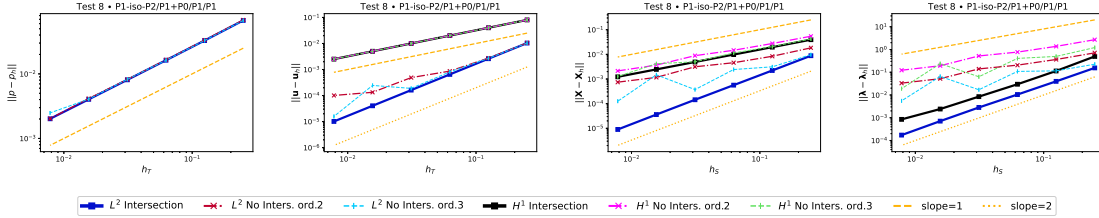


FIGURE 18. Convergence plots of Test 8 with $\mathcal{P}_1 - iso - \mathcal{P}_2/\mathcal{P}_1 + \mathcal{P}_0/\mathcal{P}_1/\mathcal{P}_1$

7. CONCLUSIONS

In the opening sections of this paper, we recalled the formulation for fluid-structure interaction problem introduced in [6]: this approach, born in the spirit of the fictitious domain approach, is based on the use of a distributed Lagrange multiplier with the aim of enforcing the motion condition.

In particular, the main contribution of this paper is given by the comparison of two approaches to be used for the assembling of the interface matrix between fluid and structure: the first approach is based on the computation of the intersection between the solid and the fluid meshes, while the second one consists in computing the entries of the matrix with a direct integration on each solid element. Several numerical tests showed that the optimal convergence rate is reached only when the intersection is computed. On the other hand, when the intersection of the two meshes is not computed, the method is not fully performing and the increase of precision of the quadrature rule does not produce any significant improvement.

REFERENCES

- [1] Frédéric Alauzet, Benoit Fabrèges, Miguel A Fernández, and Mikel Landajuela. Nitsche-XFEM for the coupling of an incompressible fluid with immersed thin-walled structures. *Computer Methods in Applied Mechanics and Engineering*, 301:300–335, 2016.

- [2] Michel Bercovier and Olivier Pironneau. Error estimates for finite element method solution of the Stokes problem in the primitive variables. *Numerische Mathematik*, 33(2):211–224, 1979.
- [3] Daniele Boffi, Franco Brezzi, Michel Fortin, et al. *Mixed finite element methods and applications*, volume 44. Springer, 2013.
- [4] Daniele Boffi, Nicola Cavallini, Francesca Gardini, and Lucia Gastaldi. Local mass conservation of Stokes finite elements. *Journal of scientific computing*, 52(2):383–400, 2012.
- [5] Daniele Boffi, Nicola Cavallini, and Lucia Gastaldi. Finite element approach to immersed boundary method with different fluid and solid densities. *Mathematical Models and Methods in Applied Sciences*, 21(12):2523–2550, 2011.
- [6] Daniele Boffi, Nicola Cavallini, and Lucia Gastaldi. The finite element immersed boundary method with distributed Lagrange multiplier. *SIAM Journal on Numerical Analysis*, 53(6):2584–2604, 2015.
- [7] Daniele Boffi and Lucia Gastaldi. A fictitious domain approach with Lagrange multiplier for fluid-structure interactions. *Numerische Mathematik*, 135(3):711–732, 2017.
- [8] Daniele Boffi and Lucia Gastaldi. Existence, uniqueness, and approximation of a fictitious domain formulation for fluid-structure interactions. *arXiv preprint arXiv:2104.13624*, 2021.
- [9] Daniele Boffi and Lucia Gastaldi. On the existence and the uniqueness of the solution to a fluid-structure interaction problem. *Journal of Differential Equations*, 279:136–161, 2021.
- [10] Daniele Boffi, Lucia Gastaldi, Luca Heltai, and Charles S Peskin. On the hyper-elastic formulation of the immersed boundary method. *Computer Methods in Applied Mechanics and Engineering*, 197(25-28):2210–2231, 2008.
- [11] Daniele Boffi, Lucia Gastaldi, and Sebastian Wolf. Higher-order time-stepping schemes for fluid-structure interaction problems. *Discrete & Continuous Dynamical Systems-Series B*, 25(10), 2020.
- [12] Jan Brezina and Pavel Exner. Fast algorithms for intersection of non-matching grids using Plücker coordinates. *Computers & Mathematics with Applications*, 74(1):174–187, 2017.
- [13] Erik Burman and Miguel A Fernández. An unfitted Nitsche method for incompressible fluid–structure interaction using overlapping meshes. *Computer Methods in Applied Mechanics and Engineering*, 279:497–514, 2014.
- [14] Yu-Chung Chang, TY Hou, B Merriman, and Stanley Osher. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *Journal of computational Physics*, 124(2):449–464, 1996.
- [15] Jean Donéa, P Fasoli-Stella, and S Giuliani. Lagrangian and Eulerian finite element techniques for transient fluid-structure interaction problems. 1977.
- [16] Jean Donea, Antonio Huerta, J-Ph Ponthot, and Antonio Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian methods. *Encyclopedia of computational mechanics*, 2004.
- [17] PE Farrell and JR Maddison. Conservative interpolation between volume meshes by local Galerkin projection. *Computer Methods in Applied Mechanics and Engineering*, 200(1-4):89–100, 2011.
- [18] Christophe Geuzaine and Jean-Francois Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009.
- [19] Roland Glowinski, Tsorng-Whay Pan, Todd I Hesla, Daniel D Joseph, and Jacques Periaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow. *Journal of computational physics*, 169(2):363–426, 2001.
- [20] Roland Glowinski, Tsorng-Whay Pan, and Jacques Periaux. 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. *Comptes Rendus de l’Académie des Sciences-Series I-Mathematics*, 324(3):361–369, 1997.
- [21] Cyrill W Hirt, Anthony A Amsden, and JL Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of computational physics*, 14(3):227–253, 1974.

- [22] Thomas JR Hughes, Wing Kam Liu, and Thomas K Zimmermann. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Computer methods in applied mechanics and engineering*, 29(3):329–349, 1981.
- [23] Rolf Krause and Patrick Zulian. A parallel approach to the variational transfer of discrete fields between arbitrarily distributed unstructured finite element meshes. *SIAM Journal on Scientific Computing*, 38(3):C307–C333, 2016.
- [24] Jacques Louis Lions and Enrico Magenes. *Non-homogeneous boundary value problems and applications: Vol. 1*, volume 181. Springer Science & Business Media, 2012.
- [25] Charles S. Peskin. Flow patterns around heart valves: a numerical method. *Journal of computational physics*, 10(2):252–271, 1972.
- [26] Charles S. Peskin. The immersed boundary method. *Acta numerica*, 11:479–517, 2002.

COMPUTER, ELECTRICAL AND MATHEMATICAL SCIENCES AND ENGINEERING DIVISION, KING ABDULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY, THUWAL 23955, SAUDI ARABIA AND DIPARTIMENTO DI MATEMATICA ‘F. CASORATI’, UNIVERSITY OF PAVIA, VIA FERRATA 1, 27100, PAVIA, ITALY

Email address: `daniele.boffi@kaust.edu.sa`

URL: `kaust.edu.sa/en/study/faculty/daniele-boffi`

COMPUTER, ELECTRICAL AND MATHEMATICAL SCIENCES AND ENGINEERING DIVISION, KING ABDULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY, THUWAL 23955, SAUDI ARABIA AND DIPARTIMENTO DI MATEMATICA ‘F. CASORATI’, UNIVERSITY OF PAVIA, VIA FERRATA 1, 27100, PAVIA, ITALY

Email address: `fabio.credali@kaust.edu.sa`

DIPARTIMENTO DI INGEGNERIA CIVILE, ARCHITETTURA, TERRITORIO, AMBIENTE E DI MATEMATICA, UNIVERSITÀ DEGLI STUDI DI BRESCIA, VIA BRANZE 43, 25123, BRESCIA, ITALY

Email address: `lucia.gastaldi@unibs.it`

URL: `lucia-gastaldi.unibs.it`