

A Weighted Shifted Boundary Method for Immersed Moving Boundary Simulations of Stokes' Flow

Danjie Xu^a, Oriol Colomés^{a,b}, Alex Main^{a,c}, Kangan Li^{a,e}, Nabil M. Atallah^{a,f}, Nabil Abboud^d, Guglielmo Scovazzi^{a,*}

^aDepartment of Civil and Environmental Engineering, Duke University, Durham, NC 27708, USA

^bFaculty of Civil Engineering and Geosciences, Delft University of Technology, Stevinweg 1, 2628 CN, Delft, The Netherlands

^cAnsys, Canonsburg, PA 15317, USA

^dAnsys, Atlanta, GA 30361, USA

^eCollege of Earth and Mineral Sciences, The Pennsylvania State University, University Park, PA 16801, USA

^fLawrence Livermore National Laboratory, Livermore, CA 94550, USA

Abstract

The Shifted Boundary Method (SBM) belongs to the class of unfitted (or immersed, or embedded) finite element methods, and relies on reformulating the original boundary value problem over a surrogate (approximate) computational domain. The surrogate domain is constructed so as to avoid cut cells and the associated problematic implementation and numerical integration issues. Accuracy is maintained by modifying the original boundary conditions using Taylor expansions: hence the name of the method, that *shifts* the *location* and *values* of the boundary conditions. In this article, we extend the SBM to the simulation of incompressible Stokes flow, by appropriately weighting its variational form with the elemental volume fraction of active fluid. This approach allows to drastically reduce spurious pressure oscillations in time, which are produced if the total volume of active fluid were to change abruptly over a time step. The proposed Weighted SBM (W-SBM) exactly preserves states of hydrostatic equilibrium, and induces small mass and momentum conservation errors, which converge as the grid is refined. This is in analogy to cutFEMs and related unfitted approaches, which rely on an affine representation of cut boundaries. We demonstrate the robustness and accuracy of the proposed method with an extensive suite of two-dimensional tests.

Keywords: immersed boundary method; unfitted finite element method; approximate boundary; moving boundaries; computational fluid dynamics; Shifted Boundary Method.

1. Introduction

Flow problems involving moving boundaries are common in nature and have a variety of practical applications, such as the design of powertrains, propulsion/wind energy systems, fluid pumps, biomechanical devices, flexible and flapping wings, to name a few. In such applications, the moving shapes to be simulated might have complex geometry, causing the mesh generation process to be time-consuming and labor-intensive. If imperfect, non watertight representations are used for the geometry (e.g., Standard Tessellation Language), meshing may result unfeasible. In this context, unfitted (or immersed, or embedded) finite elements play an important role because they vastly reduce the meshing time and they in general make meshing more flexible.

Next, we present a brief and possibly incomplete account of the many contributions on immersed/embedded methods for moving boundary simulations, with specific focus on finite element methods. The earliest examples of Immersed Boundary Methods date back to the seminal work of Noh [1] and Peskin [2, 3], in the context of finite difference and finite volume methods. Particularly, [2, 3] introduce a finite difference/volume scheme in which the geometry to be simulated is immersed in a pre-existing grid and the effect of the boundary is mimicked by forcing

*Corresponding author: Guglielmo Scovazzi. Email: guglielmo.scovazzi@duke.edu, Phone: +1 (919) 660-5075, Fax: +1 (919) 660-5219

Email addresses: danjie.xu@duke.edu (Danjie Xu), j.o.colomesgene@tudelft.nl (Oriol Colomés), geoffrey.main@duke.edu (Alex Main), kangan.li@duke.edu (Kangan Li), atallah1@llnl.gov (Nabil M. Atallah), nabil.abboud@duke.edu (Nabil Abboud), guglielmo.scovazzi@duke.edu (Guglielmo Scovazzi)

terms in the governing equations, in the form of discrete, smoothed, Dirac delta functions. In this approach, a fictitious fluid flow is computed also inside immersed bodies, to avoid computational complexities in bookkeeping the active fluid cells in the computational domain. Recent interpretations of this approach in the context of the finite element method can be found in [4, 5].

In the later 1900s and early 2000s, new immersed method appeared in the finite volume community, denoted Direct Forcing Immersed Boundary Methods [6-16]. In these approaches, boundary conditions are applied by converting strong enforcement of the no-slip boundary condition into force terms in the discrete equations. These methods do not introduce a fictitious fluid inside the immersed bodies, and for this reason lead to discretizations with improved accuracy. Direct Forcing Immersed Boundary Methods were developed as an alternative to overset grids [17-22], although the solution/mesh tying algorithms in overset methods have some analogies with these approaches.

Around the 2000s, a series of new unfitted finite element methods [23-57] also appeared, which, like Direct Forcing Immersed Boundary Methods, employ a “sharp interface approximation.” In unfitted finite element methods, boundary conditions are weakly enforced through Lagrange multipliers or Nitsche’s method [58]. Most unfitted finite element methods rely on computing intersections between the boundaries of the geometrical shapes to be simulated and a background grid, resulting in computational domains which involve cut elements, from which the naming cutFEM [41]. This typically results in more complicated data structures for the management of the unknown and the integration over cut cells. If particularly small cut cells are produced, unfitted methods may produce numerical instabilities that need stabilization [29] or linear system of equations that have poor condition numbers. A similar challenge exists for Direct Forcing Immersed Boundary Methods, and also Immersed Boundary Methods, although with slightly different technical interpretations.

To obviate the occurrence of these numerical instabilities in unfitted finite element methods, Burman [29] introduced the *ghost penalty method*, in which the variational form is stabilized by introducing a penalization of the solution gradients at the interface separating cut and uncut elements. This method was applied to the Stokes problem in [59], to the Navier-Stokes equations in [60], and to two-phase flow in [40]. A different approach introduced for B-spline variational formulations, known as *Extended B-splines* [23, 25], involves eliminating cut B-splines with small support via an extrapolation procedure. This technique was applied to the Navier-Stokes equations for moving boundary problems in [35] and [36].

One additional challenge for the classes of methods just mentioned is that they require the geometric construction of the partial elements cut by the embedded boundary, typically a complex and computationally intensive process. Since some sort of adaptive quadrature is used [26, 27], it is often the case that a non-negligible portion of the overall wall-clock time for a simulation is spent handling the embedded/immersed boundary.

An alternative to unfitted finite elements can be found in the Fixed-Mesh ALE Method [61-64], in which the grid near the immersed boundaries is repositioned using a local Arbitrary Lagrangian Eulerian approach, combined with remeshing. Another alternative is the Method of Universal Meshes [65-68], which utilizes dynamic remeshing around a baseline fixed grid, in close proximity of the immersed boundary. These last two methodologies are not immersed approaches in a strict sense, since they use remeshing to accommodate the motion of immersed bodies across a fixed grid.

More recently, the Shifted Boundary Method (SBM) was proposed as an alternative to existing unfitted finite element methods. The SBM broadly falls in the class of immersed (or embedded, or unfitted) finite element methods, and more specifically in the class of approximate boundary methods. In the SBM, the location where boundary conditions are applied is *shifted* from the true to a surrogate boundary, which is specifically chosen so as to avoid cut cells. At the same time, modified (*shifted*) boundary conditions are applied in order to avoid a reduction in the convergence rates of the overall formulation. In fact, if the boundary conditions associated to the true domain are not appropriately modified on the surrogate domain, only first-order convergence is to be expected. The appropriate (modified) boundary conditions are then applied weakly, using a Nitsche strategy. This process yields a method which is simple, robust, accurate, efficient, and free from any of the problematic issues associated with cut elements.

The SBM was proposed for the Poisson and Stokes flow problems in [69] and was generalized in [70] to the advection-diffusion and Navier-Stokes equations, in [71] to hyperbolic conservation laws, and in [72] to the equations of static linear elasticity. An extension of the SBM in conjunction with high-order gradient approximations was presented in [73], and the benefits of its application in the context of reduced order modeling was analyzed in [74-76]. Further rigorous mathematical analysis was pursued in [77, 78] for the Poisson and the Stokes flow problems, and in [79] for the treatment of general domains with corners. A high-order version of the SBM and its numerical

analysis was presented in [80], for the Poisson and Stokes flow equations, and a penalty-free version of the same was analyzed in [81], for Poisson and linear elasticity problems. Li *et al.* further extended the SBM to problems with internal interfaces, under the names of Shifted Interface Method [82] and Shifted Fracture Method [83–85]. The authors of [72, 81] also reported an analysis showing that SBM formulations of the Poisson and linear elasticity equations produce very similar condition numbers to body-fitted FEM formulations for similar grid resolutions.

When the SBM is applied to the incompressible Navier-Stokes equations with moving boundaries [86], one additional complication is that the pressure may spuriously oscillate in time (but not in space!), as the result of an inaccurate numerical bookkeeping of the total volume of active fluid [9, 10, 17, 18]. Kim *et al.* [10] added a mass source/sink term, together with momentum forcing in Navier-Stokes equations, to compensate for this inexact geometric volume conservation. Liao *et al.* [12] investigated the relationship between the internal numerical approximation of the solid object and the external behavior of flow. Lee *et al.* [11] found that the oscillations decrease as mesh size decreases or as computational time step size increases. Seo and Mittal [9] pointed out that the main cause of the spurious pressure oscillations in time is the inaccurate mathematical modeling of the fluid near the surface of the moving object and introduced the “virtual cell-merging technique” to mitigate this issue. Many other works also attempted to alleviate the spurious pressure oscillations in the simulation of moving boundary flow problems. A list far from being comprehensive includes [13–16, 36, 87, 88].

In this work, we propose a modification of the original SBM method [69, 70], in which we “weight” the SBM variational form by the volume fraction of fluid in every element. This approach preserves the total volume of active fluid to a much higher degree of accuracy than the SBM, and, consequently, delivers a drastic reduction of the pressure oscillations in time, while still maintaining a relative simple implementation.

This variant of the Shifted Boundary Method, named the Weighted Shifted Boundary Method (WSBM), was proposed in [86] for general Neumann (i.e., traction) boundary conditions with some focused applications to free-surface flows. The present work is instead exclusively focused on the development and implementation of the WSBM with Dirichlet (e.g., no-slip) boundary conditions. The present work and [86] represent a comprehensive description of the derivation and implementation of the WSBM in the context of the most common boundary conditions encountered in computational fluid dynamics, and more generally, in computational mechanics.

We show that the mass and momentum conservation errors produced by the WSBM are typically small, in a relative sense, and converge as the grid is refined. In addition, we prove and numerically demonstrate that the WSBM can represent states of hydrostatic equilibrium exactly (i.e., within machine precision). We demonstrate in a number of numerical tests that the WSBM drastically reduces pressure oscillations.

The rest of this paper is organized as follows: Section 2 introduces the governing equations of the Stokes problem; Section 3 introduces the general formalism of the SBM framework; Section 4 includes the derivation of the WSBM, its implementation, and its salient algorithmic properties; Section 5 contains an extensive series of numerical tests; and Section 6 summarizes conclusions.

2. Strong form of the time-dependent Stokes Flow equations

The time-dependent Stokes flow is an important example of application of the WSBM to moving-boundary problems involving incompressible flow constraints. Let Ω be the physical domain where the governing equations of the problem at hand are defined, a bounded and connected open region in $\mathbb{R}^n$ ($n = 2$ or 3). Let $\Gamma = \partial\Omega$ be the Lipschitz boundary of Ω (the blue curve in the sketch of Figures 1a and 1b), composed by C^2 curves (in two dimensions) or surfaces (in three dimensions) that intersect at a finite number of vertices or edges. Let $\mathbf{n}$ denote the outward-oriented unit normal vector to Γ (see again Figure 1b).

The governing equations for viscous incompressible flow are constituted by a momentum update and a mass continuity equation, namely:

$$\rho \mathbf{u}_{,t} + \nabla p - \nabla \cdot (2\mu \boldsymbol{\varepsilon}(\mathbf{u})) - \rho \mathbf{b} = \mathbf{0}, \quad \forall \mathbf{x} \in \Omega, \quad (1a)$$

$$\nabla \cdot \mathbf{u} = 0, \quad \forall \mathbf{x} \in \Omega, \quad (1b)$$

where $\mathbf{u}_{,t} = \partial \mathbf{u} / \partial t$ is the time derivative of the velocity, $\boldsymbol{\varepsilon}(\mathbf{u}) = 1/2(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$ is the strain rate (i.e., the symmetric part of the velocity gradient), $\mu > 0$ is the dynamic viscosity, p is the pressure (playing the role of a Lagrange multiplier enforcing the continuity constraint $\nabla \cdot \mathbf{u}$), ρ is the density, and $\mathbf{b}$ is a body force per unit mass (e.g., the gravitational

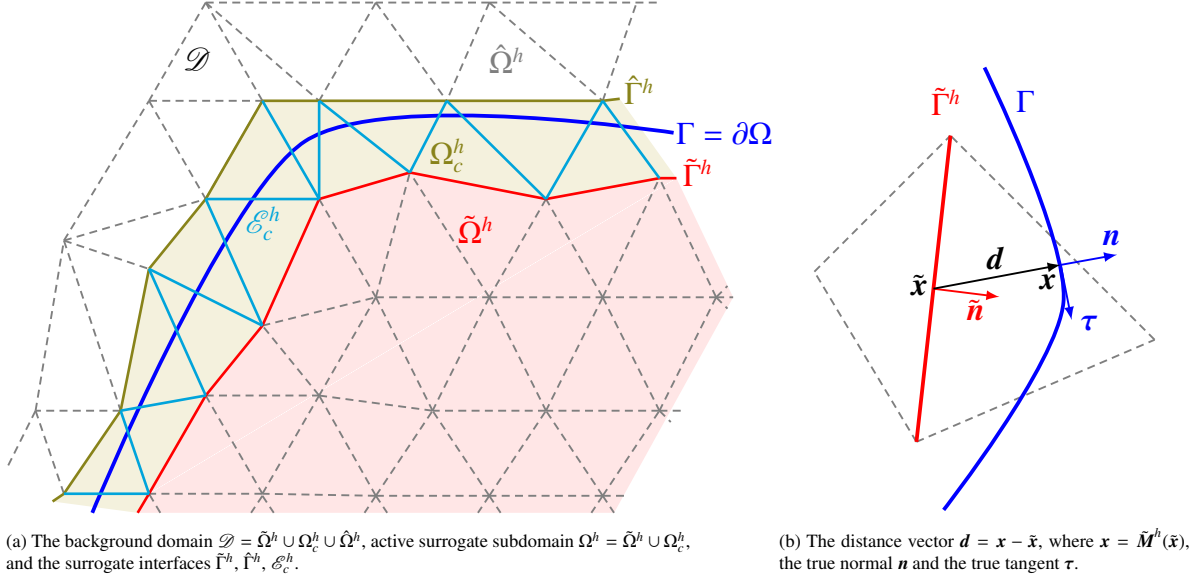


Figure 1: The surrogate domain, its boundary, and the distance vector d .

acceleration). To complete the setup of the Stokes problem, initial conditions need to be specified for the velocity, namely,

$$u(x, 0) = u_0(x) . \quad (1c)$$

and boundary conditions need to be applied. The boundary is partitioned as the non-overlapping union $\Gamma = \Gamma_D \cup \Gamma_N$ of a Dirichlet boundary Γ_D and a Neumann boundary Γ_N . Specifically, the following boundary conditions are imposed:

$$u = u_D , \quad \forall x \text{ on } \Gamma_D , \quad (1d)$$

$$\sigma n = t_N , \quad \forall x \text{ on } \Gamma_N , \quad (1e)$$

where $\sigma = -pI + 2\mu \varepsilon(u)$, u_D is the value of the velocity on the Dirichlet boundary Γ_D and t_N is the vector-valued normal stress on the Neumann boundary Γ_N . In computational fluids dynamics, it is more common to refer to the Dirichlet boundary condition (1d) as a *no-slip* boundary condition. Note that Dirichlet conditions can also be used to enforce inflow conditions, as done in some of the numerical experiments in Section 5.5 and Section 5.6. Similarly, the Neumann boundary condition (1e) is commonly referred to as a *traction* boundary condition. Traction boundary conditions are often used at outflow boundaries or when free-surfaces are present. With a slight abuse of notation, the terminology “Dirichlet” and “no-slip” boundary condition will be used interchangeably in what follows, and similarly for “Neumann” and “traction” boundary condition.

3. General notation

3.1. The physical and surrogate domains

We consider a *background* closed domain $\mathcal{D}$ in which Ω is *immersed*, that is $\text{clos}(\Omega) \subseteq \mathcal{D}$, and we introduce a family $\mathcal{T}^h$ of admissible and shape-regular triangulations of $\mathcal{D}$. The set of all edges (in two dimensions) of faces (in three dimensions) of the triangulation $\mathcal{T}^h$ is indicated as $\mathcal{E}^h$. We indicate by h_T the diameter of an element $T \in \mathcal{T}^h$ and by h the piecewise constant function in $\mathcal{D}$ such that $h|_T = h_T$ for all $T \in \mathcal{T}^h$. Then, we restrict each triangulation by selecting those elements that are *strictly* contained in the closure of Ω , denoted $\text{clos}(\Omega)$, i.e., we form

$$\tilde{\mathcal{T}}^h := \{T \in \mathcal{T}^h : T \subset \text{clos}(\Omega)\} . \quad (2)$$

$\tilde{\mathcal{T}}^h$ identifies the *inner surrogate domain* (the red-shaded area in the sketch of Figure 1a)

$$\tilde{\Omega}^h := \text{int} \left(\bigcup_{T \in \tilde{\mathcal{T}}^h} T \right) \subseteq \Omega. \quad (3)$$

Obviously, $\tilde{\mathcal{T}}^h$ is an admissible and shape-regular triangulation of $\tilde{\Omega}^h$. The set of edges/faces of the mesh in $\tilde{\Omega}^h$ is indicated as $\tilde{\mathcal{E}}^h = \{E \in \mathcal{E}^h : E \subset \tilde{\Omega}\}$ and *inner surrogate boundary* as $\tilde{\Gamma}^h := \partial\tilde{\Omega}^h$ (the red polygonal line in Figures 1a and 1b), with outward-oriented unit normal vector $\tilde{\mathbf{n}}$. Note that $\tilde{\Gamma}^h \subset \tilde{\mathcal{E}}^h$ and that $\tilde{\Gamma}^h$ may coincide with a portion of the boundary Γ , if the grid is exactly fitted to it. We also define the set of edges/faces of the grid that are internal to $\tilde{\Omega}^h$, that is $\tilde{\mathcal{E}}_i^h = \tilde{\mathcal{E}}^h \setminus \tilde{\Gamma}^h$.

Analogously, we can define the *outer triangulation*

$$\hat{\mathcal{T}}^h := \{T \in \mathcal{T}^h : T \subset \text{clos}(\mathcal{D} \setminus \Omega)\}, \quad (4)$$

and the *outer inactive domain*

$$\hat{\Omega}^h := \text{int} \left(\bigcup_{T \in \hat{\mathcal{T}}^h} T \right) \subseteq \mathcal{D} \setminus \Omega, \quad (5)$$

with a corresponding *outer surrogate boundary* $\hat{\Gamma}^h$ (the olive-green polygonal line in Figure 1a), with *inward*-oriented normal $\hat{\mathbf{n}}$ (i.e., pointing away from Ω). It will also be important to define the subset of elements that are cut by Γ , namely

$$\mathcal{T}_c^h := \{T \in \mathcal{T}^h : T \cap \Gamma \neq \emptyset\}, \quad (6)$$

and the corresponding *cut domain*

$$\Omega_c^h := \text{int} \left(\bigcup_{T \in \mathcal{T}_c^h} T \right), \quad (7)$$

and $\mathcal{E}_c^h = \{E \in \mathcal{E}^h \setminus \Gamma : E \cap \Gamma \neq \emptyset\}$, i.e., the union of edges/faces (in two/three dimensions, resp.) that are cut by Γ but do not belong to Γ (marked in light blue in Figure 1a).

In the context of the algorithm described next, the *active computational domain* is $\Omega^h(t) = \tilde{\Omega}^h(t) \cup \Omega_c^h(t)$ and may change over time. In other words, $\Gamma(t)$ is a moving boundary, which means that the sets $\tilde{\Gamma}^h(t)$, $\hat{\Gamma}^h(t)$ and $\mathcal{E}_c^h(t)$ may also change over time. In what follows, for the sake of simplicity, we will omit the explicit dependence on time when indicating these sets.

3.2. Maps from the surrogate boundaries to the physical (true) boundary

We now define the map

$$\tilde{\mathbf{M}}^h : \tilde{\Gamma}^h \rightarrow \Gamma, \quad (8a)$$

$$\tilde{\mathbf{x}} \mapsto \mathbf{x}, \quad (8b)$$

from a point $\tilde{\mathbf{x}} \in \tilde{\Gamma}^h$ on the inner surrogate boundary to a point $\mathbf{x} \in \Gamma$ on the physical, or true, boundary. $\tilde{\mathbf{M}}^h$ can be constructed using the *closest-point projection* of points in $\tilde{\Gamma}^h$ onto Γ , as shown in Figure 1b, or with analogous concepts, such as level sets. In particular, it will become very important to characterize the map $\tilde{\mathbf{M}}^h$ through a distance vector function

$$\mathbf{d}_{\tilde{\mathbf{M}}^h}(\tilde{\mathbf{x}}) = \mathbf{x} - \tilde{\mathbf{x}} = [\tilde{\mathbf{M}}^h - \mathbf{I}](\tilde{\mathbf{x}}), \quad (9)$$

where I is the identity map in dimension d . Equivalently, we can find a map $\hat{M}^h : \hat{\Gamma}^h \rightarrow \Gamma$, defined by means of the distance $\mathbf{d}_{\hat{M}^h}(\hat{\mathbf{x}}) = \mathbf{x} - \hat{\mathbf{x}}$, from a point on the outer surrogate, $\hat{\mathbf{x}} \in \hat{\Gamma}^h$, to its closest-point projection on the true boundary, $\mathbf{x} \in \Gamma$. Finally, it is also possible to define a map $M_c^h : E \in \mathcal{E}_c^h \rightarrow \Gamma$ on points of the cut edges, with relative distance vector $\mathbf{d}_{M_c^h}$. For the sake of simplicity and whenever there is no chance of confusion, the “tilde” and “hat” accents and the subscript c will be omitted from the map symbols $\tilde{M}^h$, $\hat{M}^h$ and M_c^h , and will simply write “ M^h .” Similarly, we will omit the subscripts of $\mathbf{d}_{\tilde{M}^h}$, $\mathbf{d}_{\hat{M}^h}$ and $\mathbf{d}_{M_c^h}$, and write “ $\mathbf{d}$.”

Remark 1. In spite of the segmented/faceted nature of the surrogate boundaries of $\tilde{\Gamma}^h$, $\hat{\Gamma}^h$ and $\bigcup\{E \in \mathcal{E}_c^h\}$, the closest-point projections described above are typically well defined if the boundary Γ is Lipschitz. For a portion of the boundary where Γ is smooth and only one type of boundary condition is applied, the closest-point projection algorithm will pick $\mathbf{d}$ in the direction of the true normal $\mathbf{n}$, that is $\mathbf{d} = \|\mathbf{d}\| \mathbf{n}$. We refer the reader to the recent analysis in [79], for the general case of general domains with corners and edges.

Remark 2. Other strategies in the definition of M^h are possible, among which a level set description of the true boundary, in which $\mathbf{d}$ is defined by means of a distance function and its gradient. In our case, we calculate M^h and $\mathbf{d}$ using tools developed in [89] and references therein. Most importantly, the methods that we are about to introduce do not depend on how M^h is constructed.

3.3. Extensions of fields from the surrogate to the true interfaces and Taylor expansions

Through the map M^h , it is possible to define the extension $\mathbb{E}\psi$ on $\tilde{\Gamma}^h$ of a function ψ initially defined only on Γ , as

$$\mathbb{E}\psi(\tilde{\mathbf{x}}) := \psi(M^h(\tilde{\mathbf{x}})) . \quad (10)$$

For example, the unit normal vector $\mathbf{n}$ and unit tangential vectors $\boldsymbol{\tau}_i$ ($1 < i < n_d - 1$) to the true boundary Γ can be extended to the surrogate boundary $\tilde{\Gamma}^h$ as follows:

$$\mathbb{E}\mathbf{n}(\tilde{\mathbf{x}}) := \mathbf{n}(M^h(\tilde{\mathbf{x}})) , \quad (11a)$$

$$\mathbb{E}\boldsymbol{\tau}_i(\tilde{\mathbf{x}}) := \boldsymbol{\tau}_i(M^h(\tilde{\mathbf{x}})) . \quad (11b)$$

In what follows, with the purpose of simplifying the notation, we will omit the symbol $\mathbb{E}$ in front of $\mathbf{n}$ and $\boldsymbol{\tau}_i$, whenever this does not cause confusion. Therefore when we write $\mathbf{n}(\tilde{\mathbf{x}})$ we actually mean $\mathbb{E}\mathbf{n}(\tilde{\mathbf{x}})$, and similarly for $\boldsymbol{\tau}_i(\tilde{\mathbf{x}})$ and $\mathbb{E}\boldsymbol{\tau}_i(\tilde{\mathbf{x}})$. Analogous extensions can be defined on $\hat{\Gamma}^h$ or $\mathcal{E}_c^h$. These constructions are particularly useful if we consider the solution u to a partial differential equation and, for example, we desire to define an extension of a boundary condition from the true boundary Γ to a surrogate boundary $\tilde{\Gamma}^h$ (or $\hat{\Gamma}^h$, or $\mathcal{E}_c^h$). In the case of Dirichlet boundary conditions, the following Taylor expansion formula centered at $\tilde{\mathbf{x}} \in \tilde{\Gamma}^h$ holds for $\mathbf{x} = M^h(\tilde{\mathbf{x}}) \in \Gamma$:

$$\begin{aligned} 0 &= u(\tilde{\mathbf{x}}) + \nabla u(\tilde{\mathbf{x}}) \cdot (\mathbf{x} - \tilde{\mathbf{x}}) - u_D(\mathbf{x}) + O(\|\mathbf{x} - \tilde{\mathbf{x}}\|^2) \\ &= u(\tilde{\mathbf{x}}) + \nabla u(\tilde{\mathbf{x}}) \cdot (M^h(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}) - u_D(M^h(\tilde{\mathbf{x}})) + O(\|M^h(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}\|^2) \\ &= u(\tilde{\mathbf{x}}) + \nabla u(\tilde{\mathbf{x}}) \cdot \mathbf{d}(\tilde{\mathbf{x}}) - \mathbb{E}u_D(\tilde{\mathbf{x}}) + O(\|\mathbf{d}(\tilde{\mathbf{x}})\|^2) , \end{aligned} \quad (12)$$

where $u_D(\mathbf{x})$ is the value of the Dirichlet boundary condition on u . The last in the chain of equalities above can be used to develop a strategy for imposing boundary conditions in immersed domain methods. Intrinsically, this approach is at most second-order accurate, unless additional terms in the Taylor expansion are included [80, 81]. We introduce the short-hand notation for the Taylor expansion by means of the *shift* operator

$$S_d u(\tilde{\mathbf{x}}) := u(\tilde{\mathbf{x}}) + \nabla u(\tilde{\mathbf{x}}) \cdot \mathbf{d}(\tilde{\mathbf{x}}) . \quad (13)$$

In the derivation of the computational method that follow, we will make the following

Assumption 1. The Neumann (i.e., traction) boundary is body-fitted, that is $\tilde{\Gamma}_N^h = \Gamma_N$.

Remark 3. Assumption [1](#) is certainly a simplification, but as mentioned in the introduction, the WSBM was already applied to moving Neumann boundaries in [\[86\]](#). Hence we prefer to focus exclusively on immersed moving boundaries of Dirichlet type.

3.4. Infinite dimensional function spaces and integrals

Throughout the paper, we will use the Sobolev spaces $H^m(\Omega) = W^{m,2}(\Omega)$ of index of regularity $m \geq 0$ and index of summability 2, equipped with the (scaled) norm

$$\|v\|_{H^m(\Omega)} = \left(\|v\|_{L^2(\Omega)}^2 + \sum_{k=1}^m \|l(\Omega)^k \mathbf{D}^k v\|_{L^2(\Omega)}^2 \right)^{1/2}, \quad (14)$$

where $\mathbf{D}^k$ is the k th-order spatial derivative operator and $l(A) = (\text{meas}(A))^{1/n_d}$ is a characteristic length of the domain A . Note that $H^0(\Omega) = L^2(\Omega)$. As usual, we use a simplified notation for norms and semi-norms, i.e., we set $\|v\|_{m;\Omega} = \|v\|_{H^m(\Omega)}$ and $|v|_{k;\Omega} = \|\mathbf{D}^k v\|_{0;\Omega} = \|\mathbf{D}^k v\|_{L^2(\Omega)}$. In the sequel, $(v, w)_\omega = \int_\omega v w$ denotes the L^2 inner product on a subset $\omega \in \mathcal{T}^h$ and $\langle v, w \rangle_\zeta = \int_\zeta v w$ denotes the L^2 inner product on the subset ζ of $\mathcal{E}^h$.

4. The Weighted Shifted Boundary Method for Stokes flows with moving boundaries

Unfitted method involving integration on cut elements have proved effective in simulating moving boundaries, but also pose challenges:

- a) the presence of small-cut cells can produce numerical instabilities and/or produce poor condition numbers in the ensuing algebraic system;
- b) integrating the variational forms on cut elements may be computationally expensive and sometimes unfeasible if the geometry contains small gaps and overlaps;
- c) in the context of moving boundaries, the issues mentioned in a) and b) are exacerbated by the fact that boundary conditions are enforced on moving surfaces.

Since these aspects emanate from the mere existence of cut elements near the boundary, the idea in [\[69, 70\]](#) was to exclude them altogether from the simulation. Excluding these elements has the effect of moving the location of the moving boundary from Γ to a surrogate location $\tilde{\Gamma}$. Of course, if boundary conditions are naïvely applied on the surrogate boundary, then an $O(h)$ error is introduced.

In what follows, we propose a technique that extends to moving boundaries the ideas introduced in [\[69, 70\]](#) and avoids the reduction of accuracy to first order. The new approach proposed here follows the developments in [\[86\]](#) and has one major difference with respect to the developments in [\[69, 70\]](#), in that integration is not restricted to the surrogate subdomain $\tilde{\Omega}^h$ but instead extends over the active computational domain $\Omega^h := \tilde{\Omega}^h \cup \Omega_c^h$. We stress that the proposed approach does not involve the integration of the variational equations over partially cut cells, with a considerable simplification in the implementation, when compared to other unfitted FEM approaches.

4.1. Rationale

The total volume of a fluid governed by the incompressible Stokes equations is conserved over time. It is important to preserve this property in a numerical method, or at least approximate it to a high degree of accuracy. Significant errors in the total volume budget may result in spurious pressure oscillations in time, since the constant, zeroth-order pressure mode associated with the fluid volume (i.e., the average pressure over the volume) is not correctly captured (see, e.g., the analysis in [\[17, 18, 20\]](#)). We note that these pressure oscillations induce large spurious forces in time, while the pressure field remains smooth in space [\[9\]](#).

A problematic aspect that emerges immediately in the application of the standard SBM [\[69, 70\]](#) to moving boundary problems is the fact that the surrogate boundary $\tilde{\Omega}^h$ is not constant over time. Specifically, as Γ sweeps through Ω , the domain $\tilde{\Omega}^h$ will experience discrete changes due to the activation/deactivation of elements in computations.

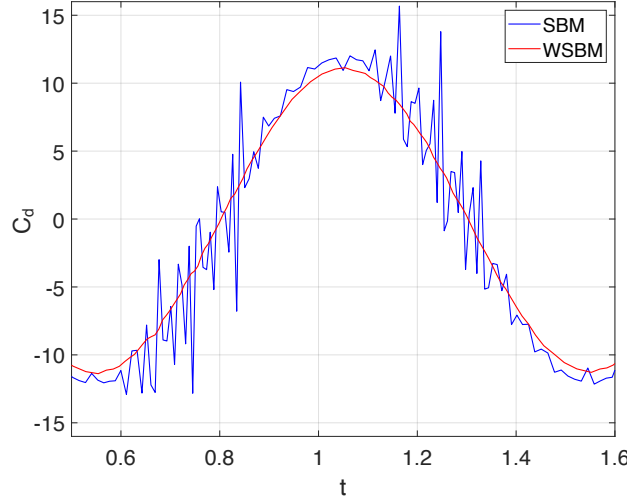


Figure 2: Oscillating cylinder in a fluid at rest [9]. Time-histories of the pressure non-dimensional coefficient C_d defined by (41), for $CFL = 0.1$ (the CFL condition is defined in (44)) and a diameter/mesh-size ratio $D/h = 16$. In blue, the result obtained with the standard SBM, which is affected by large, spurious pressure oscillations. In red, the result using the WSBM, which shows no spurious pressure oscillations.

Hence the standard SBM is prone to pressure oscillations over time, as shown for example in Figure 2 (blue curve), which depicts results from the classical benchmark of a moving cylinder in a flow initially at rest [9]. Full details on this test are presented in Section 5.3.

In what follows, we propose to control pressure oscillations by extending the computational domain beyond $\tilde{\Omega}^h$, that is to $\Omega^h = \tilde{\Omega}^h \cup \Omega_c^h$, and to appropriately *weight* the variational form by the *fluid's volume fraction*, so as to preserve the fluid volume to within the accuracy of the discretization. The net result is a Petrov-Galerkin method that we named Weighted Shifted Boundary Method (WSBM), in which the test function space differs from the trial function space due to the volume fraction weighting.

The WSBM was already successfully applied to Neumann boundary conditions for free-surface flows in [86], but the volume constraint poses considerably more challenges in the case of Dirichlet (no-slip) moving boundaries. In fact, because the pressure field is not imposed at a no-slip moving boundary, it cannot be as easily controlled as in the case of a moving traction boundary.

Figure 2 also shows the results for this improved WSBM method (red curve), and it is evident that spurious pressure oscillations are absent even in the case of no-slip (Dirichlet) boundary conditions. We present next the WSBM for this type of boundary conditions in full detail.

4.2. Variational formulation

For an element $T \in \mathcal{T}^h$ (i.e., $T \subset \Omega$), let the volume fraction of the active (fluid) domain be defined as

$$\alpha_T = \frac{\text{meas}(T \cap \Omega)}{\text{meas}(T)}. \quad (15)$$

Clearly $\alpha_T \in [0, 1]$, and more precisely: $\alpha_T = 1$ if $T \in \tilde{\mathcal{T}}^h$; $\alpha_T = 0$ if $T \in \hat{\mathcal{T}}^h$; and $0 < \alpha_T < 1$ if $T \in \mathcal{T}_c^h$. We also construct the piecewise constant function $\alpha(\mathbf{x})$ in Ω such that $\alpha|_T = \alpha_T$ for all $T \in \mathcal{T}^h$. Let us introduce the spaces of continuous piecewise-linear functions over the active computational domain $\mathbf{V}^h(\Omega^h)$ and $\mathcal{Q}^h(\Omega^h)$, for the velocity and pressure, respectively. Namely:

$$\mathbf{V}^h(\Omega^h) = \left\{ \mathbf{w} \in (\mathcal{C}(\Omega^h))^{n_d} : \mathbf{w}|_T \in \left(\mathcal{P}^1(T) \right)^{n_d}, T \in \tilde{\mathcal{T}}^h \cup \mathcal{T}_c^h \right\}, \quad (16a)$$

$$\mathcal{Q}^h(\Omega^h) = \left\{ q \in \mathcal{C}(\Omega^h) : q|_T \in \mathcal{P}^1(T), T \in \tilde{\mathcal{T}}^h \cup \mathcal{T}_c^h \right\}, \quad (16b)$$

where $\mathcal{P}^1(T)$ and $(\mathcal{P}^1(T))^{n_d}$ are the sets of scalar- and vector-valued piecewise linear functions over the element T , and $\mathcal{C}(\Omega^h)$ and $(\mathcal{C}(\Omega^h))^{n_d}$ are the sets of scalar- and vector-valued continuous functions over the active computational domain $\Omega^h = \tilde{\Omega}^h \cup \Omega_c^h$. $V^h(\Omega^h)$ and $Q^h(\Omega^h)$ will be the basis for the trial spaces to represent the discrete solutions. These spaces need not satisfy any prescribed boundary conditions, which will instead be imposed weakly.

Next, we introduce the α -weighted test spaces $V_\alpha^h(\Omega^h)$ and $Q_\alpha^h(\Omega^h)$, which are discontinuous by construction:

$$V_\alpha^h(\Omega^h) = \{ \mathbf{w}_\alpha \in (L^2(\Omega^h))^{n_d} : \mathbf{w}_\alpha = \alpha \mathbf{w}, \mathbf{w} \in V^h(\Omega^h), \alpha \in \mathcal{P}^0(\tilde{\mathcal{T}}^h \cup \mathcal{T}_c^h) \}, \quad (17a)$$

$$Q_\alpha^h(\Omega^h) = \{ q_\alpha \in L^2(\Omega^h) : q_\alpha = \alpha q, q \in Q^h(\Omega^h), \alpha \in \mathcal{P}^0(\tilde{\mathcal{T}}^h \cup \mathcal{T}_c^h) \}, \quad (17b)$$

where $\mathcal{P}^0(\tilde{\mathcal{T}}^h \cup \mathcal{T}_c^h)$ is the set of globally discontinuous functions that are piecewise-constant over any element $T \in \tilde{\mathcal{T}}^h \cup \mathcal{T}_c^h$. Multiplying the governing equation (1a)-(1b) by $\mathbf{w}_\alpha$ and q_α over a generic element T and integrating by parts, we have

$$\begin{aligned} 0 &= -(\mathbf{w}_\alpha, \nabla \cdot (-p\mathbf{I} + 2\mu \boldsymbol{\varepsilon}(\mathbf{u})))_{\Omega^h} - (\mathbf{w}_\alpha, \rho \mathbf{b})_{\Omega^h} - (q_\alpha, \nabla \cdot \mathbf{u})_{\Omega^h} \\ &= (\boldsymbol{\varepsilon}(\mathbf{w}_\alpha), -p\mathbf{I} + 2\mu \boldsymbol{\varepsilon}(\mathbf{u}))_{\Omega^h} - (\mathbf{w}_\alpha, \rho \mathbf{b})_{\Omega^h} - (q_\alpha, \nabla \cdot \mathbf{u})_{\Omega^h} - \sum_{T \in \tilde{\mathcal{T}}^h \cup \mathcal{T}_c^h} \sum_{E \subset \partial T} \langle \mathbf{w}_\alpha, (-p\mathbf{I} + 2\mu \boldsymbol{\varepsilon}(\mathbf{u})) \mathbf{n}_E \rangle_E \\ &= -(\alpha \nabla \cdot \mathbf{w}, p)_{\Omega^h} + (\alpha \boldsymbol{\varepsilon}(\mathbf{w}), 2\mu \boldsymbol{\varepsilon}(\mathbf{u}))_{\Omega^h} - (\alpha \mathbf{w}, \rho \mathbf{b})_{\Omega^h} - (\alpha q, \nabla \cdot \mathbf{u})_{\Omega^h} \\ &\quad - \langle \llbracket \alpha \mathbf{w} \rrbracket_{1-\gamma}, \llbracket 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) - p\mathbf{I} \rrbracket \rangle_{\tilde{\mathcal{E}}_i^h \cup \mathcal{E}_{c,D}^h \cup \tilde{\Gamma}_D^h} - \langle \llbracket \alpha \mathbf{w} \rrbracket, \llbracket 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) - p\mathbf{I} \rrbracket_\gamma \rangle_{\tilde{\mathcal{E}}_i^h \cup \mathcal{E}_{c,D}^h \cup \tilde{\Gamma}_D^h} \\ &\quad - \langle \alpha \mathbf{w} \otimes \hat{\mathbf{n}}, 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) - p\mathbf{I} \rangle_{\hat{\Gamma}_D^h} - \langle \alpha \mathbf{w} \otimes \mathbf{n}, 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) - p\mathbf{I} \rangle_{\Gamma_N}. \end{aligned} \quad (18)$$

Here $\mathcal{E}_{c,D}^h$ is the set of edges intersected by the (true) boundary where Dirichlet (e.g., no-slip) boundary conditions are enforced. Analogously, $\tilde{\mathcal{E}}_i^h$ indicates the set of the interior edges of the elements belonging to $\tilde{\Omega}$.

Note that we have introduced the average operator

$$\langle \cdot \rangle_\gamma := \gamma^+ (\cdot)^+ + \gamma^- (\cdot)^- \quad (19)$$

for scalars, vectors or tensors, where $\gamma^+ := \alpha^+ / (\alpha^+ + \alpha^-)$, $\gamma^- := \alpha^- / (\alpha^+ + \alpha^-) = 1 - \gamma^+$, and the superscripts $+$ and $-$ indicate the two sides of an edge E . This specific average was proposed in [90], which contains the numerical analysis of the overall WSBM formulation discussed here. As shown in [90], this specific average definition is essential in the proof of numerical stability of the WSBM method. In particular, we define the superscripts $+$ and $-$ such that $\alpha^+ \geq \alpha^-$ and we define $[\alpha] := \alpha^+ - \alpha^- \geq 0$.

Similarly, we have introduced the standard jump operator $\llbracket \cdot \rrbracket$ on an edge E , defined as $\llbracket \boldsymbol{\sigma} \rrbracket := \boldsymbol{\sigma}^+ \mathbf{n}_E^+ + \boldsymbol{\sigma}^- \mathbf{n}_E^-$, $\llbracket \mathbf{v} \rrbracket := \mathbf{v}^+ \cdot \mathbf{n}_E^+ + \mathbf{v}^- \cdot \mathbf{n}_E^-$, and $\llbracket \alpha \rrbracket := \alpha^+ \mathbf{n}_E^+ + \alpha^- \mathbf{n}_E^-$ for tensors, vectors, and scalars, respectively. Next, we require $\langle \llbracket \alpha \mathbf{w} \rrbracket_{1-\gamma}, \llbracket 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) - p\mathbf{I} \rrbracket \rangle_{\tilde{\mathcal{E}}_i^h \cup \mathcal{E}_{c,D}^h \cup \tilde{\Gamma}_D^h} = 0$, that is we weakly enforce the balance of fluxes $\llbracket 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) - p\mathbf{I} \rrbracket = \mathbf{0}$ on $\tilde{\mathcal{E}}_i^h \cup \mathcal{E}_{c,D}^h \cup \tilde{\Gamma}_D^h$. Similarly, we weakly enforce the Neumann (i.e, traction) boundary condition $(2\mu \boldsymbol{\varepsilon}(\mathbf{u}) - p\mathbf{I}) \cdot \mathbf{n} = \mathbf{t}_N$ on Γ_N .

Observe that, $\llbracket \alpha \mathbf{w} \rrbracket = 0$ over $\tilde{\mathcal{E}}_i^h$, since $\alpha^\pm = 1$ and $\mathbf{w}$ is continuous there. Also, for an edge $E \subset \hat{\Gamma}_D^h$, $\alpha^+ > 0$ and $\alpha^- = 0$, so that $[\alpha] = \alpha^+$, $\gamma^+ = 1$, and $\gamma^- = 0$. Consequently, extending by continuity $\mathbf{w}$ beyond Ω^h , we have $\llbracket \alpha \mathbf{w} \rrbracket = \alpha \mathbf{w}$, (since $\mathbf{w}^+ = \mathbf{w}^-$ on $\hat{\Gamma}_D^h$ by the continuity of $\mathbf{w}$), and $\langle \llbracket 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) - p\mathbf{I} \rrbracket_\gamma = 2\mu \boldsymbol{\varepsilon}(\mathbf{u})^+ - p\mathbf{I}$ for any bounded extension of $\boldsymbol{\varepsilon}(\mathbf{u})$ with p continuous. We define the set of edges $\tilde{\mathcal{E}}_{a,D}^h := \mathcal{E}_{c,D}^h \cup \tilde{\Gamma}_D^h$ and $\mathcal{E}_{a,D}^h := \tilde{\mathcal{E}}_{a,D}^h \cup \hat{\Gamma}_D^h$. Hence, we can compactly express (18) as

$$\begin{aligned} 0 &= -(\alpha \nabla \cdot \mathbf{w}, p)_{\Omega^h} + (\alpha \boldsymbol{\varepsilon}(\mathbf{w}), 2\mu \boldsymbol{\varepsilon}(\mathbf{u}))_{\Omega^h} - (\alpha \mathbf{w}, \rho \mathbf{b})_{\Omega^h} - (\alpha q, \nabla \cdot \mathbf{u})_{\Omega^h} \\ &\quad - \langle \llbracket \alpha \rrbracket \otimes \mathbf{w}, \llbracket 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) - p\mathbf{I} \rrbracket_\gamma \rangle_{\tilde{\mathcal{E}}_{a,D}^h} - \langle \alpha \mathbf{w} \otimes \hat{\mathbf{n}}, 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) - p\mathbf{I} \rangle_{\hat{\Gamma}_D^h} - \langle \alpha \mathbf{w}, \mathbf{t}_N \rangle_{\Gamma_N}. \end{aligned} \quad (20)$$

Dirichlet boundary conditions can be enforced using a Nitsche-type approach, akin the interior penalty method of discontinuous Galerkin discretizations. Observing that $\llbracket \alpha \mathbf{v} \rrbracket = \alpha \mathbf{v}$ on $\hat{\Gamma}_D^h$, we can augment (20) with appropriate

boundary terms:

$$\begin{aligned}
0 = & -(\alpha \nabla \cdot \mathbf{w}, p)_{\Omega^h} + (\alpha \boldsymbol{\varepsilon}(\mathbf{w}), 2\mu \boldsymbol{\varepsilon}(\mathbf{u}))_{\Omega^h} - (\alpha \mathbf{w}, \rho \mathbf{b})_{\Omega^h} - (\alpha q, \nabla \cdot \mathbf{u})_{\Omega^h} - \langle \alpha \mathbf{w}, \mathbf{t}_N \rangle_{\Gamma_N} \\
& - \langle \llbracket \alpha \rrbracket \otimes \mathbf{w}, 2\mu \llbracket \boldsymbol{\varepsilon}(\mathbf{u}) \rrbracket_\gamma - p \mathbf{I} \rangle_{\mathcal{E}_{a,D}^h} \\
& - \langle 2\mu \llbracket \boldsymbol{\varepsilon}(\mathbf{w}) \rrbracket_\gamma - q \mathbf{I}, \llbracket \alpha \rrbracket \otimes (\llbracket \mathbf{S}_d \mathbf{u} \rrbracket_\gamma - \mathbf{u}_D) \rangle_{\mathcal{E}_{a,D}^h} + \langle \delta_1 \llbracket 2\mu/h \rrbracket_\gamma [\alpha] \llbracket \mathbf{S}_d \mathbf{w} \rrbracket_\gamma, \llbracket \mathbf{S}_d \mathbf{u} \rrbracket_\gamma - \mathbf{u}_D \rangle_{\mathcal{E}_{a,D}^h}. \quad (21)
\end{aligned}$$

Here we have used the fact that $\llbracket \alpha \rrbracket = \alpha^+ \mathbf{n}_E^+ + \alpha^- \mathbf{n}_E^- = [\alpha] \mathbf{n}_E^+$, since $\mathbf{n}_E^- = -\mathbf{n}_E^+$. In conclusion, the Weighted Shifted Boundary Method variational form reads:

Find $\mathbf{u} \in \mathbf{V}^h(\Omega^h)$ and $p \in Q^h(\Omega^h)$ such that, $\forall \mathbf{w} \in \mathbf{V}^h(\Omega^h)$ and $\forall q \in Q^h(\Omega^h)$,

$$\begin{aligned}
0 = & (\alpha \mathbf{w}, \rho(\mathbf{u}_t - \mathbf{b}))_{\Omega^h} + (\alpha \boldsymbol{\varepsilon}(\mathbf{w}), 2\mu \boldsymbol{\varepsilon}(\mathbf{u}))_{\Omega^h} - (\alpha \nabla \cdot \mathbf{w}, p)_{\Omega^h} - (\alpha q, \nabla \cdot \mathbf{u})_{\Omega^h} \\
& - \langle \alpha \mathbf{w}, \mathbf{t}_N \rangle_{\Gamma_N} - \langle \llbracket \alpha \rrbracket \otimes \mathbf{w}, 2\mu \llbracket \boldsymbol{\varepsilon}(\mathbf{u}) \rrbracket_\gamma - p \mathbf{I} \rangle_{\mathcal{E}_{a,D}^h} \\
& - \langle 2\mu \llbracket \boldsymbol{\varepsilon}(\mathbf{w}) \rrbracket_\gamma - q \mathbf{I}, \llbracket \alpha \rrbracket \otimes (\llbracket \mathbf{S}_d \mathbf{u} \rrbracket_\gamma - \mathbf{u}_D) \rangle_{\mathcal{E}_{a,D}^h} + \langle \delta_1 \llbracket 2\mu/h \rrbracket_\gamma [\alpha] \llbracket \mathbf{S}_d \mathbf{w} \rrbracket_\gamma, \llbracket \mathbf{S}_d \mathbf{u} \rrbracket_\gamma - \mathbf{u}_D \rangle_{\mathcal{E}_{a,D}^h} \\
& + \langle \delta_u \llbracket \boldsymbol{\varepsilon}(\mathbf{w}) \rrbracket, \llbracket \boldsymbol{\varepsilon}(\mathbf{u}) \rrbracket \rangle_{\mathcal{E}_{a,D}^h} + \langle \delta_p \llbracket \nabla q \rrbracket, \llbracket \nabla p \rrbracket \rangle_{\mathcal{E}_{a,D}^h}. \quad (22)
\end{aligned}$$

The last two terms in equation (22) are the so-called *ghost-penalty* terms [29] and are introduced to preserve numerical stability and/or the matrix conditioning of the algebraic solver, whenever the volume fraction function α takes small values for elements in $\mathcal{T}_c^h$. This phenomenon is in a way analogous to the small cut cell problem [59], although we do not explicitly integrate the equations over cut elements or cutting boundaries. δ_u and δ_p are scaling parameters and are defined as

$$\delta_u = \alpha_u \mu \frac{2h^+ h^-}{h^+ + h^-}, \quad (23a)$$

$$\delta_p = \frac{\alpha_p}{\mu} \left(\frac{2h^+ h^-}{h^+ + h^-} \right)^3, \quad (23b)$$

In particular, we take $\alpha_u = 1.0$ and $\alpha_p = 0.01$. These definitions take inspiration from [40] and the refinements of these ideas presented in [90].

Remark 4. A study in [72, 81] showed that the standard SBM does not produce poor condition numbers in the case of the Poisson and linear elasticity equations, but no detailed studies have yet been performed for the WSBM. Preliminary results on the WSBM are shown in Section 5.2 and indicate that the conditions numbers of the WSBM applied to the Stokes problem are well behaved and equivalent to the condition numbers obtained with body-fitted grids of similar resolutions.

4.3. Stabilization of the Stokes flow equations in mixed form

It is well known that a Galerkin formulation of the Stokes flow equations with equal-order finite element interpolation spaces is not stable. Specifically, the Galerkin formulation outlined in Section 4.2 does not satisfy the LBB inf-sup conditions. We then introduce a stabilized form of the Stokes flow equations, using the variational multiscale approach [91–94], and we add to the variational form (22) the term

$$\text{STAB}(\mathbf{u}, p; \mathbf{w}_\alpha, q_\alpha) = (\nabla q_\alpha, \mathbf{u}')_{\mathcal{J}(\tilde{\Omega}^h)} - (\nabla \cdot \mathbf{w}_\alpha, p')_{\mathcal{J}(\tilde{\Omega}^h)} \quad (24a)$$

with $\mathcal{J}(\tilde{\Omega}^h) = \bigcup_{T \in \mathcal{T}^h \cup \mathcal{T}_h} \text{int}(T)$ the union of the interiors of the elements belonging to the surrogate domain and

$$\mathbf{u}' = -\tau_m (\rho \mathbf{u}_t + \nabla p - \nabla \cdot (2\mu \boldsymbol{\varepsilon}(\mathbf{u})) - \rho \mathbf{b}), \quad (24b)$$

$$p' = -\tau_c \nabla \cdot \mathbf{u}, \quad (24c)$$

$$\tau_m = (12\mu^2 \mathbf{G} : \mathbf{G})^{-1/2}, \quad (24d)$$

$$\tau_c = \frac{1}{8 \operatorname{tr}(\mathbf{G}) \tau_m}. \quad (24e)$$

Here $\mathbf{G}$ is the metric tensor, that is $G_{ij} = \frac{\partial \xi_\alpha}{\partial x_i} \frac{\partial \xi_\alpha}{\partial x_j}$, where $\boldsymbol{\xi}$ is the position vector in the parent finite element domain, and $\mathbf{x}$ is the position vector in the spatial configuration.

4.4. Euler-Lagrange equations

The Euler-Lagrange equations can be obtained integrating by parts. First, observe the following element-by-element integration-by-parts result, which applies to terms involving the stress $\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu \boldsymbol{\varepsilon}(\mathbf{u})$:

$$\begin{aligned} (\alpha \boldsymbol{\varepsilon}(\mathbf{w}), \boldsymbol{\sigma})_{\Omega^h} &= (\mathbf{w}_\alpha, \nabla \cdot \boldsymbol{\sigma})_{\Omega^h} + \langle \mathbf{w}, \llbracket \alpha \boldsymbol{\sigma} \rrbracket \rangle_{\tilde{\mathcal{E}}_{a,D}^h \cup \tilde{\mathcal{E}}_i^h} + \langle \mathbf{w}_\alpha, \boldsymbol{\sigma} \hat{\mathbf{n}} \rangle_{\hat{\Gamma}_D^h} \\ &= (\alpha \mathbf{w}, \nabla \cdot \boldsymbol{\sigma})_{\Omega^h} + \langle \mathbf{w} \otimes \llbracket \alpha \rrbracket, \llbracket \boldsymbol{\sigma} \rrbracket_\gamma \rangle_{\tilde{\mathcal{E}}_{a,D}^h \cup \tilde{\mathcal{E}}_i^h} + \langle \llbracket \alpha \rrbracket_{1-\gamma} \mathbf{w}, \llbracket \boldsymbol{\sigma} \rrbracket \rangle_{\tilde{\mathcal{E}}_{a,D}^h \cup \tilde{\mathcal{E}}_i^h} + \langle \mathbf{w}_\alpha, \boldsymbol{\sigma} \hat{\mathbf{n}} \rangle_{\hat{\Gamma}_D^h}. \end{aligned} \quad (25)$$

Then, integrating by parts the terms $(\alpha \boldsymbol{\varepsilon}(\mathbf{w}), 2\mu \boldsymbol{\varepsilon}(\mathbf{u}))_{\Omega^h}$ and $(\alpha \nabla \cdot \mathbf{w}, p)_{\Omega^h}$ in (22), we obtain

$$\begin{aligned} 0 = & (\alpha (\mathbf{w} - \tau_m \nabla q), \rho \mathbf{u}_{,t} + \nabla p - \nabla \cdot (2\mu \boldsymbol{\varepsilon}(\mathbf{u})) - \rho \mathbf{b})_{\Omega^h} - (\alpha (q - \tau_c \nabla \cdot \mathbf{w}), \nabla \cdot \mathbf{u})_{\Omega^h} \\ & - \langle \alpha \mathbf{w}, p\mathbf{I} - 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) \rangle_{\mathbf{n}} + \langle \mathbf{t}_N \rangle_{\Gamma_N} \\ & + \langle \llbracket \alpha \rrbracket_{1-\gamma} \mathbf{w}, \llbracket -p\mathbf{I} + 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) \rrbracket \rangle_{\tilde{\mathcal{E}}_{a,D}^h \cup \tilde{\mathcal{E}}_i^h} \\ & + \langle (q\mathbf{I} - 2\mu \llbracket \boldsymbol{\varepsilon}(\mathbf{w}) \rrbracket_\gamma) \llbracket \alpha \rrbracket + \delta_1 \llbracket \mu/h \rrbracket_\gamma [\alpha] \mathbf{w}, \llbracket \mathbf{u} + (\nabla \mathbf{u})\mathbf{d} - \mathbf{u}_D \rrbracket_\gamma \rangle_{\tilde{\mathcal{E}}_{a,D}^h} \\ & + \langle \alpha (q\hat{\mathbf{n}} - 2\mu \boldsymbol{\varepsilon}(\mathbf{w})\hat{\mathbf{n}}) + \delta_1 \mu/h \alpha \mathbf{w}, \mathbf{u} + (\nabla \mathbf{u})\mathbf{d} - \mathbf{u}_D \rangle_{\hat{\Gamma}_D^h} \\ & + \langle \delta_u \llbracket \boldsymbol{\varepsilon}(\mathbf{w}) \rrbracket, \llbracket \boldsymbol{\varepsilon}(\mathbf{u}) \rrbracket \rangle_{\tilde{\mathcal{E}}_{a,D}^h} + \langle \delta_p \llbracket \nabla q \rrbracket, \llbracket \nabla p \rrbracket \rangle_{\tilde{\mathcal{E}}_{a,D}^h}. \end{aligned} \quad (26)$$

The first row of (26) enforces the Stokes equations on the interior of the active surrogate subdomain Ω^h ; the second row enforces the Neumann (i.e., traction) boundary condition; the third row enforces the balances of stress on all interior element edges; the fourth and fifth rows enforce the shifted Dirichlet (e.g., no-slip) boundary condition; and the sixth row enforces a smoothness condition on the velocity strain and the pressure gradient (ghost penalization).

4.5. Preservation of hydrostatic equilibrium

An important property to be satisfied is the preservation of simple stationary hydrostatic equilibrium for any shape of the domain Ω^h . This solution can be expressed as $\mathbf{u} = \mathbf{0}$ and $p = -\rho g z$, where g is the magnitude of the gravitational acceleration $\mathbf{g}$ and z is the vertical coordinate. Under hydrostatic equilibrium, the pressure exactly balances the gravitational force, namely $\nabla p = \rho \mathbf{b} = \rho \mathbf{g} = -\rho g \mathbf{e}_z$. We consider here the particular case in which ρ and $\mathbf{g}$ are constant, so that the pressure gradient is constant. Algorithms that fail to satisfy the hydrostatic equilibrium condition introduce spurious pressure wavelets that propagate across the computational domain.

Considering, for the sake of simplicity, only no-slip boundary conditions, substituting $\mathbf{u} = \mathbf{0}$ in (22), and keeping in mind that the pressure gradient is constant, we obtain

$$0 = -(\alpha \mathbf{w}, \rho \mathbf{b})_{\Omega^h} - (\alpha \nabla \cdot \mathbf{w}, p)_{\Omega^h} + \langle \mathbf{w} \otimes \llbracket \alpha \rrbracket, p\mathbf{I} \rangle_{\tilde{\mathcal{E}}_{a,D}^h} + \langle \alpha \mathbf{w} \otimes \hat{\mathbf{n}}, p\mathbf{I} \rangle_{\hat{\Gamma}_D^h}. \quad (27)$$

Using integration by parts on every element $T \in \tilde{\mathcal{T}}^h \cup \mathcal{T}_c^h$, it is not difficult to see that

$$(\alpha \mathbf{w}, \nabla p)_{\Omega^h} = -(\alpha \nabla \cdot \mathbf{w}, p)_{\Omega^h} + \langle \mathbf{w} \otimes \llbracket \alpha \rrbracket, p\mathbf{I} \rangle_{\tilde{\mathcal{E}}_{a,D}^h} + \langle \alpha \mathbf{w} \otimes \hat{\mathbf{n}}, p\mathbf{I} \rangle_{\hat{\Gamma}_D^h}, \quad (28)$$

so that comparing (27) and (28), we have

$$0 = (\alpha \mathbf{w}, -\rho \mathbf{b} + \nabla p)_{\Omega^h} \quad (29)$$

and this proves that the WSBM preserves exactly a hydrostatic solution for any shape of Ω^h .

4.6. An estimate of the global mass conservation error

We provide next an estimate of the conservation error of the WSBM, and we show it converges to zero with a quadratic rate, as the grid is refined. Consider (22), in which we set $q = 1$ (i.e., $q_\alpha = \alpha$) and $\mathbf{w} = \mathbf{0}$:

$$0 = -(\alpha, \nabla \cdot \mathbf{u})_{\Omega^h} + \langle \llbracket \alpha \rrbracket, \mathbf{u} + \llbracket \nabla \mathbf{u} \rrbracket_\gamma \mathbf{d} - \mathbf{u}_D \rangle_{\mathcal{E}_{\alpha,D}^h} + \langle \alpha \hat{\mathbf{n}}, \mathbf{u} + (\nabla \mathbf{u}) \mathbf{d} - \mathbf{u}_D \rangle_{\tilde{\Gamma}_D^h}. \quad (30)$$

Splitting the integral $(\alpha, \nabla \cdot \mathbf{u})_{\Omega^h} = (1, \nabla \cdot \mathbf{u})_{\tilde{\Omega}^h} + (\alpha, \nabla \cdot \mathbf{u})_{\Omega_c^h}$, where $\alpha|_{\tilde{\Omega}^h} = 1$, and keeping in mind that α is piecewise constant and potentially discontinuous across element boundaries, we obtain

$$\begin{aligned} 0 &= (1, \nabla \cdot \mathbf{u})_{\tilde{\Omega}^h} + (\alpha, \nabla \cdot \mathbf{u})_{\Omega_c^h} - \langle \llbracket \alpha \rrbracket, \mathbf{u} + \llbracket \nabla \mathbf{u} \rrbracket_\gamma \mathbf{d} - \mathbf{u}_D \rangle_{\mathcal{E}_{\alpha,D}^h} - \langle \alpha \hat{\mathbf{n}}, \mathbf{u} + (\nabla \mathbf{u}) \mathbf{d} - \mathbf{u}_D \rangle_{\tilde{\Gamma}_D^h} \\ &= \langle 1, \mathbf{u} \cdot \mathbf{n}^+ \rangle_{\tilde{\Gamma}_D^h} + \langle 1, \mathbf{u} \cdot \mathbf{n} \rangle_{\Gamma_N^h} + (\alpha, \nabla \cdot \mathbf{u})_{\Omega_c^h} \\ &\quad - \underbrace{\langle \alpha^+ \rangle_{\tilde{\Gamma}_D^h}}_{=1} \langle \mathbf{u} + \llbracket \nabla \mathbf{u} \rrbracket_\gamma \mathbf{d} - \mathbf{u}_D \rangle_{\tilde{\Gamma}_D^h} - \langle \alpha^- \mathbf{n}^-, \mathbf{u} + \llbracket \nabla \mathbf{u} \rrbracket_\gamma \mathbf{d} - \mathbf{u}_D \rangle_{\tilde{\Gamma}_D^h} \\ &\quad - \langle \llbracket \alpha \rrbracket, \mathbf{u} + \llbracket \nabla \mathbf{u} \rrbracket_\gamma \mathbf{d} - \mathbf{u}_D \rangle_{\mathcal{E}_{c,D}^h} - \langle \alpha \hat{\mathbf{n}}, \mathbf{u} + (\nabla \mathbf{u}) \mathbf{d} - \mathbf{u}_D \rangle_{\tilde{\Gamma}_D^h} \\ &= \langle 1, (\mathbf{u}_D - \llbracket \nabla \mathbf{u} \rrbracket_\gamma \mathbf{d}) \cdot \mathbf{n}^+ \rangle_{\tilde{\Gamma}_D^h} + \langle 1, \mathbf{u} \cdot \mathbf{n} \rangle_{\Gamma_N^h} + e_{cons}(\mathbf{u}; \mathcal{T}_c^h), \end{aligned} \quad (31a)$$

where $\tilde{\mathbf{n}}^+$ indicates the normal to $\tilde{\Gamma}_D$ pointing toward $\tilde{\Omega}_D$, $\tilde{\mathbf{n}}^- = -\tilde{\mathbf{n}}^+$, and

$$\begin{aligned} e_{cons}(\mathbf{u}; \mathcal{T}_c^h) &= (\alpha, \nabla \cdot \mathbf{u})_{\Omega_c^h} - \langle \alpha^- \tilde{\mathbf{n}}^-, \llbracket \mathbf{u} + (\nabla \mathbf{u}) \mathbf{d} - \mathbf{u}_D \rrbracket_\gamma \rangle_{\tilde{\Gamma}_D^h} - \langle \llbracket \alpha \rrbracket, \llbracket \mathbf{u} + (\nabla \mathbf{u}) \mathbf{d} - \mathbf{u}_D \rrbracket_\gamma \rangle_{\mathcal{E}_{c,D}^h} \\ &\quad - \langle \alpha \hat{\mathbf{n}}, \mathbf{u} + (\nabla \mathbf{u}) \mathbf{d} - \mathbf{u}_D \rangle_{\tilde{\Gamma}_D^h}. \end{aligned} \quad (31b)$$

Note that (31b) represents the error in the statement of discrete global mass conservation on the domain $\tilde{\Omega}$, expressed in (31a), and sheds some light on the nature of the WSBM.

In fact, as we refine the computational grid the domain Ω_c^h shrinks, and in particular we can estimate $\text{meas}(\Omega_c^h) = O(h)$, since the depth of Ω_c^h scales with h , while the area of the transversal surface is substantially unaffected. Furthermore, for a stable and convergent numerical method, we can expect $\nabla \cdot \mathbf{u} \rightarrow 0$ with some power of h . If we take as an example the standard SBM for the Stokes problem [77, 78], we could expect $\nabla \cdot \mathbf{u} = O(h)$. Hence, by Cauchy-Schwartz inequality, we can roughly estimate

$$(\alpha, \nabla \cdot \mathbf{u})_{\Omega_c^h} \leq \underbrace{\|\alpha\|_{0;\Omega_c^h}}_{O(h^{1/2})} \underbrace{\|\nabla \cdot \mathbf{u}\|_{0;\Omega_c^h}}_{O(h^{3/2})} = O(h^2). \quad (32)$$

Similarly, for a stable and convergent method, the solution will converge to the boundary conditions imposed on Dirichlet (e.g., no-slip) boundaries. For linear finite element spaces, one can expect that

$$\begin{aligned} &\langle \alpha^- \tilde{\mathbf{n}}^-, \llbracket \mathbf{u} + (\nabla \mathbf{u}) \mathbf{d} - \mathbf{u}_D \rrbracket_\gamma \rangle_{\tilde{\Gamma}_D^h} + \langle \llbracket \alpha \rrbracket, \llbracket \mathbf{u} + (\nabla \mathbf{u}) \mathbf{d} - \mathbf{u}_D \rrbracket_\gamma \rangle_{\mathcal{E}_{c,D}^h} + \langle \alpha \hat{\mathbf{n}}, \mathbf{u} + (\nabla \mathbf{u}) \mathbf{d} - \mathbf{u}_D \rangle_{\tilde{\Gamma}_D^h} \\ &\quad + \langle \alpha^- (\tilde{\mathbf{n}}^- \cdot \mathbf{n}), (\mathbf{u} + (\nabla \mathbf{u}) \mathbf{d}) \cdot \mathbf{n} - u_S \rangle_{\tilde{\Gamma}_{S'}^h} + \langle \llbracket \alpha \rrbracket \cdot \mathbf{n}, (\mathbf{u} + \llbracket \nabla \mathbf{u} \rrbracket_\gamma \mathbf{d}) \cdot \mathbf{n} - u_S \rangle_{\mathcal{E}_{c,S'}^h} \\ &\quad + \langle \alpha (\hat{\mathbf{n}} \cdot \mathbf{n}), (\mathbf{u} + (\nabla \mathbf{u}) \mathbf{d}) \cdot \mathbf{n} - u_S \rangle_{\tilde{\Gamma}_{S'}^h} = O(h^2). \end{aligned} \quad (33)$$

Summarizing, the WSBM produces a mass conservation error that, for piecewise linear interpolation, is quadratic in the mesh size h , namely,

$$\langle 1, (\mathbf{u}_D - \llbracket \nabla \mathbf{u} \rrbracket_\gamma \mathbf{d}) \cdot \tilde{\mathbf{n}}^+ \rangle_{\tilde{\Gamma}_D^h} + \langle 1, \mathbf{u} \cdot \mathbf{n} \rangle_{\Gamma_N^h} = -e_{cons}(\mathbf{u}; \mathcal{T}_c^h) = O(h^2). \quad (34)$$

Remark 5. The proposed approach delivers a statement of conservation of active volume that is similar to the one obtained in cutFEMs. In fact, in the case of cutFEM, the boundary is approximated with an affine (piecewise-linear) representation. Hence, the geometric error introduced by cutFEMs in the representation of the volume of active fluid is of order $O(h^2)$.

For higher-order interpolations, the same argument would lead to higher-order estimates of the conservation error. In particular, for a method of accuracy h^{p+1} in the velocity field, we can expect $e_{cons}(\mathbf{u}; \mathcal{T}_c^h) = O(h^{p+1})$.

4.7. An estimate of the global momentum conservation error

Next let us consider an estimate of the momentum conservation error of the WSBM, and like in [4.6](#), we show that the associated numerical error is quadratic. Set $q = 0$ and $\mathbf{w} = \mathbf{e}_i$ in [\(22\)](#) and we can get:

$$\begin{aligned} 0 &= (\alpha \mathbf{e}_i, \partial_t(\rho \mathbf{u}))_{\Omega^h} - (\alpha \mathbf{e}_i, \rho \mathbf{b})_{\Omega^h} - \langle \alpha \mathbf{e}_i, \mathbf{t}_N \rangle_{\Gamma_N} \\ &\quad + \langle \mathbf{e}_i, (p\mathbf{I} - 2\mu \boldsymbol{\varepsilon}(\mathbf{u}))_\gamma \llbracket \alpha \rrbracket + \delta_1[\alpha] \llbracket 2\mu/h \rrbracket_\gamma (\mathbf{S}_d \mathbf{u} - \mathbf{u}_D) \rangle_{\mathcal{E}_{a,D}^h} \end{aligned} \quad (35)$$

Observe that $\llbracket \alpha \boldsymbol{\varepsilon}(\mathbf{u}) \rrbracket = \llbracket \boldsymbol{\varepsilon}(\mathbf{u}) \rrbracket_\gamma \llbracket \alpha \rrbracket + \llbracket \alpha \rrbracket_{1-\gamma} \llbracket \boldsymbol{\varepsilon}(\mathbf{u}) \rrbracket$ and that, when we constructed the bilinear form, we required $\langle \llbracket \alpha \mathbf{w} \rrbracket_{1-\gamma}, \llbracket 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) - p\mathbf{I} \rrbracket \rangle_{\tilde{\mathcal{E}}_i^h \cup \mathcal{E}_{c,D}^h \cup \tilde{\Gamma}_D^h} = 0$, which yields $\langle \llbracket \alpha \rrbracket_{1-\gamma} \mathbf{e}_i, \llbracket 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) \rrbracket \rangle_{\tilde{\mathcal{E}}_i^h \cup \mathcal{E}_{c,D}^h \cup \tilde{\Gamma}_D^h} = 0$ for $\mathbf{w} = \mathbf{e}_i$, with a continuous pressure. Then we obtain:

$$\begin{aligned} 0 &= (\alpha \mathbf{e}_i, \partial_t(\rho \mathbf{u}))_{\tilde{\Omega}^h \cup \Omega_c^h} - (\alpha \mathbf{e}_i, \rho \mathbf{b})_{\tilde{\Omega}^h \cup \Omega_c^h} - \langle \alpha \mathbf{e}_i, \mathbf{t}_N \rangle_{\Gamma_N} \\ &\quad + \langle \mathbf{e}_i, \llbracket \alpha(p\mathbf{I} - 2\mu \boldsymbol{\varepsilon}(\mathbf{u})) \rrbracket + \delta_1[\alpha] \llbracket 2\mu/h \rrbracket_\gamma \llbracket \mathbf{S}_d \mathbf{u} - \mathbf{u}_D \rrbracket_\gamma \rangle_{\mathcal{E}_{a,D}^h} \\ &= (\alpha \mathbf{e}_i, \partial_t(\rho \mathbf{u}))_{\tilde{\Omega}^h \cup \Omega_c^h} - (\alpha \mathbf{e}_i, \rho \mathbf{b})_{\tilde{\Omega}^h \cup \Omega_c^h} - \langle \alpha \mathbf{e}_i, \mathbf{t}_N \rangle_{\Gamma_N} \\ &\quad + \langle \mathbf{e}_i, (p\mathbf{I} - 2\mu \boldsymbol{\varepsilon}(\mathbf{u}^+)) \underbrace{\alpha^+}_{=1} \tilde{\mathbf{n}}^+ + (p\mathbf{I} - 2\mu \boldsymbol{\varepsilon}(\mathbf{u}^-)) \alpha^- \tilde{\mathbf{n}}^- + \delta_1[\alpha] \llbracket 2\mu/h \rrbracket_\gamma \llbracket \mathbf{S}_d \mathbf{u} - \mathbf{u}_D \rrbracket_\gamma \rangle_{\tilde{\Gamma}_D^h} \\ &\quad + \langle \mathbf{e}_i, \llbracket \alpha(p\mathbf{I} - 2\mu \boldsymbol{\varepsilon}(\mathbf{u})) \rrbracket + \delta_1[\alpha] \llbracket 2\mu/h \rrbracket_\gamma \llbracket \mathbf{S}_d \mathbf{u} - \mathbf{u}_D \rrbracket_\gamma \rangle_{\mathcal{E}_{c,D}^h} \\ &\quad + \langle \alpha \mathbf{e}_i, (p\mathbf{I} - 2\mu \boldsymbol{\varepsilon}(\mathbf{u})) + 2\delta_1 \mu/h (\mathbf{S}_d \mathbf{u} - \mathbf{u}_D) \rangle_{\tilde{\Gamma}_D^h}. \end{aligned} \quad (36)$$

In the previous derivations, we also used the fact that $\mathbf{u}_D$ is uniquely defined over edges. Using the Gauss divergence theorem on every element in $\mathcal{T}_c^h$, it is not difficult to prove that

$$\begin{aligned} (\alpha \mathbf{e}_i, \nabla \cdot (p\mathbf{I} - 2\mu \boldsymbol{\varepsilon}(\mathbf{u})))_{\Omega_c^h} &= \langle \mathbf{e}_i, (p\mathbf{I} - 2\mu \boldsymbol{\varepsilon}(\mathbf{u}^-)) \alpha^- \tilde{\mathbf{n}}^- \rangle_{\tilde{\Gamma}_D^h} + \langle \mathbf{e}_i, \llbracket \alpha(p\mathbf{I} - 2\mu \boldsymbol{\varepsilon}(\mathbf{u})) \rrbracket \rangle_{\mathcal{E}_{c,D}^h} \\ &\quad + \langle \alpha \mathbf{e}_i, (p\mathbf{I} - 2\mu \boldsymbol{\varepsilon}(\mathbf{u})) \rangle_{\tilde{\Gamma}_D^h}, \end{aligned} \quad (37)$$

so that, recalling that $\alpha = 1$ on $\tilde{\Omega}^h$ and over Γ_N and $\tilde{\Gamma}_D^{h,+}$, [\(36\)](#) yields

$$\begin{aligned} 0 &= (\mathbf{e}_i, \partial_t(\rho \mathbf{u}))_{\tilde{\Omega}^h} - (\mathbf{e}_i, \rho \mathbf{b})_{\tilde{\Omega}^h} - \langle \mathbf{e}_i, \mathbf{t}_N \rangle_{\Gamma_N} + \langle \mathbf{e}_i, (p\mathbf{I} - 2\mu \boldsymbol{\varepsilon}(\mathbf{u}^+)) \tilde{\mathbf{n}}^+ + \delta_1[\alpha] \llbracket 2\mu/h \rrbracket_\gamma \llbracket \mathbf{S}_d \mathbf{u} - \mathbf{u}_D \rrbracket_\gamma \rangle_{\tilde{\Gamma}_D^h} \\ &\quad + e_{mom}(\mathbf{u}; \mathcal{T}_c^h). \end{aligned} \quad (38a)$$

where the momentum conservation error is given by

$$\begin{aligned} e_{mom}(\mathbf{u}; \mathcal{T}_c^h) &= (\alpha \mathbf{e}_i, \partial_t(\rho \mathbf{u}) - \rho \mathbf{b} + \nabla \cdot (p\mathbf{I} - 2\mu \boldsymbol{\varepsilon}(\mathbf{u})))_{\Omega_c^h} \\ &\quad + \langle \mathbf{e}_i, \delta_1[\alpha] \llbracket 2\mu/h \rrbracket_\gamma \llbracket \mathbf{S}_d \mathbf{u} - \mathbf{u}_D \rrbracket_\gamma \rangle_{\mathcal{E}_{c,D}^h} + \langle \alpha \mathbf{e}_i, 2\delta_1 \mu/h (\mathbf{S}_d \mathbf{u} - \mathbf{u}_D) \rangle_{\tilde{\Gamma}_D^h}. \end{aligned} \quad (38b)$$

For a converging discretization, we can expect the residual to converge to zero as the grid is refined and, in our specific case,

$$\partial_t(\rho \mathbf{u}) - \rho \mathbf{b} + \nabla \cdot (p\mathbf{I} - 2\mu \boldsymbol{\varepsilon}(\mathbf{u})) = O(h),$$

which implies, similar to the case of the mass conservation error, that

$$(\alpha \mathbf{e}_i, \partial_t(\rho \mathbf{u}) - \rho \mathbf{b} + \nabla \cdot (p\mathbf{I} - 2\mu \boldsymbol{\varepsilon}(\mathbf{u})))_{\Omega_c^h} = O(h^2).$$

The terms $\llbracket 2\mu/h \rrbracket_\gamma \llbracket \mathbf{S}_d \mathbf{u} - \mathbf{u}_D \rrbracket_\gamma$ scale in principle as h . In summary, we can expect:

$$e_{mom}(\mathbf{u}; \mathcal{T}_c^h) = O(h). \quad (39)$$

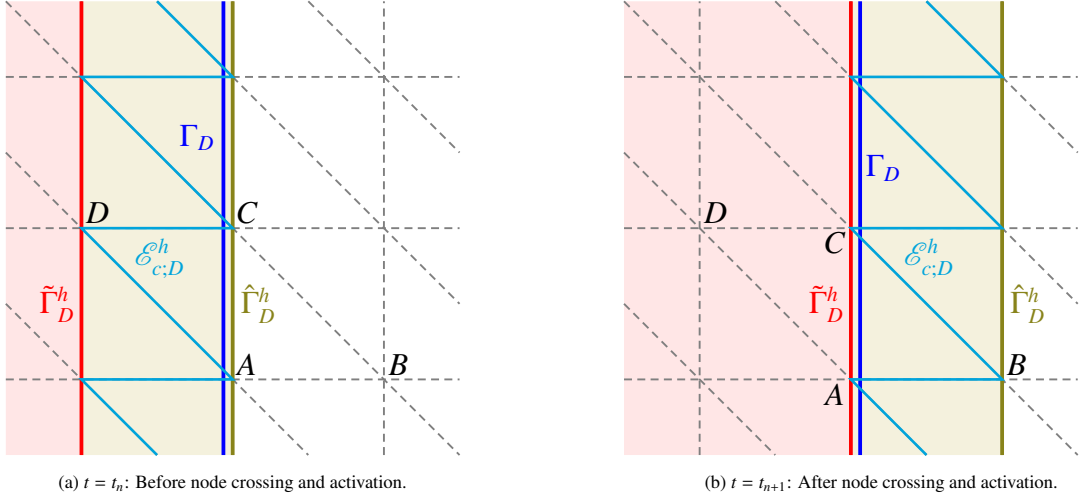


Figure 3: Sketch of the activation of new elements/nodes, and corresponding modification of the surrogate boundaries $\tilde{\Gamma}_D^h$, $\mathcal{E}_{c;D}^h$, and $\hat{\Gamma}_D^h$ (similarly for $\tilde{\Gamma}_{S'}^h$, $\mathcal{E}_{c;S'}^h$, and $\hat{\Gamma}_{S'}^h$).

4.8. Smooth evolution in time of the variational form with newly activated nodes and elements

The variational formulation associated with the proposed Weighted Shifted Boundary Method has the property of varying smoothly in time, when new elements/nodes are activated by the sweeping motion of the boundary Γ across the computational grid. Here by “smooth” we mean that the variational terms defined on the element interiors and the boundaries $\tilde{\Gamma}_D^h$, $\mathcal{E}_{c;D}^h$, and $\hat{\Gamma}_D^h$ (or, resp., $\tilde{\Gamma}_{S'}^h$, $\mathcal{E}_{c;S'}^h$, and $\hat{\Gamma}_{S'}^h$) produce forces at the nodes that change smoothly over time as the time step decreases to smaller and smaller values. We argue that this property is beneficial in reducing pressure oscillations in time, or other numerical instabilities as pointed out in [90].

To see that indeed the forces do not significantly change in value as new nodes are activated in the computational active domain, consider, for the sake of simplicity and without loss of generality, the variational equations (22) in the case of pure Dirichlet boundary conditions and without ghost penalty terms, namely

$$\begin{aligned}
0 = & (\alpha \mathbf{w}, \rho(\mathbf{u}_t - \mathbf{b}))_{\Omega^h} + (\alpha \boldsymbol{\varepsilon}(\mathbf{w}), 2\mu \boldsymbol{\varepsilon}(\mathbf{u}))_{\Omega^h} - (\alpha \nabla \cdot \mathbf{w}, p)_{\Omega^h} - (\alpha q, \nabla \cdot \mathbf{u})_{\Omega^h} \\
& - \langle [\![\alpha]\!] \otimes \mathbf{w}, 2\mu [\![\boldsymbol{\varepsilon}(\mathbf{u})]\!]_{\gamma} - p \mathbf{I} \rangle_{\mathcal{E}_{c;D}^h \cup \tilde{\Gamma}_D^h} - \langle 2\mu [\![\boldsymbol{\varepsilon}(\mathbf{w})]\!]_{\gamma} - q \mathbf{I}, [\![\alpha]\!] \otimes ([\![S_d \mathbf{u}]\!]_{\gamma} - \mathbf{u}_D) \rangle_{\mathcal{E}_{c;D}^h \cup \tilde{\Gamma}_D^h} \\
& + \langle \delta_1 [2\mu/h]_{\gamma} [\alpha] [\![S_d \mathbf{w}]\!]_{\gamma}, [\![S_d \mathbf{u}]\!]_{\gamma} - \mathbf{u}_D \rangle_{\mathcal{E}_{c;D}^h \cup \tilde{\Gamma}_D^h} \\
& - \langle \alpha \hat{\mathbf{n}} \otimes \mathbf{w}, 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) - p \mathbf{I} \rangle_{\hat{\Gamma}_D^h} - \langle 2\mu \boldsymbol{\varepsilon}(\mathbf{w}) - q \mathbf{I}, \alpha \hat{\mathbf{n}} \otimes (S_d \mathbf{u} - \mathbf{u}_D) \rangle_{\hat{\Gamma}_D^h} \\
& + \langle \delta_1 2\mu/h \alpha S_d \mathbf{w}, S_d \mathbf{u} - \mathbf{u}_D \rangle_{\hat{\Gamma}_D^h}.
\end{aligned} \tag{40}$$

For the motion of Γ_D described in Figure 3, we indicate by $t = t_c$ the time at which a new vertical row of nodes is crossed, and we set $t_n = t_c - \epsilon$ and $t_{n+1} = t_c + \epsilon$, where $\epsilon > 0$ is a very small number.

We are interested in assessing whether or not the forces produced at the nodes by the variational terms in (40) vary abruptly between t_n and t_{n+1} . We refer to the nodes and elements associated with the vertices A , B , C , and D , as an example of typical nodes and elements in the discretization. Note that, for the geometry at hand, $\tilde{\Gamma}_D^h(t_{n+1}) \equiv \hat{\Gamma}_D^h(t_n)$. Certainly, we are considering here a specific geometric configuration of the grid and a specific motion of the boundary Γ_D , but much of the analysis presented next can be generalized to arbitrary grids and an arbitrary motion of Γ_D .

In our discussion, we will assume that the numerical discretization is stable, which means that the numerical solution is smooth and gradients do not abruptly vary between t_n and t_{n+1} . Our goal is in fact to establish that no abrupt changes in the nodal forces occur over time if the solution is smooth.

Consider first the nodal forces associated with node D , and note that the interface integrals on $\tilde{\Gamma}_D^h(t_n)$, $\mathcal{E}_{c;D}^h(t_n)$ involve the jumps $[\![\alpha]\!]$ and $[\alpha]$, in which $\alpha^{\pm}$ are either unity or very close to unity. Hence these integrals will produce very small contributions to the forces at D . In fact, most of the interface integral contribution comes from the integral

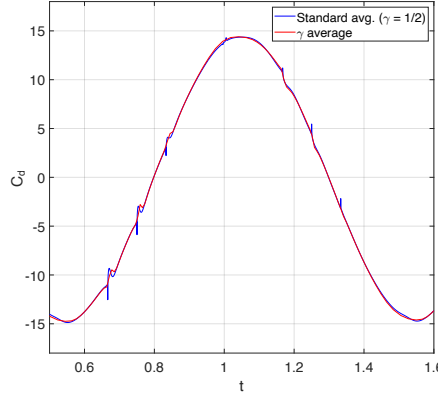


Figure 4: Oscillating square in a fluid at rest. The setup of this problem is described in detail in Section 5.4. Time-histories of the pressure non-dimensional coefficient C_d defined by (41), for $CFL = 0.1$ (the CFL condition is defined in (44)) and a side length/mesh-size ratio $L/h = 16$. In blue, the result obtained with the standard average ($\gamma^+ = \gamma^- = 1/2$). In red, the result using the γ -average (19), which shows less spurious pressure oscillations.

terms over $\hat{\Gamma}_D^h$. Also, the interior integrals over the elements connected to D will involve values of α equal to or very close to unity at t_n . At $t = t_{n+1}$, the value of α in these elements will be exactly unity. Hence the forces produced by these elements will not substantially change between t_n and t_{n+1} .

If we now consider the contribution of the integrals over $\tilde{\Gamma}_D^h$ at t_{n+1} , we can observe that $\alpha^+ = 1$ and $\alpha^- \approx 0$, since the boundary Γ_D produces small cuts. For this reason, $[\![\alpha]\!] \approx \mathbf{n}^+$, $[\alpha] \approx 1$, $\gamma^+ \approx 1$, $\gamma^- \approx 0$, and $\{\mathbf{w}\}_\gamma \approx \mathbf{w}^+$. Consequently, we have that the integrals over $\tilde{\Gamma}_D^h(t_{n+1})$ produce values very close to the integrals over $\hat{\Gamma}_D^h(t_n)$.

The previous arguments show that we can expect a smooth evolution of the forces associated with node D , and analogous conclusions can be reached for nodes A and C , observing that, at t_{n+1} , the integrals over $\mathcal{E}_{c,D}^h(t_{n+1})$ and $\hat{\Gamma}_D^h(t_{n+1})$ involve very small values of the volume fraction α .

Similarly, node B is inactive at time t_n and the forces at time t_{n+1} build up gently, since they are scaled with very small values of α , including the term over $\tilde{\Gamma}_D(t_{n+1})$, involving $\{\mathbf{S}_d \mathbf{w}\}_\gamma$ and $\{\mathbf{e}(\mathbf{w})\}_\gamma$, which produce contributions to the nodal forces in B that scale with $\gamma^- \approx 0$. In conclusion, the nodal forces grow smoothly as long as the numerical solution stays smooth during the time evolution.

We now show how small changes in the definition of edge averages can perturb the situation described above. Figure 4 illustrates the comparison between outcomes derived from the a standard edge average $\gamma^+ = \gamma^- = 1/2$, and the γ -average described in (19), for a numerical problem described in full detail in Section 5.4, involving an oscillating square in a fluid initially at rest. A quantitative measure of the forces produced by the pressure field is given by the *pressure drag coefficient* along the direction of motion of the moving boundary (indicated by the unit vector $\mathbf{e}_{mb}$):

$$C_d = \frac{\int_\Gamma p(\mathbf{n} \cdot \mathbf{e}_{mb}) d\Gamma}{(1/2)\rho L^3 f_0^2} \approx \frac{\int_{\tilde{\Gamma}} (p + \nabla p \cdot \mathbf{d})(\mathbf{n} \cdot \mathbf{e}_{mb})(\tilde{\mathbf{n}} \cdot \mathbf{n}) d\tilde{\Gamma}}{(1/2)\rho L^3 f_0^2}, \quad (41)$$

where f_0 is the frequency of oscillation, L is a characteristic length of the problem under consideration (the side L of the square in this case), Γ is the boundary of the moving shape, $\tilde{\Gamma}$ is its surrogate counterpart, and $\tilde{\mathbf{n}} \cdot \mathbf{n} = |\mathrm{d}\Gamma|/|\mathrm{d}\tilde{\Gamma}|$ is the corresponding *signed* area ratio. The sign of $\tilde{\mathbf{n}} \cdot \mathbf{n}$ avoids to double count area contributions when zig-zagging boundaries are present [82, 83]. Observe that, depending on the specific test, C_d may have the nominal physical interpretation of a drag coefficient or a lift coefficient, but this does not alter the conclusions that follow. It is evident that utilizing the γ -average technique results in fewer spurious pressure oscillations and produces more reliable results.

If we now want to take into account the effect of the ghost penalty terms, we can argue that if the numerical solutions are smooth, the ghost penalty terms will be in general rather small, and they will not substantially change the overall discussion. Figure 5 shows the effect of a change in the ghost penalty parameter α_u in (23a) on the pressure drag coefficient C_d defined in (41). Two grid resolutions are considered as well as values of the parameter α_u that

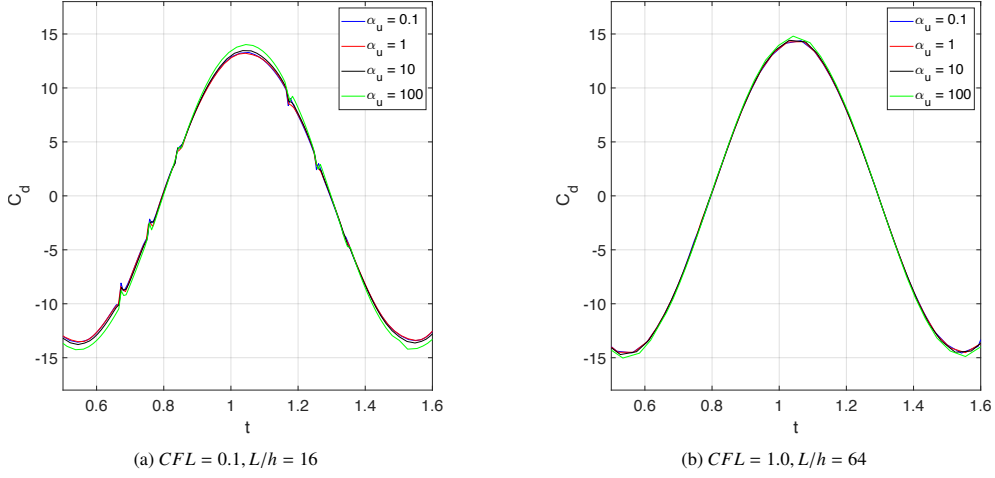


Figure 5: Effect of a change of the ghost penalty parameter α_u on the pressure drag coefficient C_d . Left: $CFL = 0.1$ and a coarse grid resolution ($L/h = 16$). Right: $CFL = 1.0$ and fine grid resolution ($L/h = 64$). The CFL condition is defined in (44) and $\alpha_p = 0.01$.

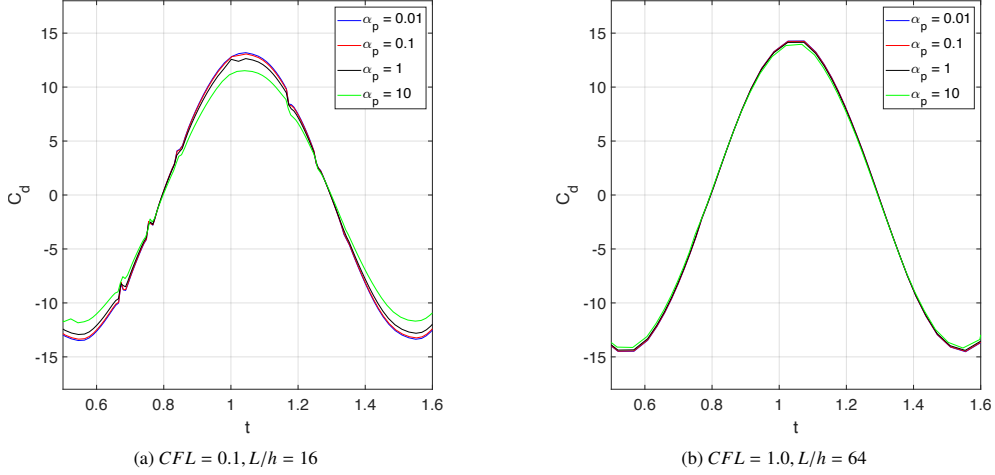


Figure 6: Effect of a change of the ghost penalty parameter α_p on the pressure drag coefficient C_d . Left: $CFL = 0.1$ and a coarse grid resolution ($L/h = 16$). Right: $CFL = 1.0$ and fine grid resolution ($L/h = 64$). The CFL condition is defined in (44) and $\alpha_u = 1.0$.

span four orders of magnitude. Figure 5 shows that the effect of the ghost penalty is minimal on the solution, whether a coarse or a fine grid are used. Of course, in the case of a coarse grid, the time history of C_d is rougher, for reasons that will be explained in Section 4.9, but the parameter α_u does not significantly perturb the numerical results. Similar conclusions can be drawn if the parameter α_p in (23b) is varied by orders of magnitude, as shown in Figure 6.

4.9. Velocity initialization for newly activated nodes and its effect on pressure oscillations in time

The governing equations are integrated using a BDF2 time integrator, namely

$$\mathbf{u}_{,t} \approx \frac{3\mathbf{u}^{n+1} - 4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\Delta t}. \quad (42)$$

One important issue to be discussed is how to initialize the velocity field $\mathbf{u}$ in regions of the grid that become active at t_{n+1} but are inactive at previous time instants. The velocity initialization has in fact some important effects on spurious pressure oscillations in time that occur for immersed methods.

Most existing references [9, 17, 18, 20] are focused on properly accounting that the total volume of active fluid is

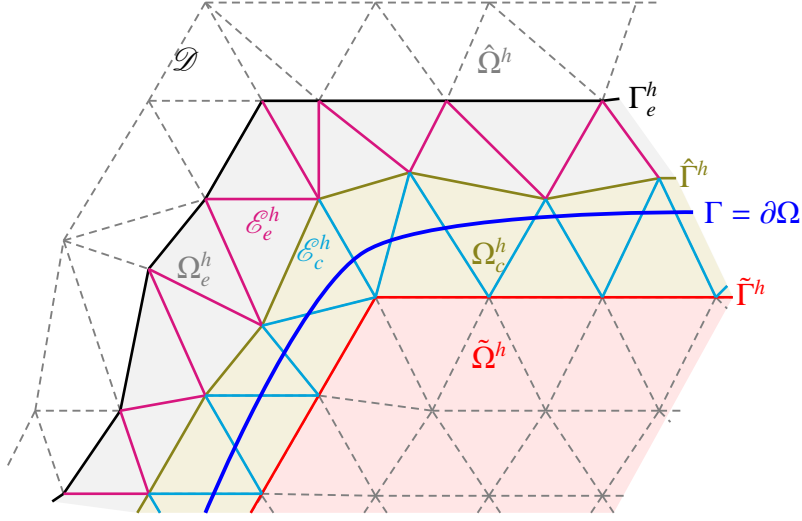


Figure 7: The background domain $\mathcal{D} = \tilde{\Omega}^h \cup \Omega_c^h \cup \Omega_e^h$, active surrogate subdomain $\Omega^h = \tilde{\Omega}^h \cup \Omega_c^h$, extension domain $\Omega_e^h \subset \tilde{\Omega}^h$, and the surrogate interfaces $\tilde{\Gamma}^h, \hat{\Gamma}^h, \mathcal{E}_c^h$.

conserved to a high degree of accuracy. Here instead, we want to point out a different issue that has a profound effect on spurious pressure oscillation in time. Consider Figure 7 and in particular the nodes on the boundary Γ_e^h at a certain time step. To be specific Γ_e^h is the boundary that contains the exterior nodes that are connected by an edge or face to nodes in $\hat{\Gamma}^h$, in two or three dimensions, respectively. As the true boundary Γ moves, it is possible that nodes on Γ_e^h will be activated, and become part of the boundary $\hat{\Gamma}^h$ over the next time step. Because the velocity time derivative needs to be computed to advance in time the velocity itself, one needs to initialize the velocity on the boundary Γ_e^h . For example, in the context of this work, we use the velocity of the body moving inside the fluid to initialize the velocity on nodes on Γ_e^h .

Now consider the limit case in which the time step goes to zero. Only a few terms will dominate the variational form. As $\Delta t \rightarrow 0$, the time derivative term could become increasingly large, since it scales like the inverse of Δt . Other terms in (22) can be as large, in the limit of a small time step: the mass conservation terms, since the conservation equation is independent from the momentum equation, and the pressure terms, since the pressure represents the Lagrange multiplier that enforces the incompressibility constraint. The viscous, body force, and velocity boundary condition terms, instead, do not scale like the inverse of Δt and become negligible as the time step is reduced. In summary, as $\Delta t \rightarrow 0$, (22) reduces to:

$$\begin{aligned} 0 \approx & \frac{1}{2\Delta t} (\alpha \mathbf{w}, \rho (3\mathbf{u}^{n+1} - 4\mathbf{u}^n + \mathbf{u}^{n-1}))_{\Omega^h} - (\alpha \nabla \cdot \mathbf{w}, p)_{\Omega^h} + \langle \llbracket \alpha \rrbracket \cdot \mathbf{w}, p \rangle_{\mathcal{E}_{a,D}^h} \\ & - (\alpha q, \nabla \cdot \mathbf{u})_{\Omega^h} + \langle \delta_p \llbracket \nabla q \rrbracket, \llbracket \nabla p \rrbracket \rangle_{\tilde{\mathcal{E}}_{a,D}^h}. \end{aligned} \quad (43)$$

Now we can return to the discussion about the initialization of the solutions $\mathbf{u}^n$ (and $\mathbf{u}^{n-1}$) for nodes on Γ_e^h , which was initiated at the beginning of this section. In fact, the extensions of $\mathbf{u}^n$ and $\mathbf{u}^{n-1}$ to nodes on Γ_e^h may not precisely respect the incompressibility condition over the elements shaded in gray in Figure 7 that is the elements lying between $\hat{\Gamma}^h$ and Γ_e^h . As a result, the pressure field will need to activate to re-establish the incompressibility constraint for the solution at time t_{n+1} and, in particular, the pressure average over the active domain will adjust.

This is in a nutshell the mechanism that leads to spurious pressure oscillations in time, even when the total volume of fluid is correctly accounted for, that is approximated within the spatial accuracy of the underlying numerical method. Note that this issue does not depend on the spatial discretization, in the sense that it should appear in unfitted FEMs, cutFEMs, as well as the current formulation. Observe that the ghost penalty term $\langle \delta_p \llbracket \nabla q \rrbracket, \llbracket \nabla p \rrbracket \rangle_{\tilde{\mathcal{E}}_{a,D}^h}$ is typically smooth in time, as already argued from the results in Figures 5 and 6, and does not substantially alter the previous discussion.

We could not find this discussion in the existing literature, in the sense that the work in [9] and related contributions tend to emphasize the importance of accounting for the correct conserved volume, but there is no mention to the effect of solution initialization on the incompressibility constraint. The key aspect then becomes to quantify how large the oscillations are as the time step is reduced, and we will present extensive numerical experiments on this subject.

5. Numerical results

We present next a number of numerical tests for the performance assessment of the proposed approach. These tests are primarily focused on the demonstration of the WSBM's accuracy and the WSBM's properties discussed in Sections 4.5, 4.6, 4.7, and 4.8.

We will also test the WSBM in the limit of very small time steps, to check the occurrence of the pressure oscillations described in Section 4.9. We would like to stress that we are not aware of any immersed/embedded/unfitted method (of finite difference/volume/element type) that can completely avoid pressure oscillations in time, in the limit of small time steps. Methods and approaches are classified as superior or inferior to one another based on how small a time step can be taken without significant oscillations. We also would like to point out that, traditionally, pressure oscillations in time are studied in the context of the (nonlinear) Navier-Stokes equations rather than the (linear) Stokes flow equations. In the case of the Navier-Stokes equations, the pressure oscillations are exacerbated by the presence of the nonlinear convection terms, so it is not possible to make a fair comparison between some of the results presented in our simulations of Stokes' flow and references on the Navier-Stokes equations. However, the proposed analysis demonstrates that the pressure oscillations in time *are not a nonlinear phenomenon*, as they occur also in the case of Stokes's flow. This fact can spur further mathematical analysis that will be simpler in the case of the linear Stokes' system of equations, as opposed to the Navier-Stokes equations.

For simplicity and unless otherwise stated, the velocity of the fluid is initialized with the velocity of the immersed body, for nodes that lie on the extension boundary Γ_e^h , as suggested in Section 4.9. Some of the computations will be run at constant CFL , for the non-classical CFL condition

$$\Delta t \leq CFL \min_{T \in \mathcal{T}_c^h} \frac{D_T}{\|\mathbf{u}_\Gamma\|}, \quad (44)$$

where $\mathbf{u}_\Gamma$ is the boundary velocity and D_T is the diameter of the circle inscribed in an element $T \in \mathcal{T}_c^h$ (i.e., an element intersected by Γ). This advective-type CFL condition involves the magnitude of the velocity of the boundary Γ and the size of the cut elements. In the current implementation, we allow the boundary to sweep at most one layer of newly activated elements at each time step, and for this reason it is not possible to run calculations with $CFL > 1$. Of course, if more layers of nodes beyond $\hat{\Gamma}$ are initialized, it would be possible to run with $CFL > 1$.

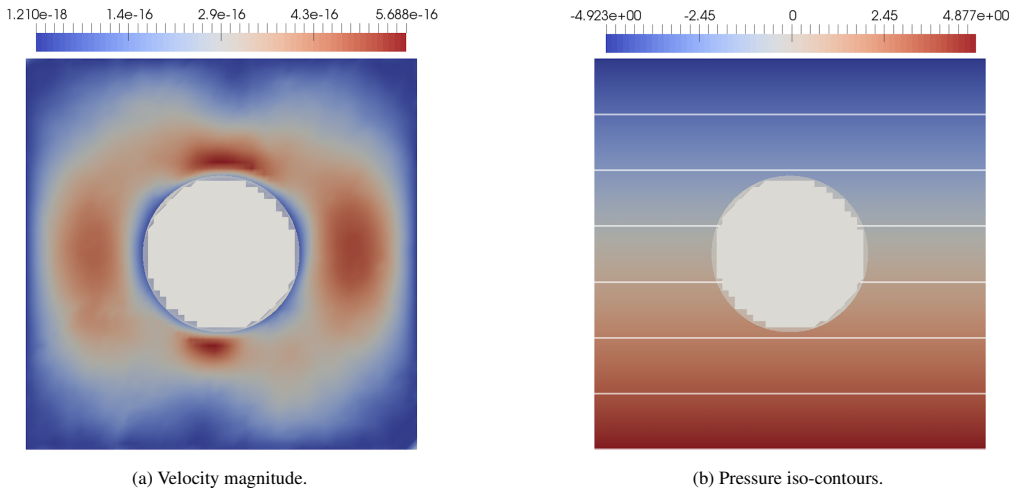


Figure 8: Hydrostatic equilibrium patch test. The pressure isolines are perfectly horizontal and the velocity is zero up to machine precision.

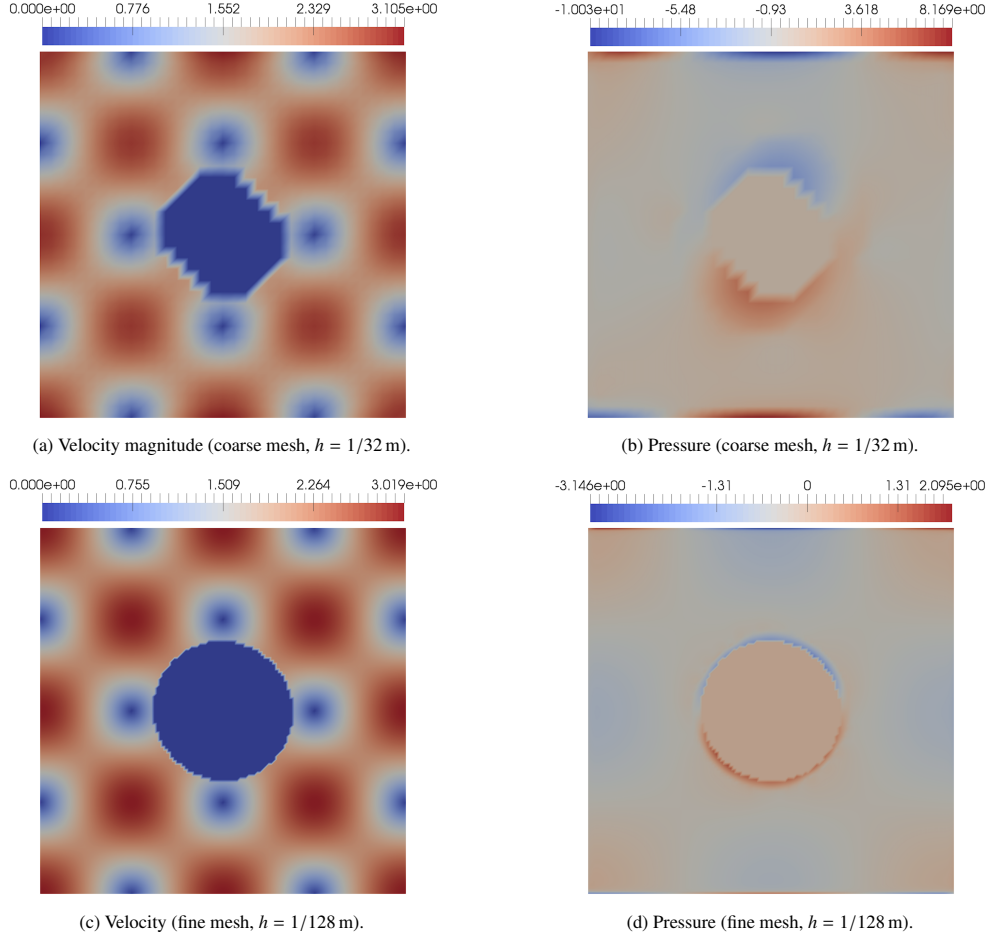


Figure 9: Convergence test. Velocity magnitude and pressure on a coarse and a fine mesh.

5.1. A patch test: fluid in hydrostatic equilibrium

This first test involves a hydrostatic equilibrium problem, which the WSBM can solve within machine precision. We consider water ($\rho = 1000 \text{ kg/m}^3$) in static equilibrium inside the square tank $[-0.5, 0.5] \times [-0.5, 0.5] \text{ m}^2$ with a circular hole of radius 0.2 m in the middle. The body force is due to gravity, namely $\mathbf{b} = -9.8\mathbf{e}_y \text{ m/s}^2$, where $\mathbf{e}_y$ is the unit vector aligned with the vertical axis.

No-slip (Dirichlet) boundary conditions $\mathbf{u} = \mathbf{0}$ are applied to the four sides of the square, where the mesh is body-fitted. The circular hole is instead embedded, and there the no-slip (Dirichlet) boundary condition $\mathbf{u} = \mathbf{0}$ is enforced with the WSBM.

The results are shown in Figure 8: the pressure and velocity match the analytical values within machine precision, and confirm the analysis developed in Section 4.5. In particular, the pressure contours are perfectly horizontal and the velocity magnitude, which coincides with the magnitude of the velocity error, is on the order of 10^{-15} m/s .

5.2. Convergence test

Next, we examine the accuracy of the WSBM using the method of manufactured solutions. The geometry is the same as in the previous test, but the center $\mathbf{x}_c$ of the circular hole boundary is now moving according to the law

$$\mathbf{x}_c(t) = \begin{Bmatrix} -X_0 \cos(2\pi f_0 t) \\ 0 \end{Bmatrix} \text{ m}, \quad (45a)$$

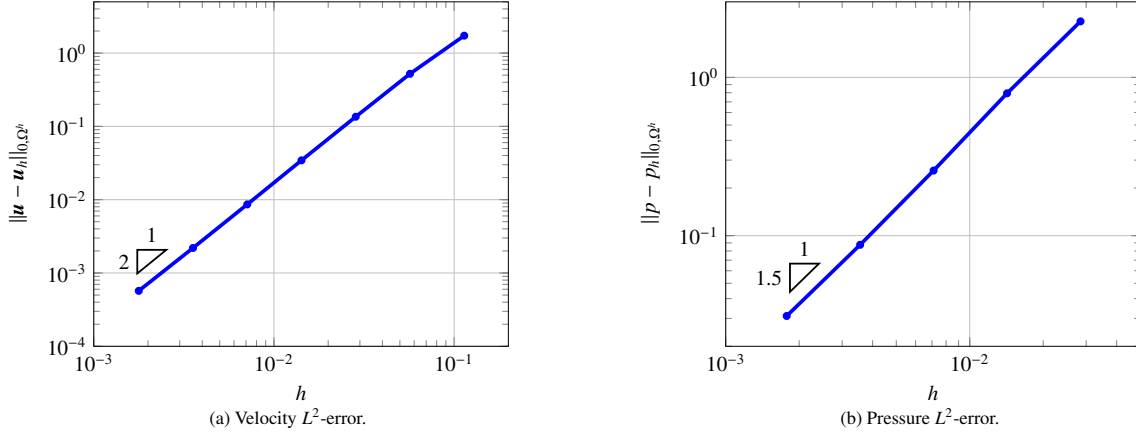


Figure 10: Convergence rates of the L^2 -norm of the errors $\|\mathbf{u} - \mathbf{u}_h\|_{0,\Omega^h}$ and $\|p - p_h\|_{0,\Omega^h}$ for a manufactured solution with an oscillating circular boundary. Numerical solutions are computed at time $t = 1.6T$.

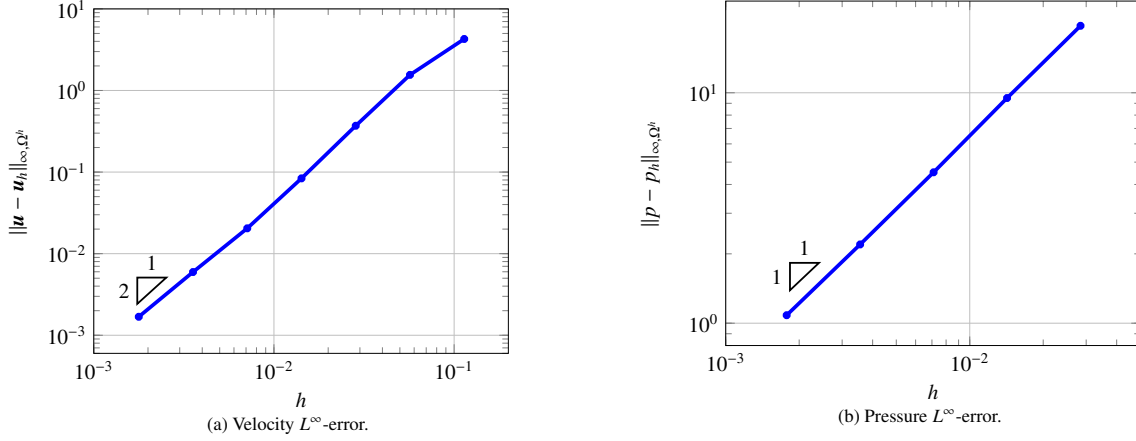


Figure 11: Convergence rates of the L^∞ -norm of the errors $\|\mathbf{u} - \mathbf{u}_h\|_{\infty,\Omega^h} = \sup_{\Omega^h} \|\mathbf{u} - \mathbf{u}_h\|$ and $\|p - p_h\|_{\infty,\Omega^h} = \sup_{\Omega^h} |p - p_h|$ for a manufactured solution with an oscillating circular boundary. Numerical solutions are computed at time $t = 1.6T$.

$$\mathbf{u}_c(t) = \begin{Bmatrix} U_0 \sin(2\pi f_0 t) \\ 0 \end{Bmatrix} \text{ m/s} , \quad (45b)$$

where $\mathbf{u}_c$ is the velocity of the center of the circular hole, $X_0 = 0.03125$ m, $U_0 = 2\pi f_0 X_0$ m/s, and $T = 1/f_0 = 1.0$ s is the period of the oscillation.

The density is $\rho = 1$ kg/m³ and we consider the following exact solution

$$\mathbf{u}(x, y) = \begin{Bmatrix} 2\pi \sin(2\pi y) \sin(2\pi x) \sin(t) \\ 2\pi \cos(2\pi x) \cos(2\pi y) \sin(t) \end{Bmatrix} \text{ m/s} , \quad (46a)$$

$$p(x, y) = (\cos(2\pi x) \cos(2\pi y) - 1) \sin(t) \text{ N/m}^2 , \quad (46b)$$

obtained by forcing the Stokes' flow equations with

$$\mathbf{b}(x, y) = \begin{Bmatrix} -(2\pi \cos(2\pi y) \sin(2\pi x) - 16\pi^3 \sin(2\pi x) \sin(2\pi y)) \sin(t) + 2\pi \sin(2\pi x) \sin(2\pi y) \cos(t) \\ -(2\pi \cos(2\pi x) \sin(2\pi y) - 16\pi^3 \cos(2\pi x) \cos(2\pi y)) \sin(t) + 2\pi \cos(2\pi x) \cos(2\pi y) \cos(t) \end{Bmatrix} \text{ m/s}^2 . \quad (47a)$$

Observe that the velocity field is divergence-free and in fact this flow can be considered as a forced version of the

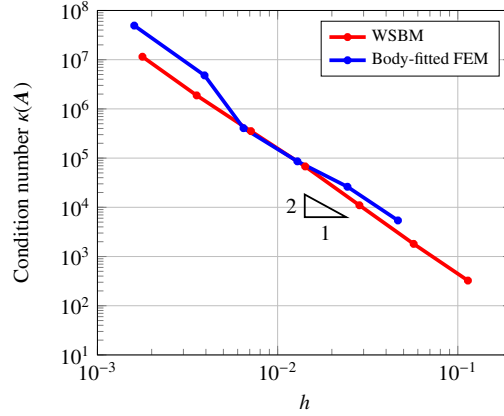


Figure 12: Comparison of matrix condition numbers for the WSBM and a body-fitted FEM. Computations were performed with two series of grids of similar resolution (h).

Taylor-Green vortex. As already mentioned, the velocity is initialized in the set of nodes in Γ_e^h using the velocity (45b) of the circular rigid immersed domain. We apply no-slip (Dirichlet) boundary conditions

$$\mathbf{u}_D = \begin{cases} 2\pi \sin(2\pi y) \sin(2\pi x) \sin(t) \\ 2\pi \cos(2\pi x) \cos(2\pi y) \sin(t) \end{cases} \text{ m/s} , \quad (47b)$$

at the top/bottom/hole boundaries and the traction (Neumann) conditions

$$\mathbf{t}_N = \begin{cases} \begin{cases} -(8\pi^2 \cos(2\pi x) \sin(2\pi y) - \cos(2\pi x) \cos(2\pi y) + 1) \sin(t) \\ 0 \end{cases} \text{ N/m}^2 \text{ at the left boundary} , \\ \begin{cases} (8\pi^2 \cos(2\pi x) \sin(2\pi y) - \cos(2\pi x) \cos(2\pi y) + 1) \sin(t) \\ 0 \end{cases} \text{ N/m}^2 \text{ at the right boundary} . \end{cases} \quad (47c)$$

This test has been solved using progressively refined meshes, obtained by first generating square meshes of sizes $h = 1/8$ m, $1/16$ m, $1/32$ m, $1/64$ m, $1/128$ m, $1/256$ m, and $1/512$ m and then splitting every square in two sub-triangles.

Computations with these mesh resolutions were run at constant time step, with corresponding values $\Delta t = 0.28971$, 0.14486 , 0.072428 , 0.036214 , 0.018107 , 0.0090535 , 0.0045268 , so that time and space errors stay balanced. Figure 9 shows the velocity and pressure profiles obtained from a coarse mesh ($h = 1/16$ m) and a relatively finer mesh ($h = 1/128$ m) at time $t = 1.6T$. Both meshes yield satisfactory results, although the solution on the coarse mesh exhibits higher error values. The convergence rates of the velocity and pressure L^2 -errors at time $t = 1.6T$ are shown in Figure 10, and match the expected rates for stabilized methods on body-fitted meshes. In particular, the convergence rate of the velocity error is quadratic and the convergence rate of the pressure error is 1.5. The L^∞ -errors at time $t = 1.6T$ are plotted in Figure 11 and show quadratic rate for the velocity and linear rate for the pressure. This means that the numerical solution smoothly converges to the exact solution, without localized spurious oscillations in space. The slight degradation in convergence rate for the L^∞ -norm of the pressure error is expected, since convergence in the L^∞ -norm is in general more challenging than in any L^p -norm.

Remark 6. The initialization techniques for the nodes that become active, described in Section 4.9, rely on assigning to nodes in the set Γ_e^h the value of the velocity in the rigid moving body. This in principle could produce a deterioration in the order of accuracy of the velocity, which we do not observe in computations. A possible explanation is that the convergence tests were run with time steps calibrated to yield a maximum CFL number over the period of oscillation approximately equal to 0.455. It is possible that in this conditions the time derivative term in the Stokes equation is not particularly large compared to the other terms and therefore does not contribute significantly to the overall error.

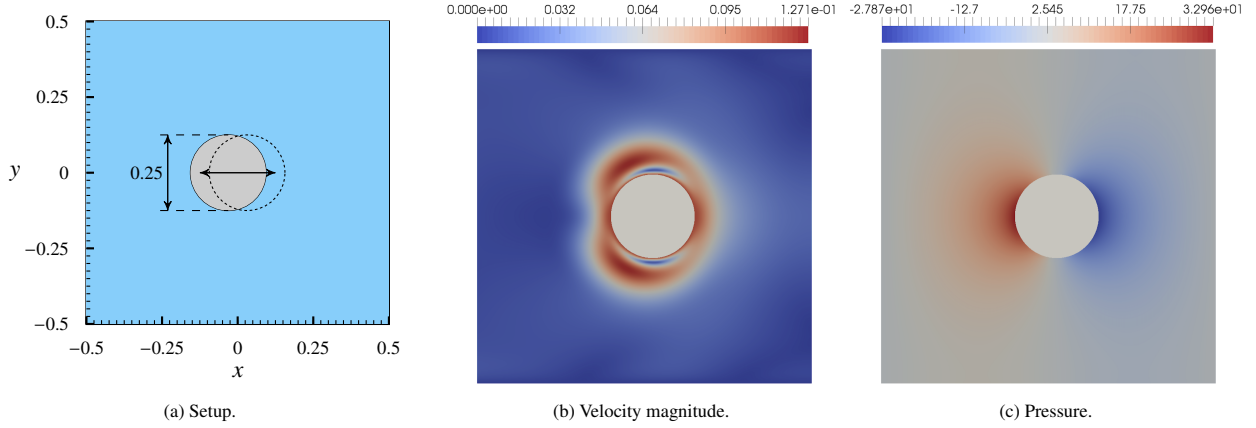


Figure 13: Oscillating cylinder in a fluid initially at rest. Left: setup. Center/right: velocity magnitude/pressure at time $t = 1.5T$, for a mesh with $D/h = 64$.

Figure 12 presents a comparison of the condition numbers $\kappa(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$ of the algebraic matrix $\mathbf{A}$ obtained discretizing, on similar grids, the WSBM and a body-fitted implementation of a stabilized finite element formulation of the Stokes equations (i.e., the body-fitted Navier-Stokes formulation discussed in [70] without convective terms). The calculations are performed with fixed boundaries, since the motion of boundaries is not relevant to the computation of condition numbers. It is clear from Figure 12 that the WSBM delivers conditions numbers that are practically identical to the body-fitted discretization, and scale as h^{-2} when grids are refined, as expected.

5.3. Oscillating cylinder in flow at rest

This benchmark was proposed by Seo and Mittal [9] in the context of finite volume simulations of the Navier-Stokes equations with direct-forcing immersed boundary approaches. It consists of an oscillating (circular) cylinder in a square domain that has been filled with a fluid initially at rest. A sketch of the setup is shown in Figure 13. The diameter of the cylinder is $D = 0.25$ m. The initial conditions are $\mathbf{u}(0) = \mathbf{u}_0 = 0$ m/s and $p(0) = p_0 = 0$ N/m². No-slip boundary conditions are enforced on the upper and lower boundaries, while a pressure boundary condition $p = p_0$ is enforced on the left and right boundaries. The center of the cylinder is initially situated at the position $(-X_0, 0)$ and moves according to the following equations:

$$\mathbf{x}_c(t) = \begin{Bmatrix} -X_0 \cos(2\pi f_0 t) \\ 0 \end{Bmatrix} \text{ m}, \quad (48a)$$

$$\mathbf{u}_c(t) = \begin{Bmatrix} U_0 \sin(2\pi f_0 t) \\ 0 \end{Bmatrix} \text{ m/s}, \quad (48b)$$

where $X_0 = 0.125D = 0.03125$ m, $U_0 = 2\pi f_0 X_0$ m/s and the period of oscillation is $T = 1/f_0 = 1.0$ s. The density of the fluid is chosen to be $\rho = 160.0$ kg/m³ and the dynamic viscosity of the fluid is $\mu = 1.0$ kg/(m s). Also for this test, the computational meshes are obtained by subdividing the domain in squares of size h and then splitting each square into two sub-triangles.

We can introduce the “Reynolds number of the moving boundary,” which clearly does not carry the same meaning as in the case of the Navier-Stokes equations: $Re = \rho U_0 D / \mu = 7.85$. Our idea is to scale the problem in a very similar fashion to the original test proposed in [9], although the absence of convection terms in the Stokes flow does make a difference. Notwithstanding the obvious differences between Stokes and Navier-Stokes flows, the numerical results will show that the pressure oscillations are primarily due to the numerical handling of the incompressibility constraint and the initialization of the solution beyond the physical (moving) boundary, as discussed in Sections 4.6, 4.8 and 4.9. Of course, the presence of convection terms in the Navier-Stokes equations may exacerbate the situation, but the root cause of the pressure oscillation problem can be explained already in the context of the simpler (and linear) Stokes flow.

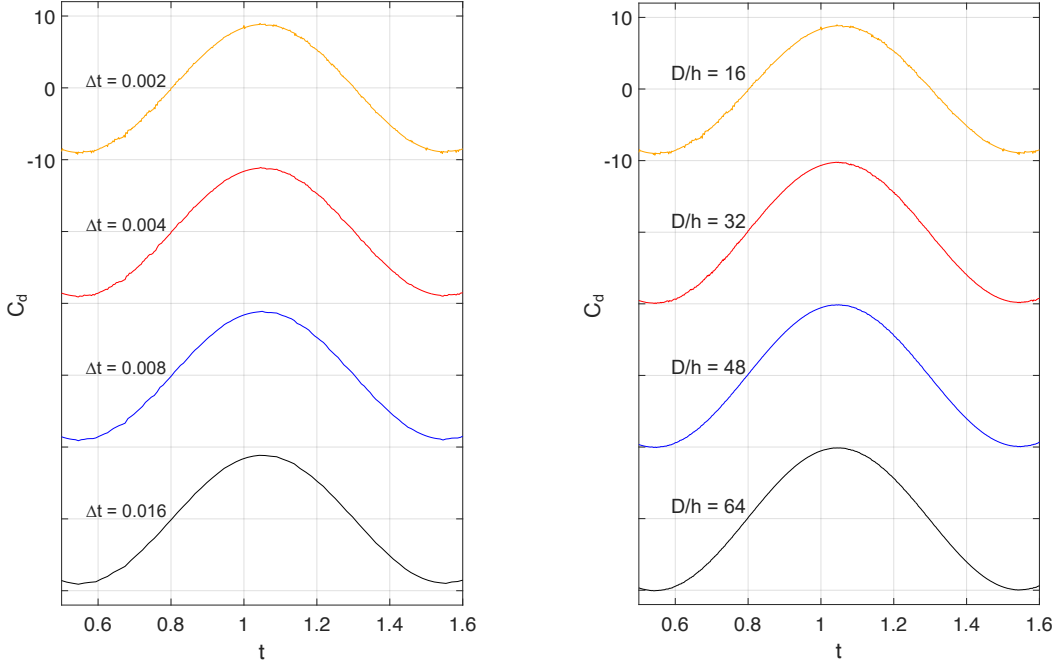


Figure 14: Oscillating cylinder in a fluid at rest: C_d time histories. Left: fixed grid size ($D/h = 16$) and varying time step Δt . Right: Fixed time step ($\Delta t = 0.002$) and varying grid sizes (D/h ratio). Note that the yellow curves on the left and right plots correspond to the same simulation.

Figure 13 shows the velocity and pressure profiles at time $t = 1.5T$, which are smooth and consistent with the theoretical expectations. Seo and Mittal [9] performed an analysis of the onset of pressure oscillations in time as the time steps is reduced in computations. Note that the pressure oscillates only in time and remains smooth in space, as already seen in Figure 13. Figure 14 shows the behavior of the WSBM for two of the same numerical tests performed in [9], that is a study of the oscillations of the pressure drag coefficient C_d , defined in (41) with $L = D$, as the grid size or the time step size are reduced, respectively. The results displayed in Figure 14 show that the SBM performs well in keeping the pressure oscillations under control even for very small time steps or very large grid sizes.

To quantitatively capture the behavior of high-frequency pressure oscillations, we first filter out the smooth component of the drag coefficient history, $\bar{C}_d$, and quantify the oscillation level only on the high-frequency components of the C_d , using the oscillation metric

$$\Theta(C_d; t_n) = C_d(t_n) - \bar{C}_d(t_n). \quad (49)$$

We achieve this by applying to the signal C_d a Butterworth high-pass filter, whose flat frequency response minimizes distortion of the waveform. The time series data for C_d is initially imported into a pandas DataFrame. To construct the filter, the `scipy.signal.butter` function of the Scientific Python library was used to compute the necessary filter coefficients. The sampling frequency is calculated from the minimum time step in the calculations, as $f_{\text{sampling}} = 1/\Delta t_{\text{min}}$, and we set the filter order to be 2, a common choice. The cutoff frequency is chosen equal to f_0 , the frequency of oscillation of the cylinder. The normalized cutoff frequency is then calculated by dividing the cutoff frequency by the Nyquist frequency $f_{\text{Nyquist}} = f_{\text{sampling}}/2$, which represents the maximum frequency that can be accurately represented, according to the Nyquist-Shannon sampling theorem. This choice of parameters will also be made in the numerical investigations described in Sections 5.4, 5.5, and 5.6. The filtering of the data is executed by the `scipy.signal.filtfilt` function, which enforces a zero-phase response by applying the filter coefficients in both forward and reverse directions.

A visual example is provided in Figure 15 displaying the original $C_d(t_n)$ (in red) and filtered signal $\Theta(C_d; t_n)$ (in blue) across the chosen time interval, offering an illustrative contrast and validation of the filtering process. A global

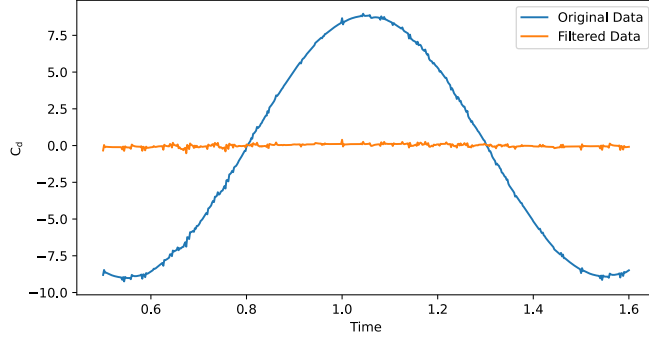


Figure 15: Comparison of original data and filtered data. Oscillating cylinder in flow at rest, grid size ($D/h = 16$) with $\Delta t = 0.002$. Specifically, C_d is shown in blue, and $\Theta(C_d; t_n) = C_d(t_n) - \bar{C}_d(t_n)$ is shown in orange.

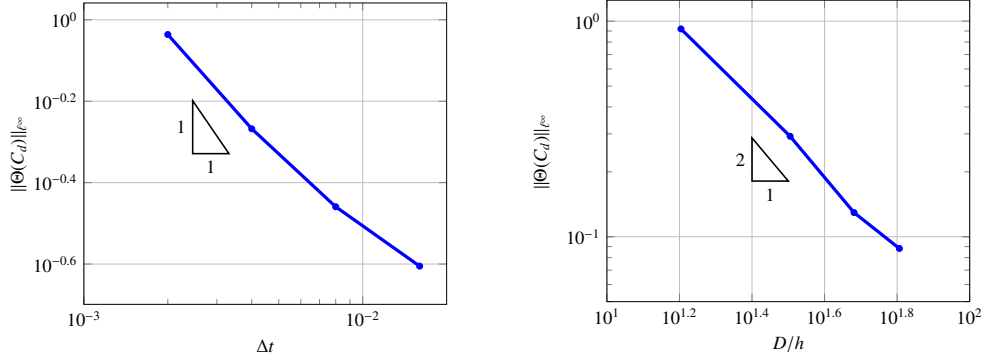


Figure 16: Oscillating cylinder in a fluid at rest: rates of decay of $\|\Theta(C_d)\|_{\ell^\infty}$. Left: fixed grid size ($D/h = 16$) and varying time step Δt . Right: Fixed time step ($\Delta t = 0.002$) and varying grid sizes (D/h ratio).

measure of the oscillations is the ℓ^∞ -norm of $\Theta(C_d; t_n)$:

$$\|\Theta(C_d)\|_{\ell^\infty} = \max_{t_n \in \mathbb{T}} |\Theta(C_d; t_n)|. \quad (50)$$

This value is calculated in a selected temporal window $\mathbb{T}$, which includes a whole period of the oscillation motion. Specifically, we choose a starting time of 0.5, and an ending time of 1.6, that is $\mathbb{T} = [0.5, 1.6]$.

Figure 16 shows the rate of decay of $\|\Theta(C_d)\|_{\ell^\infty}$ as the time step and the grid size are varied, for the same simulations depicted in Figure 14. These results are consistent with the finding by Seo and Mittal [9] that quantities analogous to $\|\Theta(C_d)\|_{\ell^\infty}$ scale like h^2 and Δt^{-1} . The numerical values of $\|\Theta(C_d)\|_{\ell^\infty}$ are reported in Table 1.

Table 1: Oscillating cylinder in a fluid at rest: decay of $\|\Theta(C_d)\|_{\ell^\infty}$. Left: fixed grid size ($D/h = 16$) and varying Δt . Right: Fixed time step ($\Delta t = 0.002$) and varying grid sizes.

Δt	$\ \Theta(C_d)\ _{\ell^\infty}$	D/h	$\ \Theta(C_d)\ _{\ell^\infty}$
0.002	0.920	16	0.920
0.004	0.540	32	0.292
0.008	0.347	48	0.129
0.016	0.248	64	0.0882

Often in the practice, computations are run at constant CFL number, because of physical parametrizations of multi-physics problems, etc. We report in Figure 17 the time histories of C_d for various CFL numbers and two grid

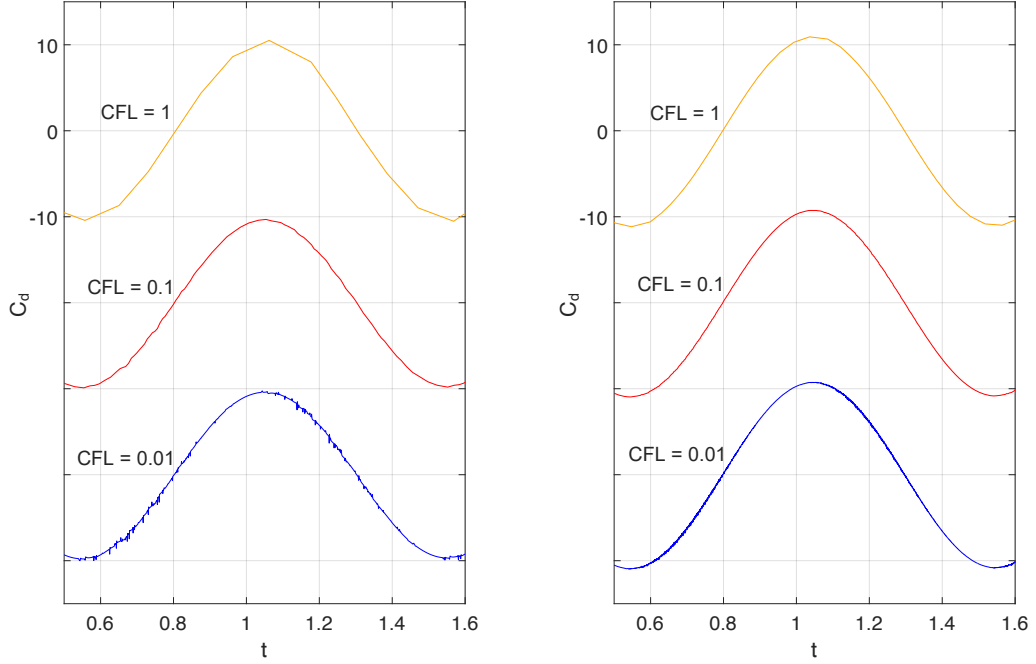
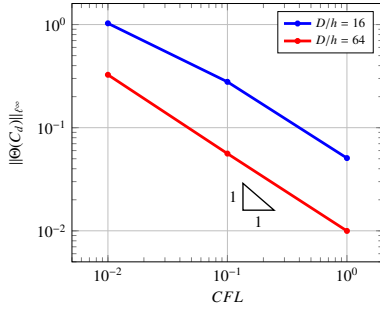


Figure 17: Oscillating cylinder in a fluid at rest. C_d time histories for various CFL numbers and two grid sizes: $D/h = 16$ (left) and $D/h = 64$ (right).

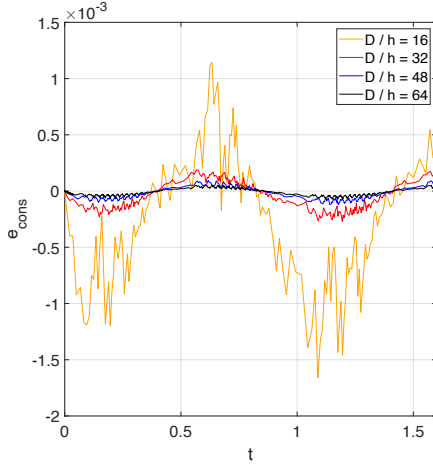


CFL	$D/h = 16$	$D/h = 64$
1.0	0.0507	0.000997
0.1	0.278	0.0561
0.01	1.03	0.326

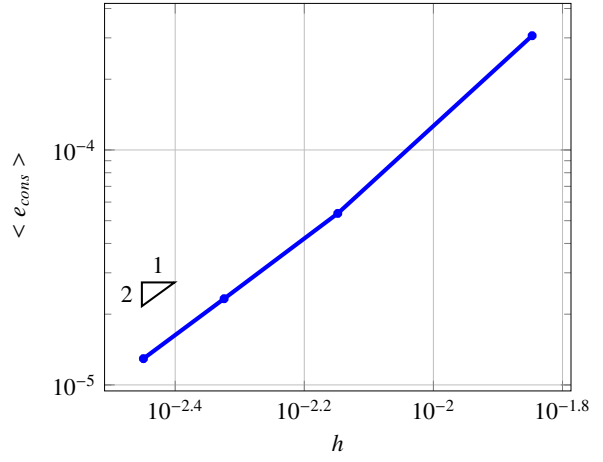
Figure 18: Oscillating cylinder in a fluid at rest. Variation of $\|\Theta(C_d)\|_{\ell^\infty}$ as a function of CFL , for two grid sizes: $D/h = 16$ (blue) and $D/h = 64$ (red). Numerical values are shown on the table on the right.

sizes, $D/h = 16$ and $D/h = 64$. These results show the excellent performance of the WSBM in avoiding pressure oscillations in time. The results in Figure 17 show that oscillations appear as the CFL approaches the value 0.01, but even in this case they remain on the order of 2-3% of the peak value of C_d , and could be acceptable in some engineering applications. Instead, when considering values of the CFL in the range $[0.1, 1]$, oscillations are minimal if not completely absent. Note also that oscillations are reduced as the grid is refined (they are virtually absent for $CFL = 0.01$ and $D/h = 64$). Figure 18 shows the decay of pressure oscillations as the CFL is increased, for the two grid sizes considered in Figure 17. In this case $\|\Theta(C_d)\|_{\ell^\infty}$ scales like CFL^{-1} , which is compatible with the scalings of $\|\Theta(C_d)\|_{\ell^\infty}$ described in Figure 16 and Table 1.

We further investigate the performance of the proposed WSBM by quantifying the mass conservation error e_{cons} , defined in (31b), in the plots of Figure 19. Figure 19a presents the time history of the mass conservation error for different mesh sizes. Specifically, the mass conservation error oscillates around zero, with smaller oscillations observed as the mesh becomes finer. Interestingly, the phase of oscillation remains consistent across all mesh sizes. The mean value of the mass conservation error over time is plotted in Figure 19b against the mesh size. Notably, a second-order convergence rate is observed, which confirms the analysis outlined in Section 4.6.

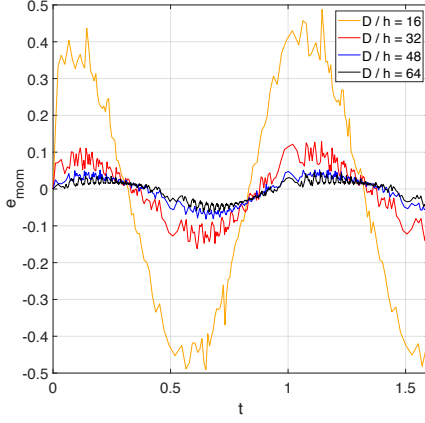


(a) Time history of mass conservation error for different mesh sizes.

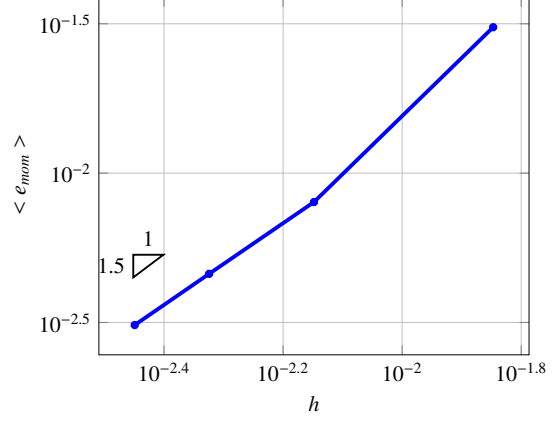


(b) Convergence of time average of the mass conservation error.

Figure 19: Mass conservation error e_{cons} for different mesh sizes and convergence of its time average.



(a) Time history of momentum conservation error for different mesh sizes.



(b) Convergence of time average of the momentum conservation error.

Figure 20: Momentum conservation error e_{mom} for different mesh sizes and convergence of its time average as the mesh is refined.

Similar plots are presented in Figure 20 for the momentum conservation error, defined in (38b) and discussed in Section 4.7.

In this case we observe super-convergence with respect to the estimate provided in Section 4.7 that is, the empirically observed order of convergence of e_{mom} is 1.5 instead of the theoretically predicted value of 1.

5.4. Oscillating square in flow at rest

We consider now a variant of the previous test in which the circular cylinder is replaced by a square with a side of length $D = 0.25$. The domain geometry, physical parameters, initial conditions, boundary conditions, prescribed boundary motion, and meshes remain identical to the previous test, besides the geometry of the oscillating shape.

The square is positioned in such a way that two of its sides are perfectly aligned with the edges of the various meshes tested. This configuration is considerably more challenging for the pressure oscillation problem, in the following sense. In the case of the circular cylinder, it is typical for only a single node at a time to be activated or deactivated. In fact, we could precisely observe that the time oscillations correspond to a single node being activated, as already found in [9]. In the case of the square shape, an entire row of elements can be activated over a particular time step, and controlling potential pressure oscillations are expected to be more challenging.

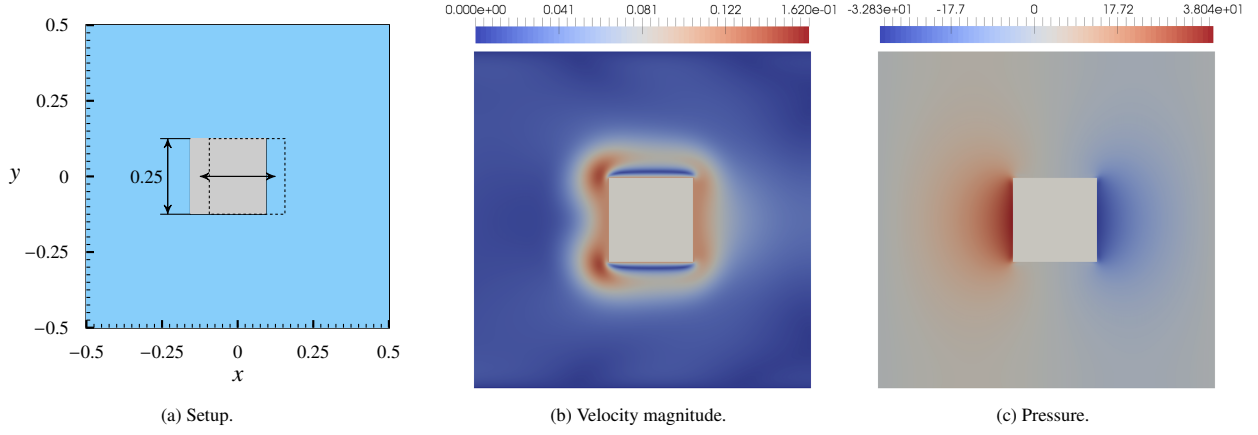


Figure 21: Oscillating square in a fluid initially at rest. Left: setup. Center/right: velocity magnitude/pressure at time $t = 1.5T$, for a mesh with $D/h = 64$.

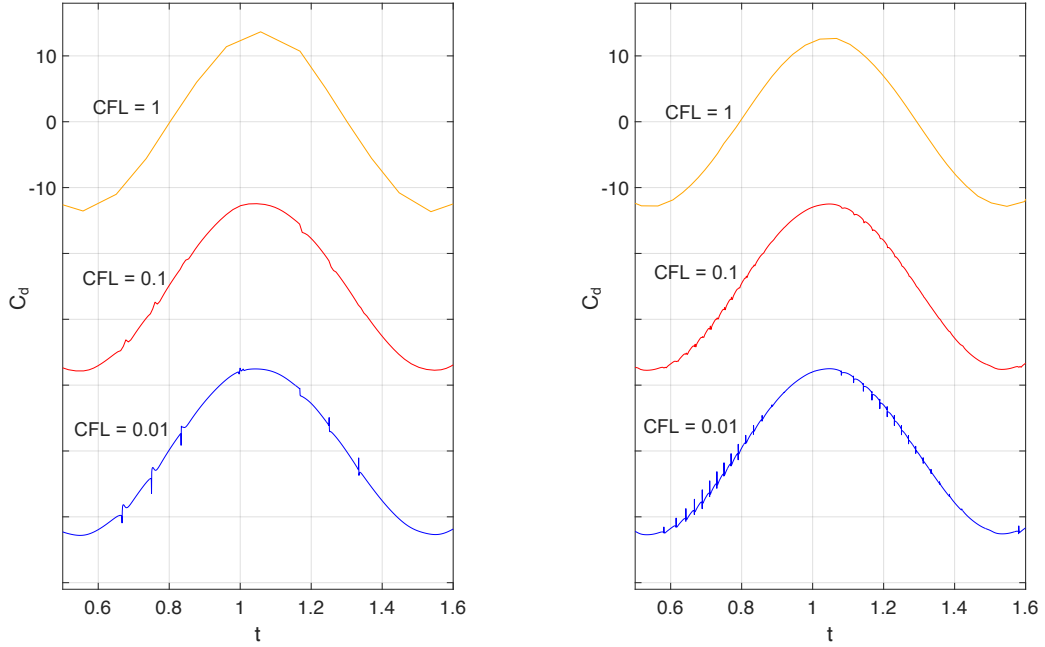


Figure 22: Oscillating square in a fluid at rest. C_d time histories for various CFL numbers and two grid sizes: $D/h = 16$ (left) and $D/h = 64$ (right).

Figures 21b and 21c display the velocity and pressure profiles at time $t = 1.5T$. These velocity and pressure solutions are smooth in space: observe the velocity profile around the square and near sharp corners, and the pressure exerted by the fluid in proximity of the sides of the square.

Figure 22 shows the time histories of the pressure drag coefficient C_d defined by (41) for various CFL numbers and the same two grid resolutions already utilized for the oscillating cylinder test. These plots should be compared with the ones in Figures 17. For $CFL = 0.01$, pressure oscillations are larger in the oscillating square test than in the oscillating cylinder test, while CFL s in the range $[0.1, 1.0]$ deliver accurate solutions with relatively small oscillations.

Figure 23 shows the decay of pressure oscillations as the CFL is increased, for the two grid sizes considered in Figure 22. Also in this case, $\|\Theta(C_d)\|_{\ell^\infty}$ scales roughly like CFL^{-1} , similar to the case of the oscillating cylinder test. In summary, as in the previous battery of tests on the circular cylinder, also the present test demonstrate that the WSBM is a viable approach for Stokes flow simulations involving moving boundaries.

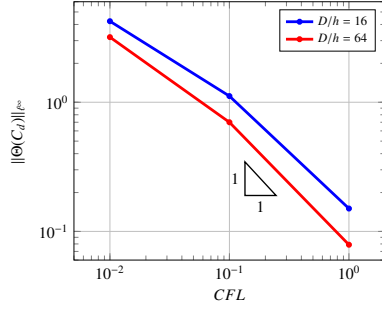


Figure 23: Oscillating square in a fluid at rest. Variation of $\|\Theta(C_d)\|_{\ell^\infty}$ as a function of CFL , for two grid sizes: $D/h = 16$ (blue) and $D/h = 64$ (red). Numerical values are shown on the table on the right.

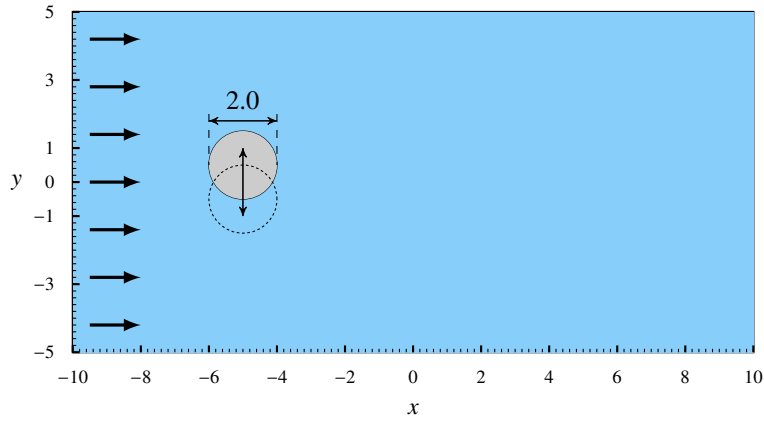


Figure 24: Oscillating cylinder in cross flow. Test configuration: a vertically oscillating circular cylinder is immersed in a rectangular flow domain.

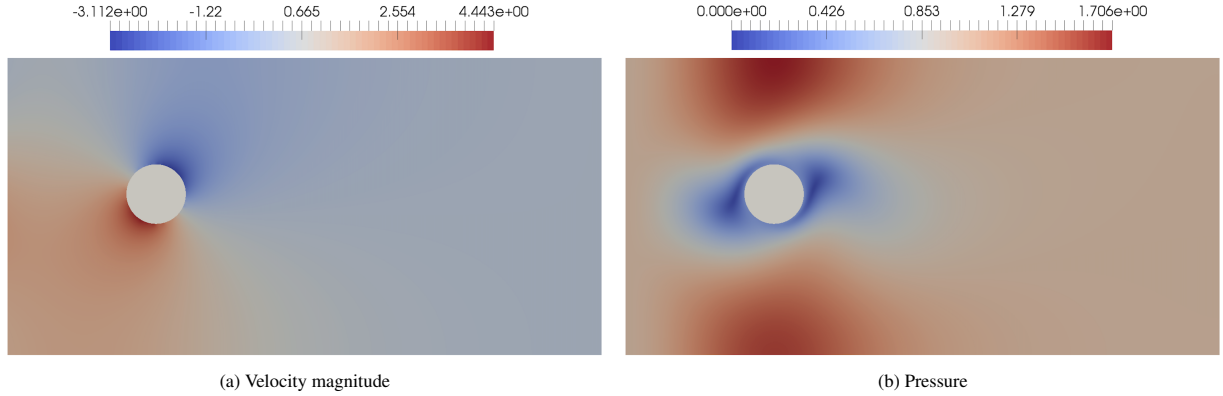


Figure 25: Oscillating cylinder in cross flow. Velocity magnitude (left) and pressure field (right) at time $t = 2T$.

5.5. Oscillating cylinder in cross flow

We consider next the simulation of an oscillating circular cylinder in cross flow, a benchmark proposed in [36] in the context of the incompressible Navier-Stokes equations. A cylinder of diameter $D = 2.0$ is immersed in the rectangular domain $[-10, 10] \times [-5, 5] \text{ m}^2$, as depicted in Figure 24. The fluid density is $\rho = 1.0 \text{ kg/m}^3$, and the dynamic viscosity is $\mu = 1.0 \text{ Kg/(m s)}$.

The initial position of the cylinder center is set to $(0, -X_0)$, where $X_0 = 0.25D = 0.5 \text{ m}$. The cylinder translates

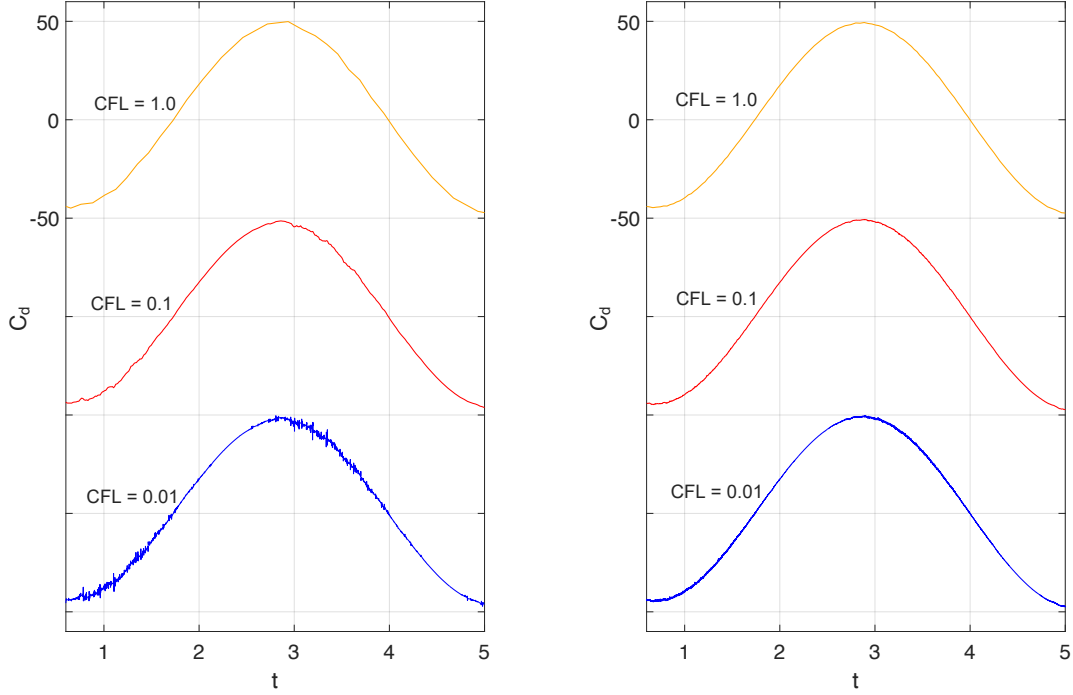


Figure 26: Oscillating cylinder in cross flow. C_d time histories for different CFL values, for a grid of size $D/h = 20$ (left) and $D/h = 80$ (right).

rigidly, following a motion expressed in terms of the location and velocity of its barycenter:

$$\mathbf{x}_c(t) = (-5.0, X_0 \cos(2\pi f_0 t)) , \quad (51a)$$

$$\mathbf{u}_c(t) = (0, -U_0 \sin(2\pi f_0 t)) , \quad (51b)$$

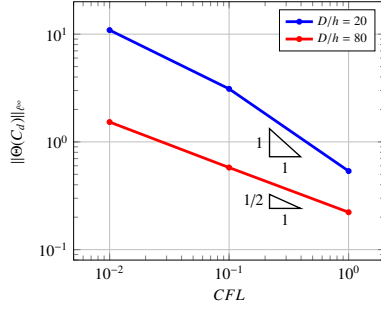
where $U_0 = 2\pi f_0 X_0$ m/s is the velocity scale, $f_0 = 0.2237$ Hz is the oscillation frequency, and $T = 1/f_0 \approx 4.5$ s is the oscillation period.

The nominal Reynolds number associated with the boundary motion is then $Re = \rho U_0 D / \mu = 1.41$. This is just a formal definition since we recall again that there are no convective terms in the Stokes flow equations. A slip boundary condition $u_y = 0.0$ m/s is enforced at the upper and lower boundaries of the rectangular domain. An inflow boundary condition is enforced on the left side by means of the Dirichlet condition $\mathbf{u}_{\text{left}} = \{1.0, 0.0\}$ m/s. At the outflow boundary, on the right side of the rectangle, the pressure boundary condition $p_{\text{out}} = 0.0$ N/m² is enforced.

Figure 25 illustrates the contours of the velocity magnitude and the pressure at time $t = 2T$, both of which are smooth, consistent with the theoretical expectations. The velocity field around the cylinder shows large gradients, associated with the interaction of the incoming flow with the surface of the moving cylinder, where a no-slip condition is enforced. The pressure profile is influenced by the flow pattern in the wake of the cylinder and gives some indication about the forces acting on the cylinder.

Figures 26 illustrates the time history of the drag coefficient for different choices of CFL numbers and two grids: a coarse grid with $D/h = 20$ and finer grid with $D/h = 80$. At $CFL = 1$, the oscillations are nearly absent for both grids, while some small oscillations start to creep in as the CFL approaches 0.1, especially for the coarse grid. Still, the simulations with CFL in the range $[0.1, 1]$ seem to perform well from an engineering perspective, since the oscillations are just a few percents of the peak values of the C_d .

Figure 27 shows the decay of pressure oscillations as the CFL is increased, for the two grid sizes considered in Figure 30. $\|\Theta(C_d)\|_{\ell^\infty}$ scales as $CFL^{-\zeta}$, for a power ζ which is slightly less than unity. This result is not necessarily a negative result, since on average the decrease of error between the coarse and the fine grid simulations is at least fourfold, consistent with the results already found in Figure 16 and Table 1. The slow growth of the oscillations on the fine grid as the CFL is reduced is actually a positive outcome. This may be due to the fact that, as already mentioned,



CFL	D/h = 20	D/h = 80
1	0.536	0.222
0.1	3.11	0.578
0.01	10.9	1.529

Figure 27: Oscillating cylinder in cross flow. Variation of $\|\Theta(C_d)\|_{\ell^\infty}$ as a function of CFL , for two grid sizes: $D/h = 20$ (blue) and $D/h = 80$ (red). Numerical values are shown on the table on the right.

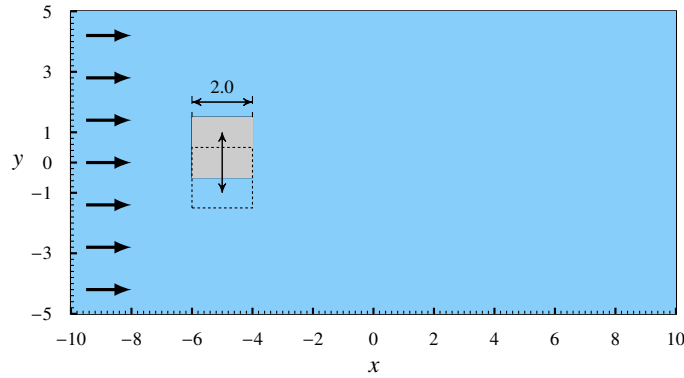


Figure 28: Oscillating square in cross flow. Test configuration: a vertically oscillating square cylinder is immersed in a rectangular flow domain.

several nodes are activated at the same time by the sweeping motion of the square rigid body. In summary, for CFL s in the range $[0.1, 1]$, the WSBM delivers results that have very mild oscillations, and can be used for predictive simulations of engineering applications.

5.6. Oscillating square in cross flow

This final test we present is more demanding than the benchmark described in Section 5.5. The setup is as follows:

- The geometric configuration, described in Figure 28, is almost identical, except for the fact that the cylinder has been replaced by a square with a side of length $D = 0.25$.
- The flow properties and grids utilized are the same, but this time the vertical sides of the square are perfectly aligned and overlapped with the grid lines, producing a multiplicative effect on the number of nodes that could be activated over a time step.
- The law of motion governing the displacement in time of the square is given again by (51a) and (51b).

Figure 29 shows the contours of the velocity magnitude and the pressure at time $t = 2T$, both of which are smooth away from the corners of the moving boundary, where singularities may occur.

Figure 30 plots the time history of the drag coefficient C_d for different choices of CFL numbers and two grids, with $D/h = 20$ and $D/h = 80$, respectively. These figure should be compared with Figure 26, relative to the test of the oscillating cylinder in cross flow. Oscillations are more pronounced in Figure 30, particularly for the course grid and with CFL is in the range $[0.01, 0.1]$. Also in this test, the oscillations of C_d are greatly reduced using the finer grid.

Figure 31 shows the decay of pressure oscillations as the CFL is increased, for the two grid sizes considered in Figure 30. $\|\Theta(C_d)\|_{\ell^\infty}$ scales as $CFL^{-\zeta}$, for a power ζ which is equal to unity for larger CFL numbers and less than unity for smaller CFL numbers. The fact that $\zeta < 1$ for smaller CFL s is actually a positive outcome, in the sense that

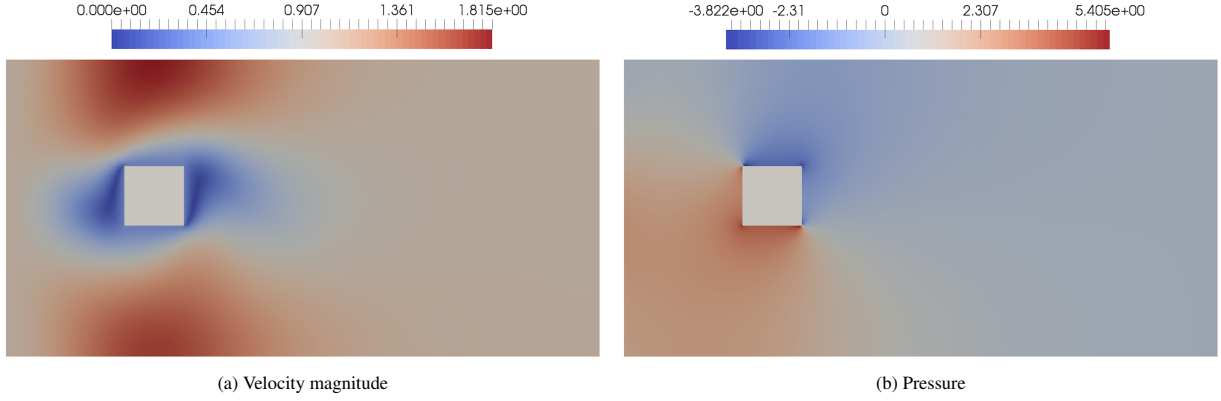


Figure 29: Oscillating square in cross flow. Velocity magnitude (left) and pressure field (right) at time $t = 2T$.

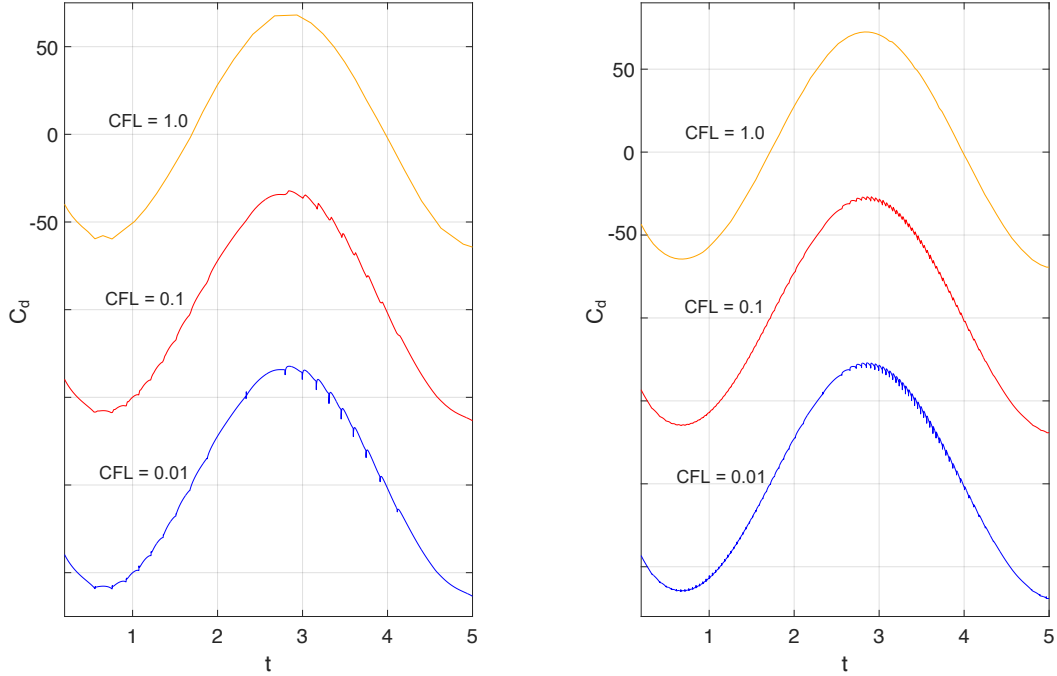
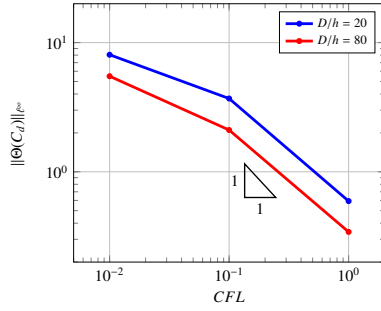


Figure 30: Oscillating square in cross flow. C_d time histories for different CFL values with fixed mesh size $D/h = 20$ and $D/h = 80$.

the growth of the oscillations is less than what was reported with other methods in prior works. Also for this test, for CFL s in the range $[0.1, 1]$, the WSBM delivers results that have very mild oscillations, and can be used for predictive simulations of engineering applications.

6. Conclusions

In this paper, a Weighted Shifted Boundary Method (W-SBM) is proposed for the simulation of immersed moving domain with Dirichlet-type (e.g., no-slip) boundary conditions. The present work complements the work of Colomés *et al.* [86], where Neumann (traction) boundary conditions were discussed. The WSBM preserves the total volume of active fluids with drastically improved accuracy with respect to the SBM and produces small errors in the total mass and momentum conservation budgets, which converge at least with quadratic and linear rates, respectively. The proposed WSBM is capable of exactly representing states of hydrostatic equilibrium, and produces nodal forces that



CFL	D/h = 20	D/h = 80
1	0.593	0.342
0.1	3.69	2.11
0.01	8.06	5.49

Figure 31: Oscillating square in cross flow. Variation of $\|\Theta(C_d)\|_{\ell^\infty}$ as a function of CFL , for two grid sizes: $D/h = 20$ (blue) and $D/h = 80$ (red). Numerical values are shown on the table on the right.

smoothly transition as the true boundary sweeps across the computational grid. The performance of the WSBM was evaluated with a battery of numerical experiments, aimed at testing its accuracy and robustness, particularly with regard to pressure oscillations in time that might occur in the limit of small time steps. The evidence from the numerical tests corroborates the conclusion that the SBM is a viable, accurate, and robust approach for moving boundary simulations of Stokes flow. The natural evolution of the presented work is the extension of the WSBM to three dimensions and to the Navier-Stokes equations for domains with rigid and flexible moving boundaries.

Acknowledgments

The support of the U.S. Office of Naval Research under grant N00014-14-1-0311, ExxonMobil Upstream Research Company (Houston, TX), and the National Science Foundation under Grant 2207164 (Division of Mathematical Sciences – DMS), are gratefully acknowledged. Part of Nabil M. Atallah’s contributions have been performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, LLNL-JRNL-854220.

References

- [1] William F Noh. CEL: A time-dependent, two-space-dimensional, coupled Eulerian-Lagrange code. Technical report, Lawrence Radiation Lab., Univ. of California, Livermore, 1963.
- [2] Charles S Peskin. Numerical analysis of blood flow in the heart. *Journal of computational physics*, 25(3):220–252, 1977.
- [3] Charles S Peskin. The Immersed Boundary Method. *Acta numerica*, 11:479–517, 2002.
- [4] Roland Glowinski, Tsorng-Whay Pan, and Jacques Periaux. Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies. *Computer methods in applied mechanics and engineering*, 151(1-2):181–194, 1998.
- [5] Daniele Boffi and Lucia Gastaldi. A finite element approach for the immersed boundary method. *Computers & Structures*, 81(8):491 – 501, 2003. K.J Bathe 60th Anniversary Issue.
- [6] Ever A Fadlun, Roberto Verzicco, Paolo Orlandi, and Jamaludin Mohd-Yusof. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of computational physics*, 161(1):35–60, 2000.
- [7] Fotis Sotiropoulos and Xiaolei Yang. Immersed boundary methods for simulating fluid–structure interaction. *Progress in Aerospace Sciences*, 65:1–21, 2014.
- [8] Zeng X. and Farhat C. A systematic approach for constructing higher-order immersed boundary and ghost fluid methods for fluid–structure interaction problems. *Journal of Computational Physics*, 231(7):2892–2923, 2012.
- [9] Jung Hee Seo and Rajat Mittal. A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations. *Journal of computational physics*, 230(19):7347–7363, 2011.
- [10] Dokyun Kim and Haechon Choi. Immersed boundary method for flow around an arbitrarily moving body. *Journal of Computational Physics*, 212(2):662–680, 2006.
- [11] Jongho Lee, Jungwoo Kim, Haechon Choi, and Kyung-Soo Yang. Sources of spurious force oscillations from an immersed boundary method for moving-body problems. *Journal of Computational Physics*, 230(7):2677–2695, 2011.
- [12] Chuan-Chieh Liao, Yu-Wei Chang, Chao-An Lin, and J.M. McDonough. Simulating flows with moving rigid boundary using immersed-boundary method. *Computers & Fluids*, 39(1):152–167, 2010.
- [13] Jinmo Lee and Donghyun You. An implicit ghost-cell immersed boundary method for simulations of moving body problems with control of spurious force oscillations. *Journal of Computational Physics*, 233:295–314, 2013.
- [14] Lennart Schneiders, Daniel Hartmann, Matthias Meinke, and Wolfgang Schröder. An accurate moving boundary formulation in cut-cell methods. *Journal of Computational Physics*, 235:786–809, 2013.

- [15] Yu-Heng Tseng and Joel H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192(2):593–623, 2003.
- [16] Jianming Yang and Elias Balaras. An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries. *Journal of Computational Physics*, 215(1):12–40, 2006.
- [17] William D Henshaw. A fourth-order accurate method for the incompressible Navier–Stokes equations on overlapping grids. *Journal of computational physics*, 113(1):13–25, 1994.
- [18] W Henshaw and NA Petersson. A split-step scheme for the incompressible Navier–Stokes equations. Technical report, Lawrence Livermore National Lab., CA (US), 2001.
- [19] SP Domino. Towards verification of sliding mesh algorithms for complex applications using MMS. In *Proceedings of the Summer Program*, page 167, 2010.
- [20] Jeffrey W Banks, William D Henshaw, and Donald W Schwendeman. An analysis of a new stable partitioned algorithm for fsi problems. Part I: Incompressible flow and elastic solids. *Journal of Computational Physics*, 269:108–137, 2014.
- [21] Qi Tang. A stable partitioned FSI algorithm for rigid bodies and incompressible flow. Part I: Model problem analysis. *Journal of Computational Physics*, 343:432–468, 2017.
- [22] Qi Tang. A stable partitioned fsi algorithm for rigid bodies and incompressible flow in three dimensions. *Journal of Computational Physics*, 373:455–492, 2018.
- [23] Klaus Höllig, Ulrich Reif, and Joachim Wipper. Weighted extended B-spline approximation of Dirichlet problems. *SIAM Journal on Numerical Analysis*, 39(2):442–462, 2001.
- [24] Anita Hansbo and Peter Hansbo. An unfitted finite element method, based on Nitsche’s method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering*, 191(47):5537 – 5552, 2002.
- [25] Klaus Hollig. Finite element methods with B-splines, volume 26. Siam, 2003.
- [26] Jamshid Parvizian, Alexander Düster, and Ernst Rank. Finite Cell Method. *Computational Mechanics*, 41(1):121–133, 2007.
- [27] Alexander Düster, Jamshid Parvizian, Zhengxiong Yang, and Ernst Rank. The finite cell method for three-dimensional problems of solid mechanics. *Computer methods in applied mechanics and engineering*, 197(45-48):3768–3782, 2008.
- [28] Erik Burman and Peter Hansbo. Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method. *Computer Methods in Applied Mechanics and Engineering*, 199(41-44):2680–2686, 2010.
- [29] Erik Burman. Ghost penalty. *Comptes Rendus Mathématique*, 348(21-22):1217–1220, 2010.
- [30] Bernardo Cockburn, Deepa Gupta, and Fernando Reith. Boundary-conforming discontinuous Galerkin methods via extensions from subdomains. *Journal of Scientific Computing*, 42(1):144, 2010.
- [31] Erik Burman and Peter Hansbo. Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. *Applied Numerical Mathematics*, 62(4):328–341, 2012.
- [32] Dominik Schillinger, Luca Dede, Michael A Scott, John A Evans, Michael J Borden, Ernst Rank, and Thomas JR Hughes. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and t-spline cad surfaces. *Computer Methods in Applied Mechanics and Engineering*, 249–252:116 – 150, 2012.
- [33] Bernardo Cockburn and Manuel Solano. Solving Dirichlet boundary-value problems on curved domains by extensions from subdomains. *SIAM Journal on Scientific Computing*, 34(1):A497–A519, 2012.
- [34] Joan Baiges, Ramon Codina, Florian Henke, Shadan Shahmiri, and Wolfgang A Wall. A symmetric method for weakly imposing Dirichlet boundary conditions in embedded finite element meshes. *International Journal for Numerical Methods in Engineering*, 90(5):636–658, 2012.
- [35] Thomas Rüberg and F Cirak. Subdivision-stabilised immersed b-spline finite elements for moving boundary flows. *Computer Methods in Applied Mechanics and Engineering*, 209:266–283, 2012.
- [36] T Rüberg and F Cirak. A fixed-grid b-spline finite element technique for fluid–structure interaction. *International Journal for Numerical Methods in Fluids*, 74(9):623–660, 2014.
- [37] 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.
- [38] Bernardo Cockburn, Weifeng Qiu, and Manuel Solano. A priori error analysis for HDG methods using extensions from subdomains to achieve boundary conformity. *Mathematics of Computation*, 83(286):665–699, 2014.
- [39] André Massing, Mats Larson, Anders Logg, and Marie Rognes. A Nitsche-based cut finite element method for a fluid-structure interaction problem. *Communications in Applied Mathematics and Computational Science*, 10(2):97–120, 2015.
- [40] B Schott, U Rasthofer, V Gravemeier, and WA Wall. A face-oriented stabilized Nitsche-type extended variational multiscale method for incompressible two-phase flow. *International Journal for Numerical Methods in Engineering*, 104(7):721–748, 2015.
- [41] Erik Burman, Susanne Claus, Peter Hansbo, Mats G Larson, and André Massing. CutFEM: discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering*, 104(7):472–501, 2015.
- [42] Fei Xu, Dominik Schillinger, David Kamensky, Vasco Varduhn, Chenglong Wang, and Ming-Chen Hsu. The tetrahedral finite cell method for fluids: Immersogeometric analysis of turbulent flow around complex geometries. *Computers & Fluids*, 141:135–154, 2016.
- [43] Alexei Lozinski. A new fictitious domain method: Optimal convergence without cut elements. *Comptes Rendus Mathématique*, 354(7):741–746, 2016.
- [44] WG Dettmer, C Kadapa, and D Perić. A stabilised immersed boundary method on hierarchical B-spline grids. *Computer Methods in Applied Mechanics and Engineering*, 311:415–437, 2016.
- [45] C Kadapa, WG Dettmer, and D Perić. A stabilised immersed boundary method on hierarchical B-spline grids for fluid–rigid body interaction with solid–solid contact. *Computer Methods in Applied Mechanics and Engineering*, 318:242–269, 2017.
- [46] Erik Burman, Peter Hansbo, and Mats G Larson. A cut finite element method with boundary value correction for the incompressible Stokes equations. In *European Conference on Numerical Mathematics and Advanced Applications*, pages 183–192. Springer, 2017.
- [47] David Kamensky, Ming-Chen Hsu, Yue Yu, John A Evans, Michael S Sacks, and Thomas JR Hughes. Immersogeometric cardiovascular fluid–structure interaction analysis with divergence-conforming B-splines. *Computer Methods in Applied Mechanics and Engineering*, 314:408–472, 2017.

- [48] Erik Burman, Peter Hansbo, and Mats Larson. A cut finite element method with boundary value correction. *Mathematics of Computation*, 87(310):633–657, 2018.
- [49] Erik Burman, Daniel Elfverson, Peter Hansbo, Mats G Larson, and Karl Larsson. Shape optimization using the cut finite element method. *Computer Methods in Applied Mechanics and Engineering*, 328:242–261, 2018.
- [50] C Kadapa, WG Dettmer, and D Perić. A stabilised immersed framework on hierarchical b-spline grids for fluid-flexible structure interaction with solid–solid contact. *Computer Methods in Applied Mechanics and Engineering*, 335:472–489, 2018.
- [51] Erik Burman, Peter Hansbo, and Mats G Larson. Dirichlet boundary value correction using Lagrange multipliers. *arXiv preprint arXiv:1903.07104*, 2019.
- [52] Marc Núñez, Iñigo López, Joan Baiges, and Riccardo Rossi. An embedded approach for the solution of the full potential equation with finite elements. *Computer Methods in Applied Mechanics and Engineering*, 388:114244, 2022.
- [53] Rubén Zorrilla, Antonia Larese de Tetto, and Riccardo Rossi. A discontinuous Nitsche-based finite element formulation for the imposition of the Navier-slip condition over embedded volumeless geometries. *International Journal for Numerical Methods in Fluids*, 93(9):2968–3003, 2021.
- [54] Rubén Zorrilla, Eduardo Soudah, and Riccardo Rossi. Computational modeling of the fluid flow in type B aortic dissection using a modified finite element embedded formulation. *Biomechanics and modeling in mechanobiology*, 19:1565–1583, 2020.
- [55] Rubén Zorrilla, Riccardo Rossi, Roland Wüchner, and Eugenio Oñate. An embedded finite element framework for the resolution of strongly coupled fluid–structure interaction problems. application to volumetric and membrane-like structures. *Computer Methods in Applied Mechanics and Engineering*, 368:113179, 2020.
- [56] Soonpil Kang and Arif Masud. A variational multiscale method with immersed boundary conditions for incompressible flows. *Meccanica*, 56(6):1397–1422, 2021.
- [57] Soonpil Kang, JaeHyuk Kwack, and Arif Masud. Variational coupling of non-matching discretizations across finitely deforming fluid–structure interfaces. *International Journal for Numerical Methods in Fluids*, 94(6):678–718, 2022.
- [58] J. A. Nitsche. Über ein Variationsprinzip zur Lösung Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abh. Math. Sem. Univ., Hamburg*, 36:9–15, 1971.
- [59] Erik Burman and Peter Hansbo. Fictitious domain methods using cut elements: III. A stabilized Nitsche method for Stokes’ problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(3):859–874, 2014.
- [60] B Schott and WA Wall. A new face-oriented stabilized XFEM approach for 2D and 3D incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 276:233–265, 2014.
- [61] Ramon Codina, Guillaume Houzeaux, Herbert Coppola-Owen, and Joan Baiges. The fixed-mesh ALE approach for the numerical approximation of flows in moving domains. *Journal of Computational Physics*, 228(5):1591–1611, 2009.
- [62] Joan Baiges and Ramon Codina. The fixed-mesh ALE approach applied to solid mechanics and fluid–structure interaction problems. *International journal for numerical methods in engineering*, 81(12):1529–1557, 2010.
- [63] Joan Baiges, Ramon Codina, and Herbert Coppola-Owen. The fixed-mesh ALE approach for the numerical simulation of floating solids. *International Journal for Numerical Methods in Fluids*, 67(8):1004–1023, 2011.
- [64] Joan Baiges, Ramon Codina, Arnau Pont, and Ernesto Castillo. An adaptive fixed-mesh ALE method for free surface flows. *Computer Methods in Applied Mechanics and Engineering*, 313:159–188, 2017.
- [65] Ramsharan Rangarajan and Adrian J Lew. Analysis of a method to parameterize planar curves immersed in triangulations. *SIAM Journal on Numerical Analysis*, 51(3):1392–1420, 2013.
- [66] Ramsharan Rangarajan and Adrián J Lew. Universal meshes: A method for triangulating planar curved domains immersed in nonconforming meshes. *International Journal for Numerical Methods in Engineering*, 98(4):236–264, 2014.
- [67] Evan S Gawlik and Adrian J Lew. High-order finite element methods for moving boundary problems with prescribed boundary evolution. *Computer Methods in Applied Mechanics and Engineering*, 278:314–346, 2014.
- [68] Evan S Gawlik, Hardik Kabaria, and Adrian J Lew. High-order methods for low Reynolds number flows around moving obstacles based on universal meshes. *International Journal for Numerical Methods in Engineering*, 104(7):513–538, 2015.
- [69] Alex Main and Guglielmo Scovazzi. The Shifted Boundary Method for embedded domain computations. Part I: Poisson and Stokes problems. *Journal of Computational Physics*, 372:972–995, 2018.
- [70] Alex Main and Guglielmo Scovazzi. The Shifted Boundary Method for embedded domain computations. Part II: Linear advection–diffusion and incompressible Navier–Stokes equations. *Journal of Computational Physics*, 372:996–1026, 2018.
- [71] Ting Song, Alex Main, Guglielmo Scovazzi, and Mario Ricchiuto. The Shifted Boundary Method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows. *Journal of Computational Physics*, 369:45–79, 2018.
- [72] Nabil M. Atallah, Claudio Canuto, and Guglielmo Scovazzi. The Shifted Boundary Method for solid mechanics. *International Journal for Numerical Methods in Engineering*, 122(20):5935–5970, 2021.
- [73] Leo Nouveau, Mario Ricchiuto, and Guglielmo Scovazzi. High-order gradients with the Shifted Boundary Method: An embedded enriched mixed formulation for elliptic PDEs. *Journal of Computational Physics*, 398:108898, 2019.
- [74] Efthymios N Karatzas, Giovanni Stabile, Nabil Atallah, Guglielmo Scovazzi, and Gianluigi Rozza. A reduced order approach for the embedded Shifted Boundary FEM and a heat exchange system on parametrized geometries. In *IUTAM Symposium on Model Order Reduction of Coupled Systems*, Stuttgart, Germany, May 22–25, 2018, pages 111–125. Springer, 2020.
- [75] Efthymios N Karatzas, Giovanni Stabile, Leo Nouveau, Guglielmo Scovazzi, and Gianluigi Rozza. A reduced basis approach for PDEs on parametrized geometries based on the Shifted Boundary Finite Element Method and application to a Stokes flow. *Computer Methods in Applied Mechanics and Engineering*, 347:568–587, 2019.
- [76] Efthymios N Karatzas, Giovanni Stabile, Leo Nouveau, Guglielmo Scovazzi, and Gianluigi Rozza. A Reduced-order Shifted Boundary Method for parametrized incompressible Navier–Stokes equations. *arXiv preprint arXiv:1907.10549*, 2019.
- [77] Nabil M Atallah, Claudio Canuto, and Guglielmo Scovazzi. Analysis of the Shifted Boundary Method for the Stokes problem. *Computer Methods in Applied Mechanics and Engineering*, 358:112609, 2020.
- [78] Nabil M. Atallah, Claudio Canuto, and Guglielmo Scovazzi. The second-generation Shifted Boundary Method and its numerical analysis.

- Computer Methods in Applied Mechanics and Engineering, 372:113341, 2020.
- [79] N. M. Atallah, C. Canuto, and G. Scovazzi. Analysis of the Shifted Boundary Method for the Poisson problem in domains with corners. Math. Comp., 90:2041–2069, 2021.
 - [80] Nabil M. Atallah, Claudio Canuto, and Guglielmo Scovazzi. The high-order Shifted Boundary Method and its analysis. Computer Methods in Applied Mechanics and Engineering, 394:114885, 2022.
 - [81] J. Haydel Collins, Alexei Lozinski, and Guglielmo Scovazzi. A penalty-free Shifted Boundary Method of arbitrary order. Computer Methods in Applied Mechanics and Engineering, page 116301, 2023.
 - [82] Kangan Li, Nabil M. Atallah, G. Alex Main, and Guglielmo Scovazzi. The Shifted Interface Method: A flexible approach to embedded interface computations. International Journal for Numerical Methods in Engineering, 121(3):492–518, 2020.
 - [83] Kangan Li, Nabil M Atallah, Antonio Rodríguez-Ferran, Dakshina M Valivet, and Guglielmo Scovazzi. The Shifted Fracture Method. International Journal for Numerical Methods in Engineering, 122(22):6641–6679, 2021.
 - [84] Kangan Li, Antonio Rodríguez-Ferran, and Guglielmo Scovazzi. A blended Shifted-Fracture/Phase-Field framework for sharp/diffuse crack modeling. International Journal for Numerical Methods in Engineering, 124(4):998–1030, 2023.
 - [85] Kangan Li, Antonio Rodríguez-Ferran, and Guglielmo Scovazzi. The Simple Shifted Fracture Method. International Journal for Numerical Methods in Engineering, 124(12):2837–2875, 2023.
 - [86] Oriol Colomés, Alex Main, Léo Nouveau, and Guglielmo Scovazzi. A Weighted Shifted Boundary Method for free surface flow problems. Journal of Computational Physics, 424:109837, 2021.
 - [87] H. Dätsch, F. Durst, S. Becker, and H. Lienhart. Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan-Carpenter numbers. Journal of Fluid Mechanics, 360:249–271, 1998.
 - [88] L. Wang, Z.L. Guo, and J.C. Mi. Drafting, kissing and tumbling process of two particles with different sizes. Computers & Fluids, 96:20–34, 2014.
 - [89] Kevin Wang, J Grétarsson, A Main, and C Farhat. Computational algorithms for tracking dynamic fluid–structure interfaces in embedded boundary methods. International Journal for Numerical Methods in Fluids, 70(4):515–535, 2012.
 - [90] Nabil M. Atallah, Claudio Canuto, and Guglielmo Scovazzi. Analysis of the Weighted Shifted Boundary Method for the Poisson and Stokes problems, 2023.
 - [91] Thomas J. R. Hughes and Leopoldo P. Franca. A new finite element formulation for computational fluid dynamics: VII. The Stokes problem with various well-posed boundary conditions: Symmetric formulations that converge for all velocity/pressure spaces. Computer methods in applied mechanics and engineering, 65(1):85 – 96, 1987.
 - [92] Thomas J. R. Hughes. Multiscale phenomena: Green’s functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. Comput. Methods Appl. Mech. Eng., 127(1–4):387–401, November 1995.
 - [93] Thomas J. R. Hughes, Gonzalo R. Feijóo, Luca Mazzei, and Jean-Baptiste Quincy. The variational multiscale method – a paradigm for computational mechanics. Comput. Methods Appl. Mech. Eng., 166(1–2):3–24, November 1998.
 - [94] Thomas JR Hughes, Guglielmo Scovazzi, and Leopoldo P Franca. Multiscale and stabilized methods. Encyclopedia of computational mechanics, 2004.