TU Delft

Delft University of Technology

A weighted shifted boundary method for the Navier-Stokes equations with immersed moving boundaries

Xu, Danjie; Colomés, Oriol; Main, Alex; Li, Kangan; Atallah, Nabil M.; Abboud, Nabil; Scovazzi, Guglielmo

DOI
10.1016/j.jcp.2025.114571

Publication date
2025

Document Version
Final published version

Published in
Journal of Computational Physics

Citation (APA)
Xu, D., Colomés, O., Main, A., Li, K., Atallah, N. M., Abboud, N., & Scovazzi, G. (2025). A weighted shifted boundary method for the Navier-Stokes equations with immersed moving boundaries. *Journal of Computational Physics*, *548*, Article 114571. https://doi.org/10.1016/j.jcp.2025.114571

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository
as part of the Taverne amendment.

More information about this copyright law amendment
can be found at https://www.openaccess.nl.

Otherwise as indicated in the copyright section:
the publisher is the copyright holder of this work and the
author uses the Dutch legislation to make this work public.

Journal of Computational Physics 548 (2026) 114571

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A weighted shifted boundary method for the Navier-Stokes equations with immersed moving boundaries

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

[a] Department of Civil and Environmental Engineering, Duke University, 27708, Durham, NC, USA
[b] Faculty of Civil Engineering and Geosciences, Delft University of Technology, Stevinweg 1, 2628 CN, Delft, The Netherlands
[c] 15317, Ansys Canonsburg, PA, USA
[d] College of Earth and Mineral Sciences, The Pennsylvania State University, 16801, University Park, PA, USA
[e] Lawrence Livermore National Laboratory, 94550, Livermore, CA, USA
[f] 30361, Ansys Atlanta, GA, USA

a r t i c l e i n f o

Keywords:
Immersed boundary method
Unfitted finite element method
Approximate boundary
Moving boundaries
Computational fluid dynamics
Shifted boundary method

a b s t r a c t

The Weighted Shifted Boundary Method (WSBM) was recently introduced as an enhanced Shifted Boundary Method (SBM) for the simulation of flows with moving boundaries. Earlier work of the authors on no-slip boundary conditions for the two-dimensional Stokes flow is extended here to the more challenging case of the three-dimensional incompressible Navier-Stokes equations at low and moderate Reynolds numbers. The SBM is an immersed finite element method that reformulates an infinite-dimensional boundary value problem over a surrogate (approximate) computational domain – to avoid integrating over cut cells – and modifies the original boundary conditions using Taylor expansions – to maintain accuracy. The WSBM weights the SBM's variational form with the elemental volume fraction of active fluid, drastically reducing spurious pressure oscillations in time that occur when the total volume of active fluid changes abruptly over a time step. The WSBM induces small mass (i.e., volume) conservation errors, which converge quadratically in the case of piecewise-linear finite element interpolations, as the grid is refined. An extensive set of two- and three-dimensional tests demonstrates the robustness and accuracy of the proposed approach.

1. Introduction

Flows interacting with moving boundaries are commonly found in nature and in practical engineering applications. In many cases, the shapes of immersed moving objects involve complex geometries, which, sometimes, are not even described by standard Computer Aided Design (CAD) formats. If imperfect, non watertight representations are used for the geometry (e.g., Standard Tessellation Language, or STL), meshing may result unfeasible. Immersed (or embedded, or unfitted) domain methods play an important role in this context, by making meshing more flexible and reducing the grid-generation time. More specifically, because immersed methods can make the whole mesh generation process highly automated, they are the perfect candidate for the training of digital twins involving geometric parametrizations.

A brief and possibly incomplete account of the many contributions on immersed/embedded methods for moving boundary simulations is attempted next, with specific focus on finite element methods. Immersed Boundary Methods originated from the seminal work of Noh [1] and Peskin [2,3], in the context of finite difference and finite volume methods. In [2,3] the geometry to be simulated is immersed in a pre-existing

* Corresponding author.
E-mail addresses: danjie.xu@duke.edu (D. Xu), j.o.colomesgene@tudelft.nl (O. Colomés), geoffrey.main@duke.edu (A. Main), kangan.li@duke.edu (K. Li), atallah1@llnl.gov (N.M. Atallah), nabil.abboud@duke.edu (N. Abboud), guglielmo.scovazzi@duke.edu (G. Scovazzi).

https://doi.org/10.1016/j.jcp.2025.114571
Received 2 May 2025; Received in revised form 27 September 2025; Accepted 3 December 2025
Available online 8 December 2025
0021-9991/© 2025 Elsevier Inc. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

grid and boundary conditions are mimicked by forcing terms in the governing equations, in the form of discrete, smoothed, Dirac delta functions. Hence, a fictitious fluid flow is computed also inside immersed bodies, to avoid computational complexities in bookkeeping active fluid cells in the computational domain. These ideas were later applied to finite element methods in [4–10].

In the late 1900s and early 2000s, Discrete Forcing Immersed Boundary Methods [11–22] emerged in the finite volume method community, in which no-slip boundary conditions are strongly enforced using source terms in the discrete equations. These methods do not introduce a fictitious fluid inside the immersed bodies and delivered improved accuracy. Discrete Forcing Immersed Boundary Methods were an alternative to overset grids [23–28], with analogies to the solution/mesh tying strategies of the latter.

Around the 2000s, a series of new unfitted finite element methods [29–77] appeared, which also employed a sharp interface approximation but imposed boundary conditions weakly, through Lagrange multipliers or Nitsche's method [78]. Most unfitted finite element methods integrate the variational equations over grids that may contain cut elements, from which the naming cutFEMs [47]. The integration over cut cells requires more involved data structures as well as specific algebraic preconditioning or numerical stabilization [35], when particularly small cut cells are produced. Discrete Forcing Immersed Boundary Methods also suffer from similar difficulties. Numerical stabilization of the small-cut cell problem takes the form of ghost penalty method [35,46,79], or Extended B-splines [29,31,41,42]. Integration on cut cells is performed by subtriangulation [30,62,64–68], adaptive octree quadratures [32,33], or advanced quadratures based on moment-fitting [80,81] and the Gauss-divergence theorem [82–85]: oftentimes a non-negligible portion of the simulation wall-clock time is spent handling the embedded/immersed boundary.

An alternative to unfitted finite elements is the Fixed-Mesh ALE Method [86–89], in which the grid near the immersed boundaries is repositioned using a local Arbitrary Lagrangian Eulerian approach with remeshing, or the Method of Universal Meshes [90–93], which utilizes dynamic remeshing around a baseline fixed grid, in close proximity of the immersed boundary. Both the Fixed-Mesh ALE Method and the Method of Universal Meshes are not immersed methods 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 methods with approximate boundaries, a subclass of sharp interface/boundary representations. 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 preserve the accuracy of the overall formulation. The appropriate (modified) boundary conditions are 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 [94] and generalized in [95] to the advection-diffusion and Navier-Stokes equations, in [96] to hyperbolic conservation laws, and in [97] to the equations of static linear elasticity. The benefits of the SBM in the context of reduced order modeling was analyzed in [98–100]. Further rigorous mathematical analysis was pursued in [101,102] for the Poisson and the Stokes flow problems, and in [103] for the treatment of general domains with corners. A high-order version of the SBM and its numerical analysis was presented in [104], in the context of the Poisson and Stokes flow equations, and in [105], for a penalty free variant.

While most of the works on SBM utilized unstructured triangular/tetrahedral grids in two/three dimensions, it was shown already in [95] that this method can successfully be applied to Cartesian grids. A more recent study [106] described a methodology for the optimal selection of the surrogate boundary in the case of Cartesian/Octree grids, which was later tested in an extensive set of computational benchmarks for the incompressible Navier-Stokes equations [107,108], also including natural and forced thermal convection [109].

Li et al. further extended the SBM to problems with internal interfaces, under the names of Shifted Interface Method [110] and Shifted Fracture Method [111–114]. The authors of [97,105] 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 [115,116], 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 [14,15,23,24]. This is a known issue also in the realm of finite volume methods: Kim et al. [15] 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. [17] investigated the relationship between the internal numerical approximation of the solid object and the external behavior of flow; Lee et al. [16] found that the oscillations decrease as mesh size decreases or as computational time step size increases; Seo and Mittal [14] 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 [18–21,42,117,118]. We also point the reader to a recent and very interesting work of Olshanskii and von Wahl [119] on the connection between the change in time of the active domain and corresponding discrete approximation spaces, and the implications on the LBB (inf-sup) condition, and pressure oscillations in time.

In the present work, we modify the original SBM method [94,95], by "weighting" the SBM variational form with 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 [115] for general Neumann (i.e., traction) boundary conditions with some focused applications to free-surface flows, and in [116] in the case of the Stokes flow with no-slip boundary conditions imposed on moving boundaries.

The present work extends the no-slip boundary condition developments in [116] to the three-dimensional incompressible Navier-Stokes equations, which are considerably more challenging to solve than Stokes flow. We also note that the robust implementation of no-slip (Dirichlet-type) boundary conditions is considerably more challenging than the one of traction-free boundary conditions for the WSBM, since there is no ex-

plicit control on the pressure at the boundary, for the former. The presence of the inertia terms in the Navier-Stokes equations also exacerbate the occurrence of pressure oscillations in time, with respect to the case of Stokes flow.

Together with [115,116], the present work is a comprehensive description of the derivation and implementation of the WSBM for the most common boundary conditions encountered in computational fluid dynamics. In addition, we derive estimates for the mass and momentum conservation errors produced by the WSBM, showing that they converge as the grid is refined and they are typically small. We also prove that the WSBM exactly preserves states of hydrostatic equilibrium (i.e., within machine precision). We demonstrate in a number of numerical tests that the WSBM drastically reduces pressure oscillations. Mathematical proofs of stability and optimal convergence of the WSBM in the context of the Poisson and Stokes problems with Dirichlet boundary conditions were derived in a recent preprint [120], to which we point the interested reader.

The rest of this paper is organized as follows: Section 2 introduces the initial/boundary-value problem associated with the incompressible Navier-Stokes equations; Section 3 details the derivations of the WSBM and its algorithmic properties in terms of conservation and related aspects; Section 4 presents the results of an extensive number of two- and three-dimensional tests; conclusions are summarized in Section 5.

2. Strong form of the Navier-Stokes equations

Let Ω be a bounded and connected open region in $\mathbb{R}^n$ ($n = 2$ or 3) with Lipschitz boundary $\Gamma = \partial\Omega$ (the blue curve in the sketch of Fig. 1a and b). Let $\boldsymbol{n}$ denote the outward-oriented unit normal vector to Γ (see again Fig. 1b). The Navier-Stokes equations for viscous incompressible flow read

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

$$\nabla\cdot\boldsymbol{u} = 0\,, \qquad \forall \boldsymbol{x}\in\Omega\,, \tag{1b}$$

where ρ is the density, $\boldsymbol{b}$ is a body force per unit mass (e.g., the gravity acceleration), $\boldsymbol{u}_{,t} = \partial\boldsymbol{u}/\partial t$ is the time derivative of the velocity, $\boldsymbol{\varepsilon}(\boldsymbol{u}) = 1/2(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T)$ is the velocity strain rate (i.e., the symmetric part of the velocity gradient), and p is the pressure, which plays the role of a Lagrange multiplier enforcing the continuity constraint $\nabla\cdot\boldsymbol{u} = 0$. The initial condition for the velocity is $\boldsymbol{u}(\boldsymbol{x}, t=0) = \boldsymbol{u}_0(\boldsymbol{x})$, and the boundary is partitioned as the non-overlapping union $\Gamma = \Gamma_D \bigcup \Gamma_N$ of the Dirichlet boundary Γ_D and the Neumann boundary Γ_N. The following boundary conditions are imposed:

$$\boldsymbol{u} = \boldsymbol{u}_D\,, \qquad \forall \boldsymbol{x} \text{ on } \Gamma_D\,, \tag{1c}$$

$$-(\chi_{\Gamma_N^-}\,\rho\boldsymbol{u}\otimes\boldsymbol{u} - \boldsymbol{\sigma}(\boldsymbol{u},p))\,\boldsymbol{n} = \boldsymbol{t}_N\,, \qquad \forall \boldsymbol{x} \text{ on } \Gamma_N\,, \tag{1d}$$

where $\boldsymbol{\sigma}(\boldsymbol{u},p) = -p\boldsymbol{I} + 2\,\mu\,\boldsymbol{\varepsilon}(\boldsymbol{u})$ and $\chi_{\Gamma_N^-}$ is the characteristic function of the set Γ_N^- (equal to one on Γ_N^- and zero otherwise). Here we also introduced the definitions of inflow and outflow boundaries for the Navier-Stokes equations, which must account for the fact that $\boldsymbol{u}$ is fully known only on Γ_D. We found effective the following definitions:

$$\Gamma_D^- = \{\boldsymbol{x}\in\Gamma_D \,|\, \boldsymbol{u}_D\cdot\boldsymbol{n} < 0\}\,, \tag{2a}$$

$$\Gamma_N^- = \{\boldsymbol{x}\in\Gamma_N \,|\, \boldsymbol{u}\cdot\boldsymbol{n} < 0\}\,, \tag{2b}$$

and $\Gamma_D^+ = \Gamma_D \setminus \Gamma_D^-$ and $\Gamma_N^+ = \Gamma_D \setminus \Gamma_N^-$.

3. The weighted shifted boundary approach

This section contains the abstract and theoretical core of the present manuscript. We start by describing the geometric and approximation settings associated with the Shifted Boundary Method in Sections 3.1–3.4, in terms of true/surrogate boundaries, distance maps, shift/extension operators, and approximation spaces. We derive the WSBM formulation in Section 3.5, which ends with a discussion of the algorithmic properties of the WSBM, in terms of exact preservation of states of hydrostatic equilibrium and the rate of convergence of the mass/momentum conservation errors. In particular, Section 3.5.7, presents an analysis of how the velocity initialization of freshly activated nodes can contribute to spurious pressure oscillations in time, in the limit of small time steps. We then describe the details of the implementation of the WSBM in Section 3.5.8. The discussion contained in these two sections is essential for the understanding of many of the computational results presented in Section 4.

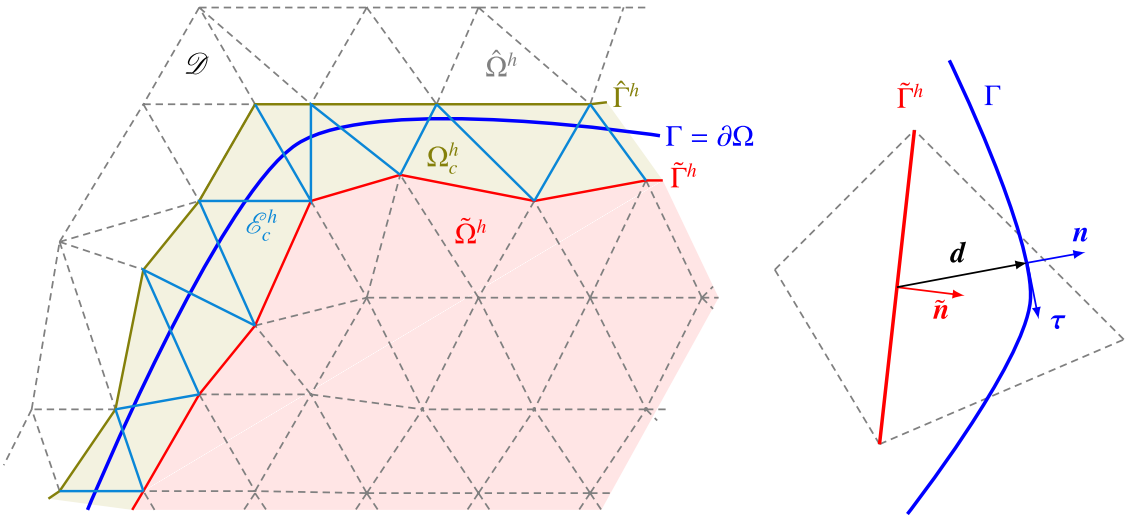
3.1. Geometric framework: the physical and surrogate domains

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 tessellations of $\mathcal{D}$. The set of all edges (in two dimensions) or faces (in three dimensions) of the tessellation $\mathcal{T}^h$ 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 each $T\in\mathcal{D}$. Then, we restrict each tessellation by selecting those elements that are strictly contained in the closure of Ω, denoted $\text{clos}(\Omega)$, i.e., we form

$$\tilde{\mathcal{T}}^h := \{T\in\mathcal{T}^h \,:\, T\subset\text{clos}(\Omega)\}\,. \tag{3}$$

Obviously, $\tilde{\mathcal{T}}^h$ is an admissible and shape-regular tessellation of $\tilde{\Omega}^h$. $\tilde{\mathcal{T}}^h$ identifies the inner surrogate domain (the red-shaded area in the sketch of Fig. 1a)

$$\tilde{\Omega}^h := \text{int}\left(\bigcup_{T\in\tilde{\mathcal{T}}^h} T\right)\subseteq\Omega\,, \tag{4}$$



(a) Surrogate interfaces $\tilde{\Gamma}^h$, $\hat{\Gamma}^h$, $\mathscr{E}_c^h$ and the active surrogate subdomain $\Omega^h = \tilde{\Omega}^h \cup \Omega_c^h$. The global domain is $\mathscr{D} = \hat{\Omega}^h \cup \Omega_c^h \cup \hat{\Omega}^h$.

(b) The distance vector $\boldsymbol{d}$, the true normal $\boldsymbol{n}$ and the true tangent $\boldsymbol{\tau}$.

Fig. 1. The surrogate domain, its boundary, and the distance vector $\boldsymbol{d}$.

with a corresponding set of edges/faces $\tilde{\mathcal{E}}^h = \{E \in \mathcal{E}^h : E \subset \tilde{\Omega}\}$ and inner surrogate boundary $\tilde{\Gamma}^h := \partial\tilde{\Omega}^h$ (the red polygonal line in Fig. 1a and b) and corresponding outward-oriented unit normal vector $\tilde{\boldsymbol{n}}$ (see again Fig. 1b, and note that $\tilde{\Gamma}^h \subset \tilde{\mathcal{E}}^h$). Note that $\tilde{\Gamma}^h$ may coincide with a portion of the boundary Γ, if that portion can be represented as a polygonal segment in two dimensions or a polyhedral surface in three dimensions and the computational grid is fitted to it. We also define $\tilde{\mathcal{E}}_i^h = \tilde{\mathcal{E}}^h \setminus \tilde{\Gamma}^h$. Analogously, we can define the outer tessellation and the outer inactive domain:

$$\hat{\mathcal{T}}^h := \{T \in \mathcal{T}^h : T \subset \mathrm{clos}(\mathcal{D} \setminus \Omega)\}, \qquad \hat{\Omega}^h := \mathrm{int}\left(\bigcup_{T \in \hat{\mathcal{T}}^h} T\right) \subseteq \mathcal{D} \setminus \Omega, \tag{5}$$

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

$$\mathcal{T}_c^h := \{T \in \mathcal{T}^h : T \cap \Gamma \neq \emptyset\}, \qquad \Omega_c^h := \mathrm{int}\left(\bigcup_{T \in \mathcal{T}_c^h} T\right), \tag{6}$$

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 (marked in cyan (light blue) in Fig. 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

$$\tilde{\boldsymbol{M}}^h : \tilde{\Gamma}^h \to \Gamma, \tag{7a}$$

$$\tilde{\boldsymbol{x}} \mapsto \boldsymbol{x}, \tag{7b}$$

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

$$\boldsymbol{d}_{\tilde{\boldsymbol{M}}^h}(\tilde{\boldsymbol{x}}) = \boldsymbol{x} - \tilde{\boldsymbol{x}} = [\,\tilde{\boldsymbol{M}}^h - \boldsymbol{I}\,](\tilde{\boldsymbol{x}}), \tag{8}$$

where $\boldsymbol{I}$ is the identity map. Equivalently, we can find a map $\hat{\boldsymbol{M}}^h : \hat{\Gamma}^h \to \Gamma$, defined by means of the distance $\boldsymbol{d}_{\hat{\boldsymbol{M}}^h}(\hat{\boldsymbol{x}}) = \boldsymbol{x} - \hat{\boldsymbol{x}}$, from a point on the outer surrogate, $\hat{\boldsymbol{x}} \in \hat{\Gamma}^h$, to its closest-point projection on the true boundary, $\boldsymbol{x} \in \Gamma$. Finally, it is also possible to define a map $\boldsymbol{M}_c^h : E \in \mathcal{E}_c^h \to \Gamma$ and relative distance vector $\boldsymbol{d}_{\boldsymbol{M}_c^h}$. 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 $\tilde{\boldsymbol{M}}^h$, $\hat{\boldsymbol{M}}^h$ and $\boldsymbol{M}_c^h$, and will simply write "$\boldsymbol{M}^h$." Similarly, we will omit the subscripts of $\boldsymbol{d}_{\tilde{\boldsymbol{M}}^h}$, $\boldsymbol{d}_{\hat{\boldsymbol{M}}^h}$ and $\boldsymbol{d}_{\boldsymbol{M}_c^h}$, and write "$\boldsymbol{d}$."

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 smooth portion of Γ, the closest-point projection algorithm will pick $\boldsymbol{d}$ in the direction of the true normal $\boldsymbol{n}$, that is $\boldsymbol{d} = \|\boldsymbol{d}\|\,\boldsymbol{n}$. In our case, we calculate $\boldsymbol{M}^h$ and $\boldsymbol{d}$ using tools developed in [121,122] and references therein. We refer the reader to the recent analysis in [103], for the general case of general domains with corners and edges.

Remark 2. Other strategies in the definition of $\boldsymbol{M}^h$ are possible, among which a level set description of the true boundary, in which $\boldsymbol{d}$ is defined by means of a distance function and its gradient. Most importantly, the methods that we are about to introduce do not depend on how $\boldsymbol{M}^h$ is constructed.

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

Through the map $\boldsymbol{M}^h$, it is possible to define an extension $\mathbb{E}$ operator on $\tilde{\Gamma}^h$ of functions initially defined only on Γ. For example, the unit normal vector $\boldsymbol{n}$ to the boundary Γ can be extended to the boundary $\tilde{\Gamma}^h$:

$$\mathbb{E}\boldsymbol{n}(\tilde{\boldsymbol{x}}) \equiv \boldsymbol{n}(\boldsymbol{M}^h(\tilde{\boldsymbol{x}}))\,. \tag{9a}$$

In what follows, with the purpose of simplifying the notation, we will omit the symbol $\mathbb{E}$, whenever this does not cause confusion. Therefore when we write $\boldsymbol{n}(\tilde{\boldsymbol{x}})$ we actually mean $\mathbb{E}\boldsymbol{n}(\tilde{\boldsymbol{x}})$. 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 $\mathcal{E}_c^h$). Considering, without loss of generality, a Dirichlet boundary condition, the following Taylor expansion formula centered at $\tilde{\boldsymbol{x}} \in \tilde{\Gamma}^h$ holds for $\boldsymbol{x} = \boldsymbol{M}^h(\tilde{\boldsymbol{x}}) \in \Gamma$:

$$\boldsymbol{S}_{\boldsymbol{d}}u(\tilde{\boldsymbol{x}}) \;:=\; u(\tilde{\boldsymbol{x}}) + \nabla u(\tilde{\boldsymbol{x}})\cdot\boldsymbol{d}(\tilde{\boldsymbol{x}}) \;=\; \mathbb{E}u_D(\tilde{\boldsymbol{x}}) + O\big(\|\boldsymbol{d}(\tilde{\boldsymbol{x}})\|^2\big)\,, \tag{10}$$

where $u_D(\boldsymbol{x})$ is the value of the Dirichlet boundary condition on u and $\boldsymbol{S}_{\boldsymbol{d}}u(\tilde{\boldsymbol{x}})$ is a shift operator. This formula 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.

3.4. Infinite dimensional function spaces and integrals

Let $\omega \in \mathcal{T}^h$ be a subset of the tessellation $\mathcal{T}^h$, that is the union of some elements that belong to $\mathcal{T}^h$. 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

$$\|v\|_{H^m(\omega)} = \left(\|v\|^2_{L^2(\omega)} + \sum_{k=1}^{m} \|l(\omega)^k\,\boldsymbol{D}^k v\|^2_{L^2(\omega)}\right)^{1/2}, \tag{11}$$

where $\boldsymbol{D}^k$ is the kth-order spatial derivative operator and $l(A) = (\mathrm{meas}(A))^{1/n_d}$ 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 $|v|_{k;\omega} = \|\boldsymbol{D}^k v\|_{0;\omega} = \|\boldsymbol{D}^k v\|_{L^2(\omega)}$. In the sequel, $(v,w)_\omega = \int_\omega v\,w$ denotes the L^2 inner product on $\omega \in \mathcal{T}^h$ and $\langle v,w\rangle_\zeta = \int_\zeta v\,w$ denotes the L^2-inner product on the subset ζ of $\mathcal{E}^h$.

3.5. A volume-weighted shifted boundary method for flows with moving boundaries

In what follows, we propose a technique that extends to moving boundaries the original Shifted Boundary Method (SBM) introduced in [94,95]. In contrast to the original developments in [94,95], integration is not restricted to the surrogate subdomain $\tilde{\Omega}^h$ but instead extends over the active computational domain $\Omega^h := \tilde{\Omega}^h \cup \Omega_c^h$. Similar to the original SBM, the integration of the variational equations is not performed over partially cut cells, with a considerable simplification in the implementation, when compared to cutFEM approaches.

3.5.1. Rationale

The total volume of a fluid governed by the incompressible Navier-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 [23,24,26]). These pressure oscillations are particularly problematic since they induce large spurious forces and only occur in time, while the pressure field remains smooth in space.

A problematic aspect that emerges immediately in the application to moving boundary problems of the standard shifted boundary framework [94,95] is the fact that the surrogate boundary $\tilde{\Omega}^h$ is not constant over time. Specifically, as Γ sweeps through Ω, the domain $\tilde{\Omega}^h$ experiences discrete changes due to the activation/deactivation of elements in computations, and these changes induce the mentioned spurious pressure oscillations in time. 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. Because of this feature, we named this scheme the Weighted Shifted Boundary Method (WSBM). The net result is a Petrov-Galerkin method, in which the test function space differs from the trial function space due to the geometric weighting just mentioned.

3.5.2. Variational formulation of the Navier-Stokes equations

For the moment, we consider the case of moderate Reynolds numbers and we do not introduce turbulence models, which can be added explicitly as detailed in [95] for the Spalart-Allmaras model, or implicitly as in the variational multiscale model proposed in [123,124] (see also Appendix A). In the derivation of the computational method that follows, we will make the following:

Assumption 1. The Neumann (i.e, traction) boundary is body-fitted, that is $\tilde{\Gamma}_N^h = \Gamma_N^h$ (where Γ_N^h a discrete manifold that interpolates Γ_N), and only Dirichlet boundary conditions are immersed/shifted.

Remark 3. Assumption 1 is certainly a simplification, but is also the common practice in immersed methods for computational fluid dynamics, since Neumann conditions are typically inflow and outflow conditions. Adapting the formulation to be described next to pressure boundary conditions is relatively easy, taking inspiration from the formulation for moving free surfaces described in [115]. As shown in [95], and more precisely in [97], general Neumann conditions can also be treated with the shifted boundary framework. These strategies could be useful for adiabatic walls in thermal Navier-Stokes formulations, although simpler approaches could also be adopted in this case, taking inspiration from [109].

Now, 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

$$\alpha_T = \frac{\text{meas}(T \cap \Omega)}{\text{meas}(T)} \ . \tag{12}$$

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 $\alpha(\boldsymbol{x})$ 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 $\boldsymbol{V}^h(\Omega^h)$ and $Q^h(\Omega^h)$, for the velocity and pressures, respectively. Namely:

$$\boldsymbol{V}^h(\Omega^h) = \left\{ \boldsymbol{w} \in (C(\Omega^h))^{n_d} \ : \ \boldsymbol{w}_{|T} \in \left(\mathcal{P}^1(T)\right)^{n_d} \ , \ T \in \tilde{\mathcal{T}}^h \cup \mathcal{T}_c^h \right\} , \tag{13a}$$

$$Q^h(\Omega^h) = \left\{ q \in C(\Omega^h) \ : \ q_{|T} \in \mathcal{P}^1(T) \ , \ T \in \tilde{\mathcal{T}}^h \cup \mathcal{T}_c^h \right\} , \tag{13b}$$

where $\mathcal{P}^1(T)$ is the set of piecewise linear functions over the element T, and $C(\Omega^h)$ and $(C(\Omega^h))^{n_d}$ are the sets of scalar- and vector-valued continuous functions over the active domain $\Omega^h = \tilde{\Omega}^h \cup \Omega_c^h$. $\boldsymbol{V}^h(\Omega^h)$ 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. The α-weighted test spaces,

$$\boldsymbol{V}_\alpha^h(\Omega^h) = \left\{ \boldsymbol{w}_\alpha \in (L^2(\Omega^h))^{n_d} \ : \ \boldsymbol{w}_\alpha = \alpha \boldsymbol{w} \ , \ \boldsymbol{w} \in \boldsymbol{V}^h(\Omega^h) \, , \ \alpha \in \mathcal{P}^0(\tilde{\mathcal{T}}^h \cup \mathcal{T}_c^h) \right\} , \tag{14a}$$

$$Q_\alpha^h(\Omega^h) = \left\{ q_\alpha \in L^2(\Omega^h) \ : \ q_\alpha = \alpha q \ , \ q \in Q^h(\Omega^h) \, , \ \alpha \in \mathcal{P}^0(\tilde{\mathcal{T}}^h \cup \mathcal{T}_c^h) \right\} , \tag{14b}$$

are discontinuous by construction. Multiplying the governing Eq. (1a)-(1b) by $\boldsymbol{w}_\alpha$ and q_α and integrating by parts over a generic element $T \subset \Omega^h$ (with E the generic edge on the element boundary ∂T), we have

$$\begin{aligned}
0 &= (\, \boldsymbol{w}_\alpha \, , \, \rho \, (\boldsymbol{u}_{,t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla \cdot (p\boldsymbol{I} - 2\,\mu\,\boldsymbol{\varepsilon}(\boldsymbol{u})) - \boldsymbol{b}) \,)_{\Omega^h} - (q_\alpha \, , \, \nabla \cdot \boldsymbol{u})_{\Omega^h} \\
&= (\, \boldsymbol{w}_\alpha \, , \, \rho \, (\boldsymbol{u}_{,t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \boldsymbol{b}) \,)_{\Omega^h} + (\boldsymbol{\varepsilon}(\boldsymbol{w}_\alpha) \, , \, -p\boldsymbol{I} + 2\,\mu\,\boldsymbol{\varepsilon}(\boldsymbol{u}))_{\Omega^h} - (q_\alpha \, , \, \nabla \cdot \boldsymbol{u})_{\Omega^h} \\
&\quad - \sum_{T \in \tilde{\mathcal{T}}^h \cup \mathcal{T}_c^h} \sum_{E \subset \partial T} \langle \boldsymbol{w}_\alpha \, , \, (-p\boldsymbol{I} + 2\,\mu\,\boldsymbol{\varepsilon}(\boldsymbol{u}))\boldsymbol{n}_E \rangle_E \\
&= (\, \boldsymbol{w}_\alpha \, , \, \rho \, (\boldsymbol{u}_{,t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \boldsymbol{b}) \,)_{\Omega^h} - (\alpha \, \nabla \cdot \boldsymbol{w} \, , \, p)_{\Omega^h} + (\alpha \, \boldsymbol{\varepsilon}(\boldsymbol{w}) \, , \, 2\,\mu\,\boldsymbol{\varepsilon}(\boldsymbol{u}))_{\Omega^h} - (\alpha \, q \, , \, \nabla \cdot \boldsymbol{u})_{\Omega^h} \\
&\quad - \langle \{\{\alpha \, \boldsymbol{w}\}\}_{1-\gamma} \, , \, [\![2\mu \, \boldsymbol{\varepsilon}(\boldsymbol{u}) - p\boldsymbol{I}]\!] \rangle_{\tilde{\mathcal{E}}_i^h \cup \mathcal{E}_{c;D}^h \cup \tilde{\Gamma}_D^h} - \langle [\![\alpha \, \boldsymbol{w}]\!] \, , \, \{\{ 2\mu \, \boldsymbol{\varepsilon}(\boldsymbol{u}) - p\boldsymbol{I} \}\}_\gamma \rangle_{\tilde{\mathcal{E}}_i^h \cup \mathcal{E}_{c;D}^h \cup \tilde{\Gamma}_D^h} \\
&\quad - \langle \alpha \, \boldsymbol{w} \otimes \hat{\boldsymbol{n}} \, , \, 2\mu \, \boldsymbol{\varepsilon}(\boldsymbol{u}) - p\boldsymbol{I} \rangle_{\hat{\Gamma}_D^h} - \langle \alpha \, \boldsymbol{w} \otimes \boldsymbol{n} \, , \, 2\mu \, \boldsymbol{\varepsilon}(\boldsymbol{u}) - p\boldsymbol{I} \rangle_{\Gamma_N^h} \ .
\end{aligned} \tag{15}$$

Here $\mathcal{E}_{c;D}^h$ and $\mathcal{E}_{c;D}^h$ are the sets of edges intersected by the (true) boundaries where Dirichlet boundary conditions are enforced. Analogously, $\tilde{\mathcal{E}}_i^h$ indicates the set of the interior edges of the elements belonging to $\tilde{\Omega}^h$. Note that we have introduced the average operator $\{\{\cdot\}\}_\gamma := \gamma^+(\cdot)^+ + \gamma^-(\cdot)^-$ 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 definition of the average operator is important for numerical stability as elucidated in [120].

We always choose the labels + and - so that $\alpha^+ \geq \alpha^-$ and we define the biased jump operator $[\alpha] := \alpha^+ - \alpha^- \geq 0$ on an edge E, as well as the unbiased jump operators $[\![\boldsymbol{\sigma}]\!] := \boldsymbol{\sigma}^+ \boldsymbol{n}_E^+ + \boldsymbol{\sigma}^- \boldsymbol{n}_E^-$, $[\![\boldsymbol{v}]\!] := \boldsymbol{v}^+ \cdot \boldsymbol{n}_E^+ + \boldsymbol{v}^- \cdot \boldsymbol{n}_E^-$, and $[\![\alpha]\!] := \alpha^+ \boldsymbol{n}_E^+ + \alpha^- \boldsymbol{n}_E^-$ for tensors, vectors, and scalars, respectively.

Next, we weakly enforce the balance of fluxes on the inter-element interfaces, that is $[\![2\mu \, \boldsymbol{\varepsilon}(\boldsymbol{u}) - p\boldsymbol{I}]\!] = \boldsymbol{0}$ on $\tilde{\mathcal{E}}_i^h \cup \mathcal{E}_{c;D}^h \cup \tilde{\Gamma}_D^h$, and the Neumann boundary conditions $-(\chi_{\Gamma_N^{h;-}} \rho \boldsymbol{u} \otimes \boldsymbol{u} + p\boldsymbol{I} - 2\mu \, \boldsymbol{\varepsilon}(\boldsymbol{u}))\boldsymbol{n} = \boldsymbol{t}_N$ on Γ_N^h. Observe that, $[\![\alpha \, \boldsymbol{w}]\!] = 0$ over $\tilde{\mathcal{E}}_i^h$, since $\alpha^\pm = 1$ and $\boldsymbol{w}$ 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 $\boldsymbol{w}$ beyond Ω^h, we have $[\![\alpha \, \boldsymbol{w}]\!] = \alpha \boldsymbol{w}$, (since $\boldsymbol{w}^+ = \boldsymbol{w}^-$ on $\hat{\Gamma}_D^h$ by the continuity of $\boldsymbol{w}$), and $\{\{ 2\mu \, \boldsymbol{\varepsilon}(\boldsymbol{u}) - p\boldsymbol{I} \}\}_\gamma = 2\mu \, \boldsymbol{\varepsilon}(\boldsymbol{u})^+ - p\boldsymbol{I}$ for any bounded extension of $\boldsymbol{\varepsilon}(\boldsymbol{u})$ 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, (15) becomes

$$\begin{aligned}
0 &= (\, \alpha \, \boldsymbol{w} \, , \, \rho \, (\boldsymbol{u}_{,t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \boldsymbol{b}) \,)_{\Omega^h} - (\alpha \, \nabla \cdot \boldsymbol{w} \, , \, p)_{\Omega^h} + (\alpha \, \boldsymbol{\varepsilon}(\boldsymbol{w}) \, , \, 2\,\mu\,\boldsymbol{\varepsilon}(\boldsymbol{u}))_{\Omega^h} - (\alpha \, q \, , \, \nabla \cdot \boldsymbol{u})_{\Omega^h} \\
&\quad - \langle [\![\alpha]\!] \otimes \boldsymbol{w} \, , \, \{\{ 2\mu \, \boldsymbol{\varepsilon}(\boldsymbol{u}) - p\boldsymbol{I} \}\}_\gamma \rangle_{\tilde{\mathcal{E}}_a^h} - \langle \alpha \hat{\boldsymbol{n}} \otimes \boldsymbol{w} \, , \, 2\mu \, \boldsymbol{\varepsilon}(\boldsymbol{u}) - p\boldsymbol{I} \rangle_{\hat{\Gamma}_D^h} - \langle \boldsymbol{w} \, , \, \boldsymbol{t}_N + \chi_{\Gamma_N^{h;-}} (\rho \boldsymbol{u} \otimes \boldsymbol{u}) \boldsymbol{n} \rangle_{\Gamma_N^h} \ .
\end{aligned} \tag{16}$$

Dirichlet boundary conditions can be enforced using a Nitsche-type approach, akin the interior penalty method. Observing that $[\![\alpha \, \boldsymbol{v}]\!] = \alpha \boldsymbol{v}$ on $\hat{\Gamma}_D^h$, we can augment (16) with appropriate boundary terms:

$$0 = (\, \alpha \, \boldsymbol{w} \, , \, \rho \, (\boldsymbol{u}_{,t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \boldsymbol{b}) \,)_{\Omega^h} - (\alpha \, \nabla \cdot \boldsymbol{w} \, , \, p)_{\Omega^h} + (\alpha \, \boldsymbol{\varepsilon}(\boldsymbol{w}) \, , \, 2\,\mu\,\boldsymbol{\varepsilon}(\boldsymbol{u}))_{\Omega^h} - (\alpha \, q \, , \, \nabla \cdot \boldsymbol{u})_{\Omega^h}$$

$$
\begin{aligned}
&- \langle [\![\alpha]\!] \otimes \boldsymbol{w}\,,\, 2\mu\,\{\{\boldsymbol{\varepsilon}(\boldsymbol{u})\}\}_\gamma - p\boldsymbol{I}\rangle_{\mathcal{E}^h_{a;D}} - \langle 2\mu\,\{\{\boldsymbol{\varepsilon}(\boldsymbol{w})\}\}_\gamma - q\boldsymbol{I}\,,\, [\![\alpha]\!] \otimes (\{\{\boldsymbol{S_d u}\}\}_\gamma - \boldsymbol{u}_D)\rangle_{\mathcal{E}^h_{a;D}} \\
&+ \langle \delta_1 \{\{2\,\mu/h\}\}_\gamma [\alpha] \{\{\boldsymbol{S_d w}\}\}_\gamma\,,\, \{\{\boldsymbol{S_d u}\}\}_\gamma - \boldsymbol{u}_D\rangle_{\mathcal{E}^h_{a;D}} - \langle \boldsymbol{w}\,,\, \boldsymbol{t}_N + \chi_{\Gamma^{h;-}_N}(\rho\boldsymbol{u}\otimes\boldsymbol{u})\boldsymbol{n}\rangle_{\Gamma^h_N}\,,
\end{aligned}
\tag{17}
$$

where we have used the fact that $[\![\alpha]\!] = \alpha^+\boldsymbol{n}^+_E + \alpha^-\boldsymbol{n}^-_E = [\alpha]\boldsymbol{n}^+_E$, since $\boldsymbol{n}^-_E = -\boldsymbol{n}^+_E$. The previous derivations lead to the following variational form:

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

$$
\begin{aligned}
0 = &(\alpha\,\boldsymbol{w}\,,\, \rho\,(\boldsymbol{u}_{,t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} - \boldsymbol{b}))_{\Omega^h} + (\alpha\,\boldsymbol{\varepsilon}(\boldsymbol{w}), 2\mu\,\boldsymbol{\varepsilon}(\boldsymbol{u}))_{\Omega^h} - (\alpha\,\nabla\cdot\boldsymbol{w}, p)_{\Omega^h} - (\alpha\,q, \nabla\cdot\boldsymbol{u})_{\Omega^h} \\
&- \langle \boldsymbol{w}\,,\, \boldsymbol{t}_N + \chi_{\Gamma^{h;-}_N}(\rho\boldsymbol{u}\otimes\boldsymbol{u})\boldsymbol{n}\rangle_{\Gamma^h_N} - \langle [\![\alpha]\!] \otimes \boldsymbol{w}\,,\, 2\mu\,\{\{\boldsymbol{\varepsilon}(\boldsymbol{u})\}\}_\gamma - p\boldsymbol{I}\rangle_{\mathcal{E}^h_{a;D}} \\
&- \langle 2\mu\,\{\{\boldsymbol{\varepsilon}(\boldsymbol{w})\}\}_\gamma - q\boldsymbol{I}\,,\, [\![\alpha]\!] \otimes (\{\{\boldsymbol{S_d u}\}\}_\gamma - \boldsymbol{u}_D)\rangle_{\mathcal{E}^h_{a;D}} + \langle \delta_1 \{\{2\,\mu/h\}\}_\gamma [\alpha] \{\{\boldsymbol{S_d w}\}\}_\gamma\,,\, \{\{\boldsymbol{S_d u}\}\}_\gamma - \boldsymbol{u}_D\rangle_{\mathcal{E}^h_{a;D}} \\
&+ \langle \delta_{\boldsymbol{u}} [\![\nabla\boldsymbol{w}]\!], [\![\nabla\boldsymbol{u}]\!]\rangle_{\tilde{\mathcal{E}}^h_{a;D}} + \langle \delta_p [\![\nabla q]\!], [\![\nabla p]\!]\rangle_{\tilde{\mathcal{E}}^h_{a;D}}\,.
\end{aligned}
\tag{18}
$$

The proposed formulation is complemented by the SUPG/variational multiscale stabilization described in Appendix A. The last two terms in Eq. (18) are the so-called ghost-penalty terms [35] 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 $\mathcal{T}^h_c$. This phenomenon is in a way analogous to the small cut cell problem [79], although we do not explicitly integrate the equations over cut elements or cutting boundaries. $\delta_{\boldsymbol{u}}$ and δ_p are scaling parameters and are defined with some analogy to [46], namely,

$$
\delta_{\boldsymbol{u}} = \alpha_{\boldsymbol{u}}\mu h^*\,, \qquad \delta_p = \frac{\alpha_p}{\mu} h^{*3}\,, \qquad h^* = \frac{2h^+h^-}{h^+ + h^-}\,.
\tag{19}
$$

In particular, we take $\alpha_{\boldsymbol{u}} = 0.1$ and $\alpha_p = .0001$. These definitions take inspiration from [46] and the refinements of these ideas presented in [120]. In contrast to what is recommended in [46], we decided not to scale $\delta_{\boldsymbol{u}}$ and δ_p with the flow velocity $\boldsymbol{u}$ or the time-step Δt, since these choices produced lack of convergence of the nonlinear Newton iteration in the case of three-dimensional tests. Also note that in most of the three-dimensional computations presented in Section 4.4, the element Péclet numbers are close to unity, and the effect of the velocity scaling on $\delta_{\boldsymbol{u}}$ and δ_p is fairly small. Furthermore, the time steps in those computations are chosen so that the (convective) CFL number is close to unity, and also in this case the dependency of $\delta_{\boldsymbol{u}}$ and δ_p on the time step would have a small effect.

Remark 4. Atallah et al. [120] have developed an analysis of numerical stability and convergence for the WSBM for the Poisson problem and the Stokes flow problem. The computational results in [116] demonstrated that the condition number of the WSBM is unaffected by small cuts present in the computational grid, again for the case of Stokes flow and for the same definition of the ghost penalty (19) used here.

Remark 5 (Smooth force variation at newly activated nodes). The definition of the average $\{\{\cdot\}\}_\gamma$ in the variational form (18) is engineered to produce smooth variations of the nodal forces in time when new nodes and elements are activated in the computational domain. The case of the Navier-Stokes equations is not very different from the simpler case of the Stokes flow problem, which was analyzed in [116] with computational experiments. Furthermore, it was mathematically proved in [120] that this specific definition of average should be preferred to the arithmetic average in order to ensure a stable and convergent method for the Poisson and Stokes problems.

3.5.3. Euler-Lagrange equations

Observe the following element-by-element integration-by-parts result, where $\boldsymbol{\sigma} = -p\boldsymbol{I} + 2\,\mu\,\boldsymbol{\varepsilon}(\boldsymbol{u})$:

$$
\begin{aligned}
(\alpha\,\boldsymbol{\varepsilon}(\boldsymbol{w}), \boldsymbol{\sigma})_{\Omega^h} &= (\boldsymbol{w}_\alpha, \nabla\cdot\boldsymbol{\sigma})_{\Omega^h} + \langle \boldsymbol{w}\,,\, [\![\alpha\,\boldsymbol{\sigma}]\!]\rangle_{\tilde{\mathcal{E}}^h_{a;D}\cup\tilde{\mathcal{E}}^h_i} + \langle \boldsymbol{w}_\alpha\,,\, \boldsymbol{\sigma}\hat{\boldsymbol{n}}\rangle_{\Gamma^h_D} \\
&= (\alpha\,\boldsymbol{w}, \nabla\cdot\boldsymbol{\sigma})_{\Omega^h} + \langle \boldsymbol{w}\otimes[\![\alpha]\!],\, \{\{\boldsymbol{\sigma}\}\}_\gamma\rangle_{\tilde{\mathcal{E}}^h_{a;D}\cup\tilde{\mathcal{E}}^h_i} + \langle \{\{\alpha\}\}_{1-\gamma}\,\boldsymbol{w}\,,\, [\![\boldsymbol{\sigma}]\!]\rangle_{\tilde{\mathcal{E}}^h_{a;D}\cup\tilde{\mathcal{E}}^h_i} + \langle \boldsymbol{w}_\alpha\,,\, \boldsymbol{\sigma}\hat{\boldsymbol{n}}\rangle_{\Gamma^h_D}\,.
\end{aligned}
\tag{20}
$$

Then, integrating by parts the terms $(\alpha\,\boldsymbol{\varepsilon}(\boldsymbol{w}), 2\mu\,\boldsymbol{\varepsilon}(\boldsymbol{u}))_{\Omega^h}$ and $(\alpha\,\nabla\cdot\boldsymbol{w}, p)_{\Omega^h}$ in (18), we obtain

$$
\begin{aligned}
0 = &\,(\alpha\,\boldsymbol{w}\,,\, \rho\,\boldsymbol{u}_{,t} + \rho\,\boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p - \nabla\cdot(2\,\mu\,\boldsymbol{\varepsilon}(\boldsymbol{u})) - \rho\,\boldsymbol{b})_{\Omega^h} - (\alpha\,q, \nabla\cdot\boldsymbol{u})_{\Omega^h} \\
&- \langle \boldsymbol{w}, (\rho\boldsymbol{u}\otimes\boldsymbol{u}\,\chi_{\Gamma^{h;-}_N} + p\boldsymbol{I} - 2\,\mu\,\boldsymbol{\varepsilon}(\boldsymbol{u}))\,\boldsymbol{n} + \boldsymbol{t}_N\rangle_{\Gamma^h_N} \\
&+ \langle \{\{\alpha\}\}_{1-\gamma}\,\boldsymbol{w}\,,\, [\![-p\boldsymbol{I} + 2\,\mu\,\varepsilon(\boldsymbol{u})]\!]\rangle_{\tilde{\mathcal{E}}^h_{a;D}\cup\tilde{\mathcal{E}}^h_i} \\
&+ \langle (q\boldsymbol{I} - 2\,\mu\,\{\{\boldsymbol{\varepsilon}(\boldsymbol{w})\}\}_\gamma)[\![\alpha]\!] + \delta_1\,\{\{\mu/h\}\}_\gamma\,[\alpha]\boldsymbol{w}\,,\, \{\{\boldsymbol{u} + (\nabla\boldsymbol{u})\boldsymbol{d} - \boldsymbol{u}_D\}\}_\gamma\rangle_{\tilde{\mathcal{E}}^h_{a;D}} \\
&+ \langle \alpha(q\hat{\boldsymbol{n}} - 2\,\mu\,\boldsymbol{\varepsilon}(\boldsymbol{w})\hat{\boldsymbol{n}}) + \delta_1\,\mu/h\,\alpha\,\boldsymbol{w}\,,\, \boldsymbol{u} + (\nabla\boldsymbol{u})\boldsymbol{d} - \boldsymbol{u}_D\rangle_{\Gamma^h_D} \\
&+ \langle \delta_{\boldsymbol{u}} [\![\nabla\boldsymbol{w}]\!], [\![\nabla\boldsymbol{u}]\!]\rangle_{\tilde{\mathcal{E}}^h_{a;D}} + \langle \delta_p [\![\nabla q]\!], [\![\nabla p]\!]\rangle_{\tilde{\mathcal{E}}^h_{a;D}}\,,
\end{aligned}
\tag{21}
$$

which are the Euler-Lagrange equations associated with the WSBM variational formulation. The first row of (21) enforces the Navier-Stokes equations on the interior of the active surrogate subdomain Ω^h; the second row enforces the Neumann boundary condition; the third row enforces the balance of stress on all interior element edges; and the fourth and fifth rows enforce the shifted Dirichlet boundary condition. The last row, associated with the ghost-penalty terms, implements a requirement of continuity of the normal gradients of the solution at element interfaces in $\tilde{\mathcal{E}}^h_{a;D}$.

3.5.4. Preservation of hydrostatic equilibrium

An important property to be satisfied by computational methods for fluids is the preservation of stationary hydrostatic equilibrium for any shape of the domain Ω^h. This solution state can be expressed as $\boldsymbol{u} = \boldsymbol{0}$ and $p = -\rho g z$, where g is the magnitude of the gravitational acceleration $\boldsymbol{g}$ and

z is the vertical coordinate. Under hydrostatic equilibrium, the pressure exactly balances the gravitational force, namely $\nabla p = \rho \boldsymbol{b} = \rho \boldsymbol{g} = -\rho g \boldsymbol{e}_z$. Algorithms that fail to satisfy this condition typically introduce spurious pressure oscillations in space, when the fluid is in an initial state of rest or when it comes to rest at the end of a simulation. Substituting $\boldsymbol{u} = \boldsymbol{0}$ in (18), we obtain

$$0 = -(\alpha\, \boldsymbol{w},\, \rho\, \boldsymbol{b})_{\Omega^h} - (\alpha\, \nabla \cdot \boldsymbol{w}, p)_{\Omega^h} + \langle [\![\alpha]\!] \otimes \boldsymbol{w},\, p\boldsymbol{I} \rangle_{\mathcal{E}^h_{a;D}} - \langle \boldsymbol{w},\, \boldsymbol{t}_N \rangle_{\Gamma^h_N} \tag{22}$$

with $\boldsymbol{t}_N = -p\boldsymbol{n} = \rho g z\, \boldsymbol{n}$. Using integration by parts on every element $T \in \tilde{\mathcal{T}}^h \cup \mathcal{T}^h_c$, it is not difficult to see that

$$(\alpha\, \boldsymbol{w}, \nabla p)_{\Omega^h} = -(\alpha\, \nabla \cdot \boldsymbol{w}, p)_{\Omega^h} + \langle [\![\alpha]\!] \otimes \boldsymbol{w},\, p\boldsymbol{I} \rangle_{\mathcal{E}^h_{a;D}} + \langle \boldsymbol{w},\, p\boldsymbol{n} \rangle_{\Gamma^h_N}\,, \tag{23}$$

so that comparing (22) and (23), we have

$$0 = (\alpha\, \boldsymbol{w}, -\rho\, \boldsymbol{b} + \nabla p)_{\Omega^h}\,, \tag{24}$$

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

3.5.5. 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 (18), in which we set $q = 1$ (i.e., $q_\alpha = \alpha$) and $\boldsymbol{w} = \boldsymbol{0}$ (i.e., $\boldsymbol{w}_\alpha = \boldsymbol{0}$):

$$0 = -(\alpha\,,\, \nabla \cdot \boldsymbol{u})_{\Omega^h} + \langle [\![\alpha]\!]\,,\, \boldsymbol{u} + \{\{\nabla \boldsymbol{u}\}\}_\gamma \boldsymbol{d} - \boldsymbol{u}_D \rangle_{\tilde{\mathcal{E}}^h_{a;D}} + \langle \alpha\, \hat{\boldsymbol{n}}\,,\, \boldsymbol{u} + (\nabla \boldsymbol{u})\boldsymbol{d} - \boldsymbol{u}_D \rangle_{\hat{\Gamma}^h_D}\,. \tag{25}$$

Splitting the integral $(\alpha, \nabla \cdot \boldsymbol{u})_{\Omega^h} = (1, \nabla \cdot \boldsymbol{u})_{\tilde{\Omega}^h} + (\alpha, \nabla \cdot \boldsymbol{u})_{\Omega^h_c}$, where $\alpha_{|\tilde{\Omega}^h} = 1$, and keeping in mind that $\boldsymbol{u}$ is continuous but α could be discontinuous across element boundaries, we obtain, by the Gauss divergence theorem:

$$\begin{aligned}
0 &= (1, \nabla \cdot \boldsymbol{u})_{\tilde{\Omega}^h} + (\alpha, \nabla \cdot \boldsymbol{u})_{\Omega^h_c} - \langle [\![\alpha]\!]\,,\, \boldsymbol{u} + \{\{\nabla \boldsymbol{u}\}\}_\gamma \boldsymbol{d} - \boldsymbol{u}_D \rangle_{\tilde{\mathcal{E}}^h_{a;D}} - \langle \alpha\, \hat{\boldsymbol{n}}\,,\, \boldsymbol{u} + (\nabla \boldsymbol{u})\boldsymbol{d} - \boldsymbol{u}_D \rangle_{\hat{\Gamma}^h_D} \\
&= \langle 1\,,\, \boldsymbol{u} \cdot \tilde{\boldsymbol{n}}^+ \rangle_{\tilde{\Gamma}^h_D} + \langle 1\,,\, \boldsymbol{u} \cdot \boldsymbol{n} \rangle_{\Gamma^h_N} + (\alpha, \nabla \cdot \boldsymbol{u})_{\Omega^h_c} \\
&\quad - \langle \underbrace{\alpha^+}_{=1} \tilde{\boldsymbol{n}}^+\,,\, \boldsymbol{u} + \{\{\nabla \boldsymbol{u}\}\}_\gamma \boldsymbol{d} - \boldsymbol{u}_D \rangle_{\tilde{\Gamma}^h_D} - \langle \alpha^- \tilde{\boldsymbol{n}}^-\,,\, \boldsymbol{u} + \{\{\nabla \boldsymbol{u}\}\}_\gamma \boldsymbol{d} - \boldsymbol{u}_D \rangle_{\tilde{\Gamma}^h_D} \\
&\quad - \langle [\![\alpha]\!]\,,\, \boldsymbol{u} + \{\{\nabla \boldsymbol{u}\}\}_\gamma \boldsymbol{d} - \boldsymbol{u}_D \rangle_{\mathcal{E}^h_{c;D}} - \langle \alpha\, \hat{\boldsymbol{n}}\,,\, \boldsymbol{u} + (\nabla \boldsymbol{u})\boldsymbol{d} - \boldsymbol{u}_D \rangle_{\hat{\Gamma}^h_D} \\
&= \langle 1\,,\, (\boldsymbol{u}_D - \{\{\nabla \boldsymbol{u}\}\}_\gamma \boldsymbol{d}) \cdot \tilde{\boldsymbol{n}}^+ \rangle_{\tilde{\Gamma}^h_D} + \langle 1\,,\, \boldsymbol{u} \cdot \boldsymbol{n} \rangle_{\Gamma^h_N} + e_{cons}(\boldsymbol{u}; \mathcal{T}^h_c)\,,
\end{aligned} \tag{26}$$

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

$$\begin{aligned}
e_{cons}(\boldsymbol{u}; \mathcal{T}^h_c) = (\alpha, \nabla \cdot \boldsymbol{u})_{\Omega^h_c} &- \langle \alpha^- \tilde{\boldsymbol{n}}^-\,,\, \{\{\boldsymbol{u} + (\nabla \boldsymbol{u})\boldsymbol{d} - \boldsymbol{u}_D\}\}_\gamma \rangle_{\tilde{\Gamma}^h_D} - \langle [\![\alpha]\!]\,,\, \{\{\boldsymbol{u} + (\nabla \boldsymbol{u})\boldsymbol{d} - \boldsymbol{u}_D\}\}_\gamma \rangle_{\mathcal{E}^h_{c;D}} \\
&- \langle \alpha\, \hat{\boldsymbol{n}}\,,\, \boldsymbol{u} + (\nabla \boldsymbol{u})\boldsymbol{d} - \boldsymbol{u}_D \rangle_{\hat{\Gamma}^h_D}\,.
\end{aligned} \tag{27}$$

Definition (27) is the error in the statement (26) of discrete global mass conservation for the WSBM over $\tilde{\Omega}^h$. Specifically, $(\boldsymbol{u}_D - \{\{\nabla \boldsymbol{u}\}\}_\gamma \boldsymbol{d}) \cdot \tilde{\boldsymbol{n}}^+$ is the net mass flux across the surrogate Dirichlet boundary $\tilde{\Gamma}^h_D$ and $\boldsymbol{u} \cdot \boldsymbol{n}$ is the net mass flux across the Neumann boundary Γ^h_N. As the computational grid is refined, the domain Ω^h_c shrinks and in particular $\text{meas}(\Omega^h_c) = O(h)$, since the thickness of Ω^h_c scales with h. Assuming that the WSBM discretization is numerically stable and convergent, we can expect $\nabla \cdot \boldsymbol{u} \to 0$ with some power of h. If we took as an example the SBM for the Stokes problem constructed with piecewise-linear, continuous approximation spaces [101,102], we could expect $\nabla \cdot \boldsymbol{u} = O(h)$. Hence, by Cauchy-Schwartz inequality, we estimate

$$(\alpha, \nabla \cdot \boldsymbol{u})_{\Omega^h_c} \le \underbrace{\|\alpha\|_{0;\Omega^h_c}}_{O(h^{1/2})} \underbrace{\|\nabla \cdot \boldsymbol{u}\|_{0;\Omega^h_c}}_{O(h^{3/2})} = O(h^2)\,. \tag{28}$$

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

$$\langle \alpha^- \tilde{\boldsymbol{n}}^-\,,\, \{\{\boldsymbol{u} + (\nabla \boldsymbol{u})\boldsymbol{d} - \boldsymbol{u}_D\}\}_\gamma \rangle_{\tilde{\Gamma}_D} + \langle [\![\alpha]\!]\,,\, \{\{\boldsymbol{u} + (\nabla \boldsymbol{u})\boldsymbol{d} - \boldsymbol{u}_D\}\}_\gamma \rangle_{\mathcal{E}^h_{c;D}} + \langle \alpha\, \hat{\boldsymbol{n}}\,,\, \boldsymbol{u} + (\nabla \boldsymbol{u})\boldsymbol{d} - \boldsymbol{u}_D \rangle_{\hat{\Gamma}^h_D} = O(h^2)\,. \tag{29}$$

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

$$\langle 1\,,\, (\boldsymbol{u}_D - \{\{\nabla \boldsymbol{u}\}\}_\gamma \boldsymbol{d}) \cdot \tilde{\boldsymbol{n}}^+ \rangle_{\tilde{\Gamma}^h_D} + \langle 1\,,\, \boldsymbol{u} \cdot \boldsymbol{n} \rangle_{\Gamma^h_N} = -e_{cons}(\boldsymbol{u}; \mathcal{T}^h_c) = O(h^2)\,. \tag{30}$$

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 $e_{cons}(\boldsymbol{u}; \mathcal{T}^h_c) = O(h^{p+1})$.

Remark 6. 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, which introduces a geometric error of order $O(h^2)$ in the representation of the volume of active fluid.

3.5.6. An estimate of the global momentum conservation error

Next, we estimate the momentum conservation error of the WSBM. Setting $q = -\rho \boldsymbol{u} \cdot \boldsymbol{e}_i$ and $\boldsymbol{w} = \boldsymbol{e}_i$ in (18) and recalling that $\alpha_{|\tilde{\Omega}^h} = 1$, we get:

$$
\begin{aligned}
0 = &\, (\, \alpha\, \boldsymbol{e}_i \,,\, \rho\, (\boldsymbol{u}_{,t} - \boldsymbol{b}) \,)_{\Omega^h} + \underbrace{(\, \alpha \,,\, (\rho\, \boldsymbol{u} \cdot \nabla \boldsymbol{u}) \cdot \boldsymbol{e}_i \,+\, \rho\, (\boldsymbol{u} \cdot \boldsymbol{e}_i)\, \nabla \cdot \boldsymbol{u})_{\Omega^h}}_{=\, (\, \alpha \boldsymbol{e}_i \,,\, \nabla \cdot (\rho\, \boldsymbol{u} \otimes \boldsymbol{u}) \,)_{\Omega^h}} \\
&+ \langle \boldsymbol{w} \,,\, \boldsymbol{t}_N - \chi_{\Gamma_N^{h;-}}\, (\rho \boldsymbol{u} \otimes \boldsymbol{u}) \boldsymbol{n} \rangle_{\Gamma_N^h} - \langle [\![\alpha]\!] \,,\, (\{\{ 2 \mu\, \boldsymbol{\varepsilon}(\boldsymbol{u}) \}\}_\gamma - p \boldsymbol{I})\, \boldsymbol{e}_i \rangle_{\mathcal{E}^h_{a;D}} \\
&- \langle \rho \boldsymbol{u} \cdot \boldsymbol{e}_i [\![\alpha]\!] \,,\, \{\{ \boldsymbol{S}_{\boldsymbol{d}} \boldsymbol{u} \}\}_\gamma - \boldsymbol{u}_D \rangle_{\mathcal{E}^h_{a;D}} + \langle \delta_1 \{\{ 2\, \mu / h \}\}_\gamma [\alpha] \boldsymbol{e}_i \,,\, \{\{ \boldsymbol{S}_{\boldsymbol{d}} \boldsymbol{u} \}\}_\gamma - \boldsymbol{u}_D \rangle_{\mathcal{E}^h_{a;D}} \\
= &\, (\, \boldsymbol{e}_i \,,\, \rho\, (\boldsymbol{u}_{,t} - \boldsymbol{b}) \,)_{\tilde{\Omega}^h} + (\, \boldsymbol{e}_i \,,\, (\nabla \cdot (\rho\, \boldsymbol{u} \otimes \boldsymbol{u})))_{\tilde{\Omega}^h} + \langle\, \boldsymbol{e}_i \,,\, \boldsymbol{t}_N - \chi_{\Gamma_N^{h;-}}\, (\rho \boldsymbol{u} \otimes \boldsymbol{u}) \boldsymbol{n} \rangle_{\Gamma_N^h} \\
&+ (\, \alpha\, \boldsymbol{e}_i \,,\, \rho\, \boldsymbol{u}_{,t} + \nabla \cdot (\rho\, \boldsymbol{u} \otimes \boldsymbol{u}) - \rho\, \boldsymbol{b} \,)_{\Omega_c^h} \\
&- \langle [\![\alpha]\!] \,,\, \{\{ 2 \mu\, \boldsymbol{\varepsilon}(\boldsymbol{u}) - p \boldsymbol{I} \}\}_\gamma\, \boldsymbol{e}_i \rangle_{\mathcal{E}^h_{a;D}} - \langle \{\{ \alpha \}\}_{1-\gamma}\, \boldsymbol{e}_i \,,\, \underbrace{[\![2 \mu\, \boldsymbol{\varepsilon}(\boldsymbol{u}) - p \boldsymbol{I}]\!]}_{=\, 0 \text{ (weakly)}} \rangle_{\tilde{\mathcal{E}}^h_{a;D}} \\
&- \langle \rho \boldsymbol{u} \cdot \boldsymbol{e}_i [\![\alpha]\!] \,,\, \{\{ \boldsymbol{S}_{\boldsymbol{d}} \boldsymbol{u} \}\}_\gamma - \boldsymbol{u}_D \rangle_{\mathcal{E}^h_{a;D}} + \langle \delta_1 \{\{ 2\, \mu / h \}\}_\gamma [\alpha] \boldsymbol{e}_i \,,\, \{\{ \boldsymbol{S}_{\boldsymbol{d}} \boldsymbol{u} \}\}_\gamma - \boldsymbol{u}_D \rangle_{\mathcal{E}^h_{a;D}} \,.
\end{aligned}
\tag{31}
$$

Observe that the term $\langle \{\{ \alpha \}\}_{1-\gamma}\, \boldsymbol{e}_i \,,\, [\![2 \mu\, \boldsymbol{\varepsilon}(\boldsymbol{u}) - p \boldsymbol{I}]\!] \rangle_{\tilde{\mathcal{E}}^h_{a;D}}$ evaluates to zero, since we have required that $[\![2 \mu\, \boldsymbol{\varepsilon}(\boldsymbol{u}) - p \boldsymbol{I}]\!] = 0$ in a weak sense, when constructing the variational form (18). An application of the Gauss divergence theorem yields:

$$
\begin{aligned}
0 = &\, (\, \boldsymbol{e}_i \,,\, \rho\, (\boldsymbol{u}_{,t} - \boldsymbol{b}) \,)_{\tilde{\Omega}^h} + \langle \boldsymbol{e}_i, (\rho\, \boldsymbol{u} \otimes \boldsymbol{u}) \tilde{\boldsymbol{n}}^+ \rangle_{\tilde{\Gamma}_D^h} + \langle\, \boldsymbol{e}_i \,,\, \boldsymbol{t}_N + \chi_{\Gamma_N^{h;+}}\, (\rho \boldsymbol{u} \otimes \boldsymbol{u}) \boldsymbol{n} \rangle_{\Gamma_N^h} \\
&+ (\, \alpha\, \boldsymbol{e}_i \,,\, \rho\, \boldsymbol{u}_{,t} + \nabla \cdot (\rho\, \boldsymbol{u} \otimes \boldsymbol{u}) - \rho\, \boldsymbol{b} \,)_{\Omega_c^h} - \langle 1 \,,\, [\![\alpha\, (2 \mu\, \boldsymbol{\varepsilon}(\boldsymbol{u}) - p \boldsymbol{I})\, \boldsymbol{e}_i]\!] \rangle_{\tilde{\mathcal{E}}^h_{a;D}} - \langle \alpha \hat{\boldsymbol{n}} \,,\, (2 \mu\, \boldsymbol{\varepsilon}(\boldsymbol{u}) - p \boldsymbol{I})\, \boldsymbol{e}_i \rangle_{\hat{\Gamma}^h_{a;D}} \\
&- \langle \rho \boldsymbol{u} \cdot \boldsymbol{e}_i [\![\alpha]\!] \,,\, \{\{ \boldsymbol{S}_{\boldsymbol{d}} \boldsymbol{u} \}\}_\gamma - \boldsymbol{u}_D \rangle_{\mathcal{E}^h_{a;D}} + \langle \delta_1 \{\{ 2\, \mu / h \}\}_\gamma [\alpha] \boldsymbol{e}_i \,,\, \{\{ \boldsymbol{S}_{\boldsymbol{d}} \boldsymbol{u} \}\}_\gamma - \boldsymbol{u}_D \rangle_{\mathcal{E}^h_{a;D}} \\
= &\, (\, \boldsymbol{e}_i \,,\, \rho\, (\boldsymbol{u}_{,t} - \boldsymbol{b}) \,)_{\tilde{\Omega}^h} + \langle \boldsymbol{e}_i, (\rho\, \boldsymbol{u} \otimes \boldsymbol{u} + p \boldsymbol{I} - 2 \mu\, \boldsymbol{\varepsilon}(\boldsymbol{u})) \tilde{\boldsymbol{n}}^+ \rangle_{\tilde{\Gamma}_D^h} + \langle\, \boldsymbol{e}_i \,,\, \boldsymbol{t}_N + \chi_{\Gamma_N^{h;+}}\, (\rho \boldsymbol{u} \otimes \boldsymbol{u}) \boldsymbol{n} \rangle_{\Gamma_N^h} + e_{mom}(\boldsymbol{u}; \mathcal{T}_c^h) \,,
\end{aligned}
\tag{32}
$$

where

$$
\begin{aligned}
e_{mom}(\boldsymbol{u}; \mathcal{T}_c^h) = &\, (\, \alpha\, \boldsymbol{e}_i \,,\, \rho\, \boldsymbol{u}_{,t} + \nabla \cdot (\rho\, \boldsymbol{u} \otimes \boldsymbol{u} + p \boldsymbol{I} - 2 \mu\, \boldsymbol{\varepsilon}(\boldsymbol{u})) - \rho\, \boldsymbol{b} \,)_{\Omega_c^h} \\
&- \langle \rho \boldsymbol{u} \cdot \boldsymbol{e}_i [\![\alpha]\!] \,,\, \{\{ \boldsymbol{S}_{\boldsymbol{d}} \boldsymbol{u} \}\}_\gamma - \boldsymbol{u}_D \rangle_{\mathcal{E}^h_{a;D}} + \langle \delta_1 \{\{ 2\, \mu / h \}\}_\gamma [\alpha] \boldsymbol{e}_i \,,\, \{\{ \boldsymbol{S}_{\boldsymbol{d}} \boldsymbol{u} \}\}_\gamma - \boldsymbol{u}_D \rangle_{\mathcal{E}^h_{a;D}}
\end{aligned}
\tag{33}
$$

is the momentum conservation error. Eq. (32) is a statement of (approximate) momentum conservation. In particular, $(\rho\, \boldsymbol{u} \otimes \boldsymbol{u} + p \boldsymbol{I} - 2 \mu\, \boldsymbol{\varepsilon}(\boldsymbol{u})) \tilde{\boldsymbol{n}}^+$ is the net momentum flux across $\tilde{\Gamma}_D^h$ and $\boldsymbol{t}_N + \chi_{\Gamma_N^{h;+}}\, (\rho \boldsymbol{u} \otimes \boldsymbol{u}) \boldsymbol{n}$ is the net momentum flux across Γ_N^h, both of which contribute with the distributed force $\rho \boldsymbol{b}$ to a total change of momentum in the surrogate domain $\tilde{\Omega}^h$. To quantify the momentum conservation error $e_{mom}(\boldsymbol{u}; \mathcal{T}_c^h)$, as before, we observe that $\text{meas}(\Omega_c^h) = O(h)$ and, for a converging discrete solution,

$$
\rho\, \boldsymbol{u}_{,t} + \nabla \cdot (\rho\, \boldsymbol{u} \otimes \boldsymbol{u} + p \boldsymbol{I} - 2 \mu\, \boldsymbol{\varepsilon}(\boldsymbol{u})) - \rho\, \boldsymbol{b} = O(h) \,,
\tag{34}
$$

because of the presence of the derivatives of the solution in time and space. Then

$$
(\, \alpha\, \boldsymbol{e}_i \,,\, \rho\, \boldsymbol{u}_{,t} + \nabla \cdot (\rho\, \boldsymbol{u} \otimes \boldsymbol{u} + p \boldsymbol{I} - 2 \mu\, \boldsymbol{\varepsilon}(\boldsymbol{u})) - \rho\, \boldsymbol{b} \,)_{\Omega_c^h} = O(h^2) \,.
\tag{35}
$$

Now in principle we have that, because of the Taylor expansion residuals,

$$
-\langle \rho \boldsymbol{u} \cdot \boldsymbol{e}_i [\![\alpha]\!] \,,\, \{\{ \boldsymbol{S}_{\boldsymbol{d}} \boldsymbol{u} \}\}_\gamma - \boldsymbol{u}_D \rangle_{\mathcal{E}^h_{a;D}} + \langle \delta_1 \{\{ 2\, \mu / h \}\}_\gamma [\alpha] \boldsymbol{e}_i \,,\, \{\{ \boldsymbol{S}_{\boldsymbol{d}} \boldsymbol{u} \}\}_\gamma - \boldsymbol{u}_D \rangle_{\mathcal{E}^h_{a;D}} = O(h) \,,
\tag{36}
$$

which implies

$$
e_{mom}(\boldsymbol{u}; \mathcal{T}_c^h) \;=\; O(h) \,.
\tag{37}
$$

For higher-order interpolations, the same argument would lead to the estimate $e_{cons}(\boldsymbol{u}; \mathcal{T}_c^h) = O(h^p)$, for polynomial finite element spaces of order p.

3.5.7. Pressure oscillations in time and velocity initialization of newly activated nodes

The governing equations are integrated using a BDF2 time integrator, compatible with adaptive time stepping. In the case of a constant time step computation, the BDF2 discretization of the time derivative reads:

$$
\boldsymbol{u}_{,t} \approx \frac{3 \boldsymbol{u}^{n+1} - 4 \boldsymbol{u}^n + \boldsymbol{u}^{n-1}}{2\, \Delta t} \,.
\tag{38}
$$

Hence, every node of the grid that is freshly activated at time t_{n+1} by the motion of the boundary must have valid states at t_n and t_{n-1}. This in turns implies a CFL-type condition that is based on the motion of the boundary:

$$
\Delta t \;\leq\; CFL_\Gamma \min_{T \in \mathcal{T}_c^h} \frac{D_T}{\| \boldsymbol{v}_\Gamma \|} \,,
\tag{39}
$$

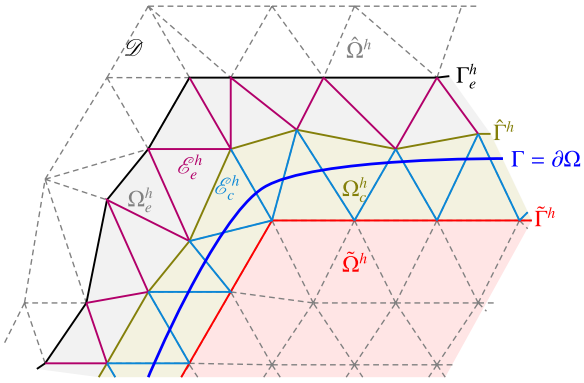


Fig. 2. The background domain $\mathcal{D} = \tilde{\Omega}^h \cup \Omega_c^h \cup \hat{\Omega}^h$, active surrogate subdomain $\Omega^h = \tilde{\Omega}^h \cup \Omega_c^h$, extension domain $\Omega_e^h \subset \hat{\Omega}^h$, and the surrogate interfaces $\tilde{\Gamma}^h$, $\hat{\Gamma}^h$, $\mathcal{E}_c^h$.

where $\boldsymbol{v}_\Gamma$ is the boundary velocity and D_T is the diameter of the circle inscribed in the element $T \in \mathcal{T}_c^h$ (i.e., an element intersected by Γ) and $CFL_\Gamma < 1/2$. In plain words, we require that the motion of the boundary sweeps an element over at least two time steps, so that t_n and t_{n-1} are always well defined. Of course, if more layers of nodes were initialized beyond $\hat{\Gamma}$, it is possible to seamlessly run for $CFL_\Gamma > 1/2$. In some numerical tests, we will also explore the case in which $1/2 \leq CFL_\Gamma < 1$, to evaluate the potential for instabilities.

Condition (39) can be combined with a classical advective CFL condition of the form

$$\Delta t \;\leq\; CFL \min_{T \in \tilde{\mathcal{T}}^h \cup \mathcal{T}_c^h} \frac{D_T}{\|\boldsymbol{u}\|} \,, \tag{40}$$

which is computed over all the elements in the active domain Ω^h, using the velocity field $\boldsymbol{u}$, with $CFL = 1$. Because the time integrator used in this work is implicit, this second CFL condition is not necessary, although beneficial for the convergence of the Newton-Raphson iterative solver. Taking the smallest time step resulting from conditions (39) and (40) yields the final time step.

While most existing references [14,23,24,26] focus on properly enforcing that the total volume of active fluid is conserved, the velocity initialization has in fact an important effect on spurious pressure oscillations in time, which may plague immersed methods. Looking at Fig. 2, nodes on the boundary Γ_e^h 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, some nodes on Γ_e^h may be activated, and become part of the boundary $\hat{\Gamma}^h$ over the next time step. Hence, the velocity field needs to be initialized in nodes in Γ_e^h, so that a time derivative can be computed. Now consider the limit case in which the time-step goes to zero. Only a few terms will dominate the variational form. As $\Delta t \to 0$, the time derivative term could become increasingly large, since it scales like the inverse of Δt. Other terms in (18) 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 inertia, 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 $\Delta t \to 0$, (18) reduces to:

$$\begin{aligned} 0 \approx\; & \frac{1}{2\,\Delta t}(\alpha\,\boldsymbol{w}\,,\,\rho\,(3\boldsymbol{u}^{n+1} - 4\boldsymbol{u}^{n} + \boldsymbol{u}^{n-1}))_{\Omega^h} - (\alpha\,\nabla\cdot\boldsymbol{w}, p^{n+1})_{\Omega^h} + \langle[\![\alpha]\!]\cdot\boldsymbol{w}\,,\,p^{n+1}\rangle_{\mathcal{E}^h_{a;D}} \\ & - (\alpha\,q, \nabla\cdot\boldsymbol{u}^{n+1})_{\Omega^h} + \langle\delta_p[\![\nabla q]\!], [\![\nabla p^{n+1}]\!]\rangle_{\tilde{\mathcal{E}}^h_{a;D}} \,. \end{aligned} \tag{41}$$

Now the extensions of $\boldsymbol{u}^n$ and $\boldsymbol{u}^{n-1}$ to nodes on Γ_e^h may not precisely respect the incompressibility condition over the elements shaded in gray in Fig. 2, that is the elements lying between $\hat{\Gamma}^h$ and Γ_e^h. For example, in all computational experiments, we initialize the velocity of the nodes in Γ_e^h with the velocity of the rigid body that is moving inside the fluid: a simple, effective, and commonly employed option. In these circumstances, the action of the pressure field p^{n+1} will attempt to re-establish the incompressibility of the velocity field at time t_{n+1} and, in particular, the pressure average over the active domain will adjust.

According to our analysis, even when the total volume of fluid is correctly accounted for, spurious pressure oscillations in time may arise due to improper initialization. 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. 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. Besides our earlier work on Stokes flow [116], we could not find a discussion on this topic in the existing literature. The work in [14] tends to emphasize the importance of accounting for the correct conserved volume, but there is no mention to the effect of solution initialization. A very recent and interesting work [119] is investigating the relationship between the LBB (inf-sup) stability condition and the pressure oscillations in time.

3.5.8. Algorithmic implementation

For completeness, we summarize in Algorithm 1 the overall solution procedure that implements the WSBM variational formulation (18). The WSBM is implemented here using a Newton solver for the coupled velocity-pressure system. The choice of a coupled Newton solver was due to its availability in our in-house computational suite and a pressure projection approach would be a better choice if computational efficiency and performance are the goals. The startup of the BDF2 time integrator is performed employing one BDF1 (backward Euler) step at the very beginning

of the computation. For all computations presented in the numerical tests, 3–4 iterations of the Newton solver are sufficient to drive the relative residual below the threshold of 10^{-10}.

Algorithm 1: Implementation of the WSBM for a BDF2 time integrator.

1: initialization ($t = t_0 = 0$, $n - 1 = 0$):
2: set the CFL number to a prescribed value (typically less than 0.5) and compute $\Delta t_1 = t_1 - t_0$
3: move the true boundary to its position at $t_n = t_1$
4: perform the geometric preprocessing at t_1 to evaluate $\tilde{\Omega}^h$, Ω_c^h, $\tilde{\Gamma}^h$, $\hat{\Gamma}^h$, $\mathcal{E}_c^h$, $\boldsymbol{d}$, etc.
5: set the initial velocity $\boldsymbol{u}_{n-1} = \boldsymbol{u}_0$ on $\tilde{\Omega}^h$
6: extrapolate the initial velocity $\boldsymbol{u}_{n-1}$ to nodes on Γ_e^h, to initialize the velocity over all Ω^h
7: solve for $\boldsymbol{u}_1$ and p_1 on $\Omega^h(t_1)$, using the BDF1 (backward Euler) integrator
8: compute the force coefficients $\boldsymbol{C}_f$ and torque coefficients $\boldsymbol{C}_t$ at t_1
9: end initialization
10: loop over time step $n \geq 1$ while $t_{n+1} < t_{final}$
11: extrapolate $\boldsymbol{u}_n$ to nodes on $\Gamma_e^h(t_n)$
12: compute $\Delta t_{n+1} = t_{n+1} - t_n$ using the prescribed CFL condition
13: if $t_{n+1} > t_{final}$ then set $\Delta t_{n+1} = t_{final} - t_n$ endif
14: move the true boundary to its position at t_{n+1}
15: perform the geometric preprocessing at t_{n+1} to evaluate $\tilde{\Omega}^h$, Ω_c^h, $\tilde{\Gamma}^h$, $\hat{\Gamma}^h$, $\mathcal{E}_c^h$, $\boldsymbol{d}$, etc.
16: note: $\boldsymbol{u}_{n-1}$ and $\boldsymbol{u}_n$ are available at all nodes of Ω^h, due to the extrapolation in stage 11
17: solve for $\boldsymbol{u}_{n+1}$ and p_{n+1} on $\Omega^h(t_{n+1})$, using the BDF2 integrator with $\boldsymbol{u}_{n-1}$ and $\boldsymbol{u}_n$, previously computed
18: compute the force coefficients $\boldsymbol{C}_f$ and torque coefficients $\boldsymbol{C}_t$ at t_{n+1}
19: end loop over n

The algebraic system of equations originating from the Newton iteration is solved using the Trilinos library [125]: the sparse linear algebra operations are handled by the Epetra package; the linear solver is a preconditioned GMRES using the AztecOO package; and a simple Additive Schwarz preconditioner is provided by the Ifpack package.

This algebraic solution strategy leverages general purpose algorithms and at the moment is not optimized for the W-SBM, a topic that will be the object of future development. However, the algebraic system of equations resulting from the spatial and temporal discretization has the same structure of the algebraic systems obtained from standard stabilized methods for incompressible Navier-Stokes equations. Therefore, we expect that specific state of the art solvers [126–128] could be directly applied to obtain optimal performance and scalability.

Additionally, we would like to highlight a few appealing features of the moving boundary tracking strategy, which computes the active and inactive nodes/elements. We utilize a mesh/geometry preprocessing that represents the moving boundary as a collection of linear segments in two dimensions of flat triangular faces in three dimensions (STL format). The geometric preprocessing computes $\tilde{\Omega}^h$, Ω_c^h, $\tilde{\Gamma}^h$, $\hat{\Gamma}^h$, $\mathcal{E}_c^h$, $\boldsymbol{d}$, etc., and relies on a modified version of the PhysBAM library[121,122]. The computation of the geometric parameters required by the WSBM is embarrassingly parallel, in the sense that no communication is needed between processors during the geometric preprocessing stage.

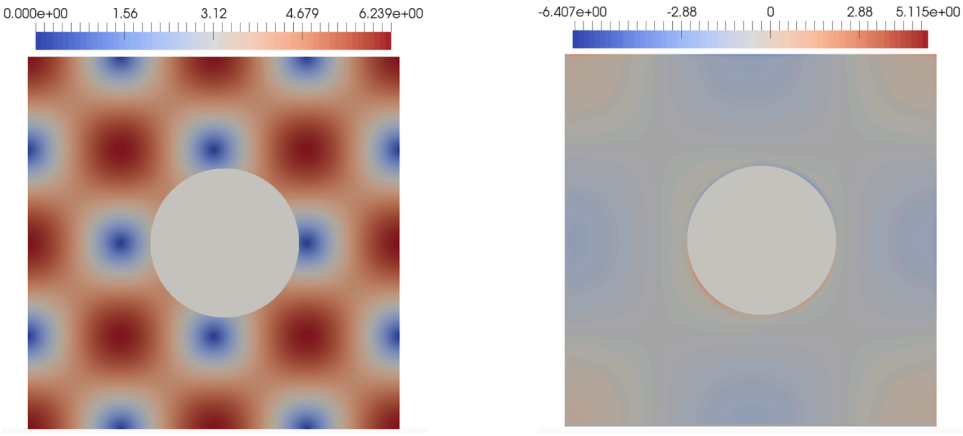
4. Numerical results

We present next a number of numerical tests to assess the performance of the proposed approach. A first series of tests is aimed at demonstrating the WSBM's numerical properties discussed in Sections 3.5.4–3.5.6, and its performance with regard to the pressure oscillations in time discussed in Section 3.5.7. To the authors' knowledge, no immersed/embedded/unfitted method of finite difference/volume/element type can completely avoid pressure oscillations in time, in the limit of small time steps. We will show however that the amplitude of pressure oscillations is relatively low for the WSBM. A second series of tests is aimed at showing the capabilities of the WSBM with moving shapes of very complex three-dimensional geometry. In this context, we will also show that the WSBM produces low levels of pressure oscillations, when the computational grids have resolution sufficient to capture the relevant flow physics. This second set of tests will clearly demonstrate that the WSBM is a viable approach to moving-boundary CFD with complex geometries rendered in non-CAD formats. For simplicity, as already indicated in Section 3.5.7, the velocity of the fluid is initialized with the velocity of the immersed body, for nodes that lie beyond the active domain Ω^h, and in particular in the extension boundary Γ_e^h. As already mentioned in Section 3.5.8, for all numerical tests in two and three dimensions, 3–4 iterations of the Newton solver were found sufficient to drive the relative residual below the threshold of 10^{-10}.

4.1. Two-dimensional convergence tests: forced Taylor-Green vortex with moving boundaries

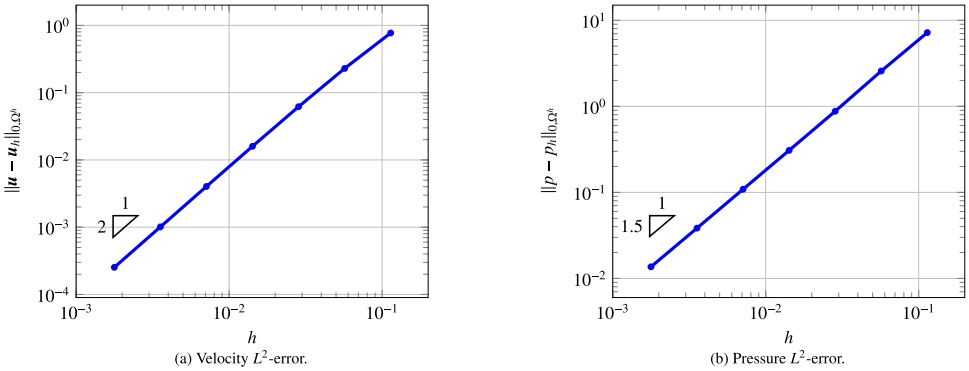
First, we examine the accuracy of the WSBM using the method of manufactured solutions. Given the square tank $[-0.5, 0.5] \times [-0.5, 0.5]\,\mathrm{m}^2$, we consider an internal moving boundary of two different shapes: a circular hole of radius 0.2 m and a square hole of side 0.25 m, both centered at the origin. The circular shape is an example of smooth boundary, while the square shape is an example of rough boundary. The center $\boldsymbol{x_c}$ of the immersed shapes moves according to the law

$$\boldsymbol{x_c}(t) = \left\{ \begin{array}{c} -X_0 \cos(2\pi f_0 t) \\ 0 \end{array} \right\} \mathrm{m}\,, \qquad \boldsymbol{u_c}(t) = \left\{ \begin{array}{c} U_0 \sin(2\pi f_0 t) \\ 0 \end{array} \right\} \mathrm{m/s}\,, \tag{42}$$



(a) Velocity. (b) Pressure.

Fig. 3. Convergence test (circular moving boundary): Velocity magnitude and pressure at $t = 1.6T$ for the mesh of size $h = 1/128$ m.



(a) Velocity L^2-error. (b) Pressure L^2-error.

Fig. 4. Convergence test (circular moving boundary): Convergence rates of the L^2-norm of the errors $||\boldsymbol{u} - \boldsymbol{u}_h||_{0,\Omega^h}$ and $||p - p_h||_{0,\Omega^h}$ for the manufactured solution (43). Numerical solutions are computed at time $t = 1.6T$.

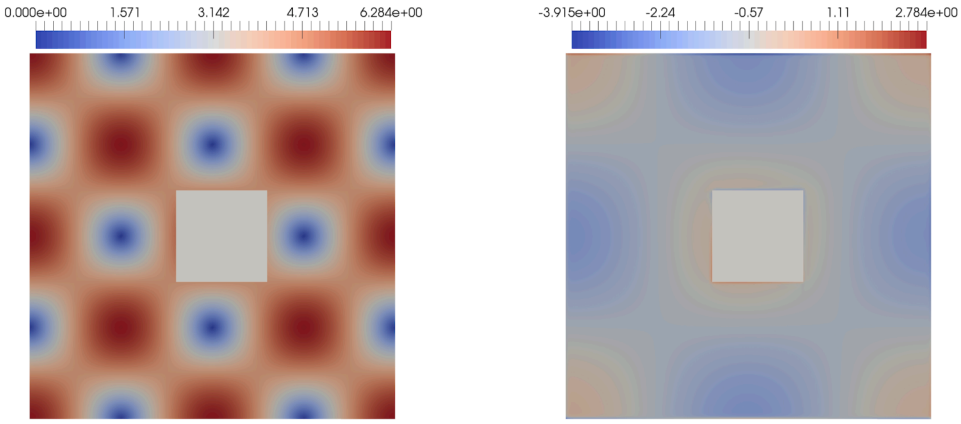
where $\boldsymbol{u_c}$ is the velocity of the shape center, $X_0 = 0.03125$ m, $U_0 = 2\pi f_0 X_0$ m/s, and $T = 1/f_0 = 1.0$ s is the period of the oscillation. The density is $\rho = 1\,\mathrm{kg/m^3}$ and we consider the following exact solution

$$\boldsymbol{u}(x,y) = \left\{ \begin{array}{c} 2\pi \sin(2\pi y)\sin(2\pi x)\sin(t) \\ 2\pi \cos(2\pi x)\cos(2\pi y)\sin(t) \end{array} \right\} \mathrm{m/s}\,, \tag{43a}$$

$$p(x,y) = (\cos(2\pi x)\cos(2\pi y) - 1)\sin(t)\,\mathrm{N/m^2}\,. \tag{43b}$$

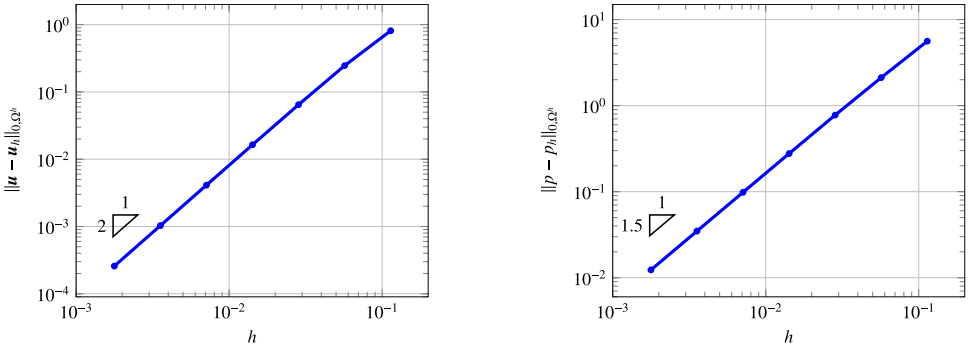
In particular, the forcing $\boldsymbol{b}(x,y)$ and boundary conditions are adjusted using the method of manufactured solutions. Observe that the solution's velocity field is divergence-free and, in fact, this flow can be considered as a forced version of the Taylor-Green vortex. As already mentioned, the velocity is initialized in the set of nodes in Γ_e^h using the velocity $\boldsymbol{u_c}$ of the immersed shapes given by (42). We apply no-slip (Dirichlet) boundary conditions at the top/bottom/hole boundaries and traction (Neumann) conditions ($\boldsymbol{\sigma n} = \boldsymbol{t_N}$) at the left and right boundaries. We use 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 such mesh resolutions were run at constant time step, with corresponding values $\Delta t = 0.28971$ s, 0.14486 s, 0.072428 s, 0.036214 s, 0.018107 s, 0.0090535 s, 0.0045268 s, so that time and space errors stay balanced. Figs. 3 and 5 show the velocity and pressure profiles obtained with the fine mesh ($h = 1/128$ m) at time $t = 1.6T$. The convergence rates of the velocity and pressure L^2-errors at time $t = 1.6T$ are shown in Figs. 4 and 6, and match the expected rates for stabilized methods on body-fitted meshes. Namely, the convergence rate of the velocity error is quadratic, the convergence rate of the pressure error is 1.5, and both fields remain smooth for each of the computational grids utilized.

Fig. 7 shows the scaling of condition numbers with the grid size h, for the circle and square geometries. In particular, the pictures show estimates of the average, minimum and maximum condition numbers for the raw, unpreconditioned algebraic system of equations over the course of simulations. It is easy to see that all these quantities scale like h^{-2}, as expected. This also seems to indicate that small-cut cells do not have a significant effect on the condition number scaling. Observe that the condition numbers of the circle case are at least an order of magnitude larger than the square case, but with smaller relative ranges between minimum and maximum values. The higher relative ranges of condition numbers for



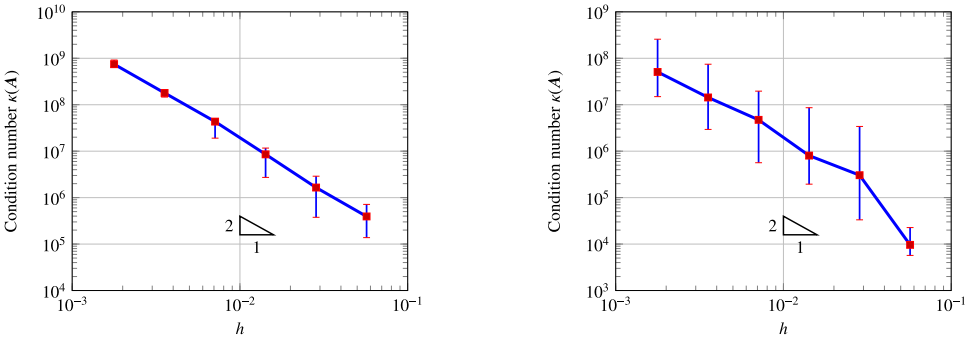
(a) Velocity. (b) Pressure.

Fig. 5. Convergence test (square moving boundary): Velocity magnitude and pressure at $t = 1.6T$ for the mesh of size $h = 1/128$ m.



(a) Velocity L^2-error. (b) Pressure L^2-error.

Fig. 6. Convergence test (square moving boundary): Convergence rates of the L^2-norm of the errors $||\boldsymbol{u} - \boldsymbol{u}_h||_{0,\Omega^h}$ and $||p - p_h||_{0,\Omega^h}$ for the manufactured solution (43). Numerical solutions are computed at time $t = 1.6T$.



(a) Oscillating circle (b) Oscillating square

Fig. 7. Convergence test with moving boundaries: Matrix condition numbers for the convergence tests discussed in Section 4.1. The average condition numbers over the time history of the computations are shown in blue, while the red error bars indicate the maximum and minimum values attained. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the square geometry could be attributed to the different flow patterns induced by the presence of sharp corners. We will elaborate further on this point in Sections 4.3.2 and 4.3.3.

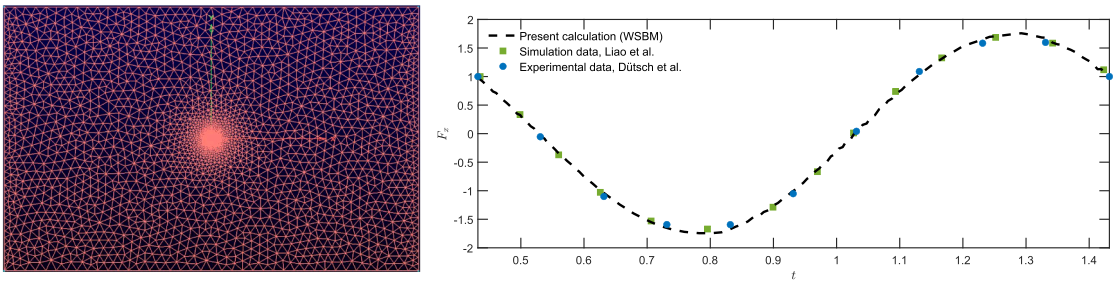


Fig. 8. Validation test: Oscillating cylinder in a fluid at rest [17,117]. Left: Zoomed view of the computational grid. Right: Comparison of the horizontal component of the force induced by the flow on the cylinder against the results of Liao et al. [17] and Dütsch et al. [117]. The simulation is performed at $Re = 100$ and $KC = 5$.

4.2. Validation against an experimental/numerical benchmark

Here, we validate the proposed WSBM against a test that was proposed by the authors of [117], who also performed numerical and experimental investigations. This test consist of solving the flow induced by the oscillations of a cylinder of diameter D in a fluid initially at rest. In particular, our numerical setup (computational domain/grid) is very similar to the subsequent computational studies in [17]. The computational domain is a rectangle of dimensions $55D \times 35D$, with a cylinder initially located at its center. The center $\boldsymbol{x}_c$ of the cylinder oscillates according to

$$\boldsymbol{x}_c = \left\{ \begin{array}{c} -\frac{\sin(2\pi f t)}{2\pi} \\ 0 \end{array} \right\} \mathrm{m/s} \, .$$

The computations are performed at a Reynolds number

$$Re = \frac{U_{\max} D}{\nu} = 100$$

and a Keulegan-Carpenter number

$$KC = \frac{U_{\max}}{f D} = 5 \, ,$$

where f is the frequency of oscillation and $U_{\max}$ is the maximum velocity of the center of the cylinder ($U_{\max} = f$ after a direct computation). A uniform grid of 80×80 square cells is used to discretize the region $[-D, D] \times [-D, D]$ and each square cell is further subdivided into two triangles. The grid is coarsened away from this square region, until it reaches a resolution such that 250 edges are found along each of the horizontal sides of the rectangular domain and 200 edges along the vertical sides. A zoomed view of the grid around the region $[-D, D] \times [-D, D]$ is shown in Fig. 8. The computation is performed with $CFL_\Gamma = 0.55$, as in [17]. Fig. 8 shows the horizontal force produced by the flow on the cylinder and includes comparisons with the experimental results in [117] and the numerical results in [17]. There is very good agreement of the WSBM results against both references.

4.3. Two-dimensional studies on the spurious pressure oscillations in time

The authors of [14] studied spurious oscillations in time of the pressure field by computing the pressure drag coefficient C_f of immersed moving bodies. In the case of the present method, we define C_f using information on the surrogate boundary $\tilde{\Gamma}^h$:

$$C_f = \frac{\int_\Gamma p \, (\boldsymbol{n} \cdot \boldsymbol{e}_{mb}) \, \mathrm{d}\Gamma}{(1/2) \, \rho \, \ell^3 \, f_0^2} \approx \frac{\int_{\tilde{\Gamma}^h} (p + \nabla p \cdot \boldsymbol{d}) \, (\boldsymbol{n} \cdot \boldsymbol{e}_{mb}) \, (\tilde{\boldsymbol{n}} \cdot \boldsymbol{n}) \, \mathrm{d}\tilde{\Gamma}^h}{(1/2) \, \rho \, \ell^3 \, f_0^2} \, , \tag{44}$$

where Γ is the boundary of the moving shape, $\tilde{\Gamma}^h$ is its surrogate counterpart, $\boldsymbol{e}_{mb}$ is the unit vector aligned with the direction of motion of the moving boundary, ℓ is a characteristic length of the problem, and $\tilde{\boldsymbol{n}} \cdot \boldsymbol{n}$ is a first order approximation of the signed area ratio $|\mathrm{d}\Gamma|/|\mathrm{d}\tilde{\Gamma}^h|$. The sign of $\tilde{\boldsymbol{n}} \cdot \boldsymbol{n}$ avoids to double count area contributions when boundaries have marked zig-zagging patterns [110,111]. We quantify oscillations with the metric

$$\| \Theta(C_f) \|_{\ell^\infty(\mathbb{T})} = \max_{t_n \in \mathbb{T}} |\Theta(C_f; t_n)| \, , \tag{45a}$$

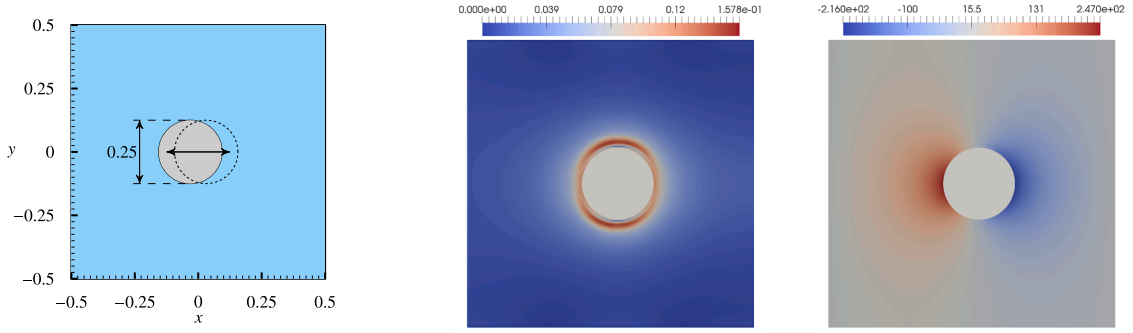
a measure of the maximum oscillation of C_f over the period $\mathbb{T}$ of oscillation of the moving shape(s), where

$$\Theta(C_f; t_n) = C_f(t_n) - \bar{C}_f(t_n) \tag{45b}$$

and $\bar{C}_f(t_n)$ is the output of a Butterworth high-pass filter applied to $C_f(t_n)$, with sampling frequency $f_{sampling} = 1/\Delta t_{\min}$, filter order 2, and cutoff frequency f_0 (i.e., the frequency of oscillation of the cylinder, normalized by the Nyquist frequency $f_{Nyquist} = f_{sampling}/2$). For a detailed description of the filtering procedure, see [116]. Data about the type and size of the grids utilized in computations are reported in Table 1. The computations were run at constant CFL_Γ, given by definition (39).

Table 1
Description of the triangular meshes used in the two-dimensional tests of Section 4.3. Each grid is obtained first creating a uniform Cartesian grid of squares with sides of length h, and further subdividing each square into two triangles.

Tests	Mesh size (h)	Number of elements
Oscillating cylinder in a fluid at rest	1/64 m	8192
	1/128 m	32,768
	1/192 m	73,728
	1/256 m	131,072
Oscillating cylinder/square in cross flow	0.1 m	46,072
	0.05 m	184,288
	0.025 m	737,152



(a) Setup. (b) Velocity magnitude. (c) Pressure.

Fig. 9. Oscillating cylinder in a fluid initially at rest. Left: Setup. Center/right: Velocity magnitude/pressure at time $t = 1.5T$, for a mesh with $D/h = 64$.

4.3.1. Oscillating cylinder in a fluid at rest

This benchmark was proposed by Seo and Mittal [14] 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 an incompressible fluid initially at rest. A sketch of the setup is shown in Fig. 9. The diameter of the cylinder is $D = 0.25$ m. The initial conditions are $\boldsymbol{v}(0) = \boldsymbol{v}_0 = 0\,\mathrm{m/s}$ and $p(0) = p_0 = 0\,\mathrm{N/m^2}$. 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 has initial coordinates $(-X_0, 0)$ and moves according to the equations, governing its position and velocity,

$$\boldsymbol{x_c}(t) = \left\{ \begin{array}{c} -X_0 \cos(2\pi f_0 t) \\ 0 \end{array} \right\} \mathrm{m}\,, \qquad \boldsymbol{u_c}(t) = \left\{ \begin{array}{c} U_0 \sin(2\pi f_0 t) \\ 0 \end{array} \right\} \mathrm{m/s}\,, \tag{46}$$

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 $\rho = 1600.0\,\mathrm{kg/m^3}$ and the dynamic viscosity of the fluid is $\mu = 1.0\,\mathrm{kg/(m\,s)}$. Hence we obtain a nominal Reynolds number $Re = \rho U_0 D/\mu = 78.5$. The computational meshes are obtained by subdividing the domain in squares of side h and then splitting each square into two sub-triangles.

Fig. 9 shows the velocity and pressure profiles at time $t = 1.5T$, which are smooth and consistent with the theoretical expectations. In fact, potential pressure oscillations are global in time and do not affect the spatial behavior. Fig. 10 presents a numerical comparison of the time histories of C_f, as the time step and grid size are changed. In the computations of the coefficient C_f, we take the length scale $\ell = D$, the diameter of the cylinder. The convergence plots at the bottom of the same figure show the behavior of oscillations as the time step and mesh size are reduced. It is clearly visible that the oscillations are proportional to $h^2/\Delta t$. This implies that the oscillations ($\|\Theta(C_f)\|_{\ell^\infty(\mathbb{T})}$) would scale as CFL_Γ^{-1}, by scaling arguments, as confirmed in the plots of Figs. 11, where we also see that oscillations appear more significant for $CFL_\Gamma = 0.05$, but even in this case they remain on the order of 2–3 % of the peak value of C_f, for the coarser grid. We show the behavior of the pressure oscillations as a function of CFL_Γ since often, in the practice, computations are run at constant CFL number, because of physical parametrizations of multi-physics problems, etc. We further investigate the performance of the proposed WSBM by quantifying in Fig. 12 the time averages of the mass conservation error e_{cons} and momentum conservation error e_{mom}, defined in (27) and (33). The mass conservation error converges as expected, while the momentum conservation error converges with rate 1.2, that is slightly faster than the theoretical estimate.

4.3.2. Oscillating cylinder in cross flow

We consider next the simulation of an oscillating circular cylinder in cross flow, a benchmark proposed in [42] in the context of the incompressible Navier-Stokes equations. A cylinder of diameter $D = 2.0$ m is immersed in the rectangular domain $[-10, 10] \times [-5, 5]\,\mathrm{m^2}$, as depicted in Fig. 13. The fluid density is $\rho = 50.0\,\mathrm{kg/m^3}$, and the dynamic viscosity is $\mu = 1.0\,\mathrm{Kg/(m\,s)}$. The initial position of the cylinder has coordinates $(0, X_0)$, where $X_0 = 0.25D = 0.5$ m. The cylinder translates rigidly, following a motion expressed in terms of the location and velocity of its

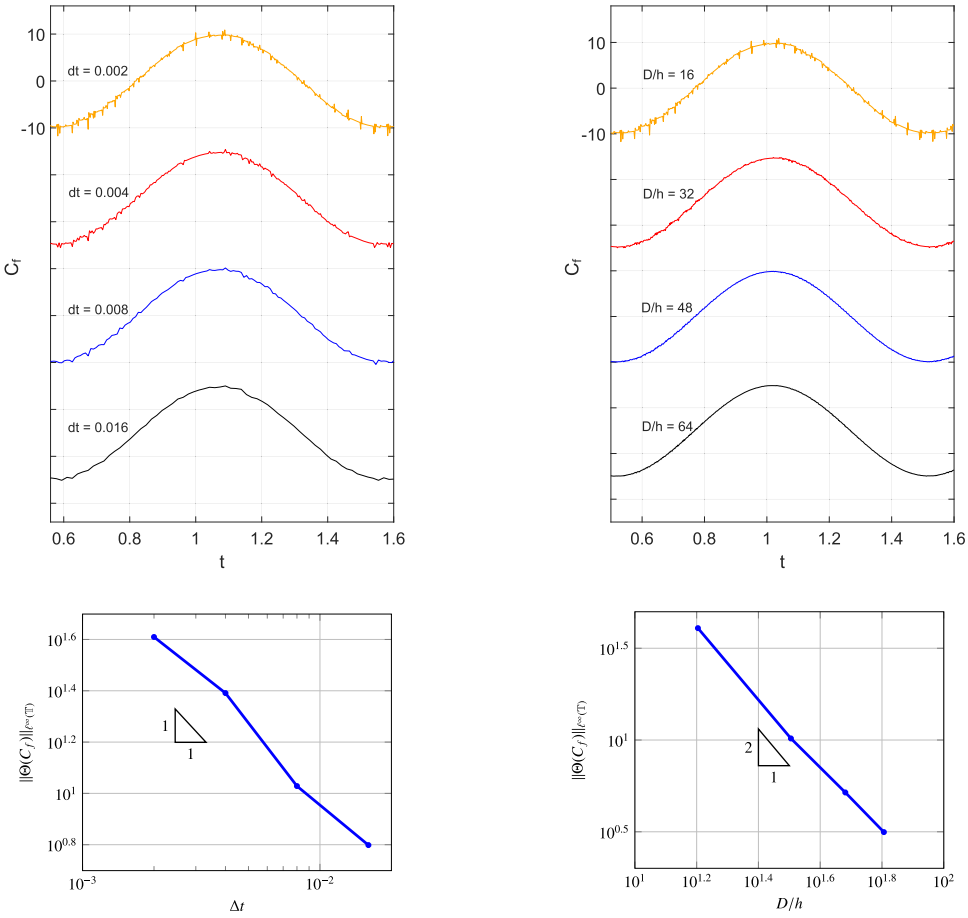
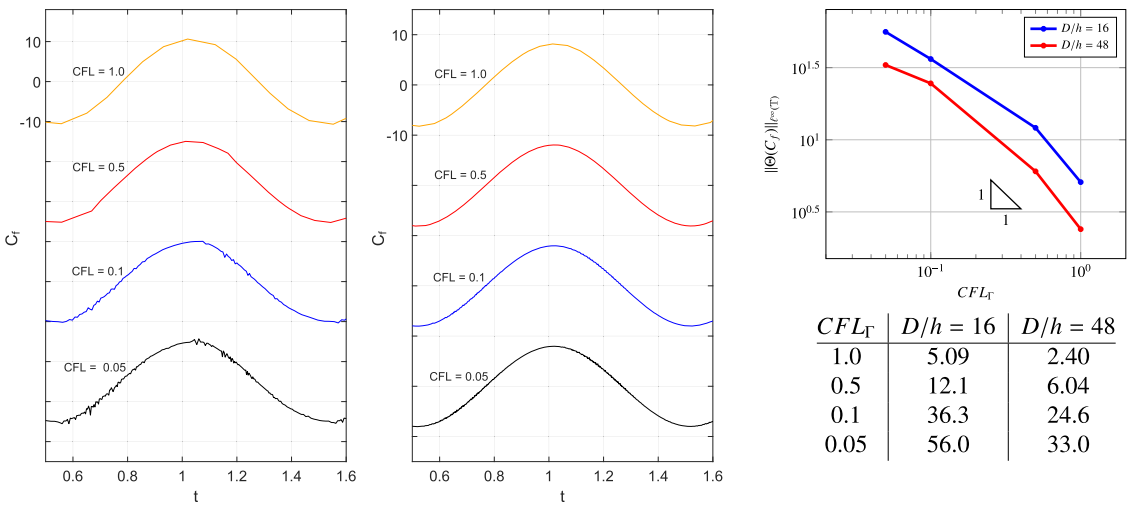


Fig. 10. Oscillating cylinder in a fluid at rest: C_f time histories (above) and convergence of the oscillation metric $\|\Theta(C_f)\|_{\ell^\infty(\mathbb{T})}$. Left: fixed grid size ($D/h = 16$) and varying time step Δt. Right: Fixed time step ($\Delta t = 0.002$) and varying grid sizes (D/h ratio). Note that the yellow curves on the top left and top right plots correspond to the same simulation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



CFL_Γ	$D/h = 16$	$D/h = 48$
1.0	5.09	2.40
0.5	12.1	6.04
0.1	36.3	24.6
0.05	56.0	33.0

Fig. 11. Oscillating cylinder in a fluid at rest: C_f time histories for various CFL_Γ numbers and two grid sizes: $D/h = 16$ (left) and $D/h = 48$ (middle). On the right, the variation of $\|\Theta(C_f)\|_{\ell^\infty(\mathbb{T})}$ as a function of CFL_Γ, for the grids of size $D/h = 16$ (blue) and $D/h = 48$ (red). Numerical values are shown on the table on the bottom-right corner. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

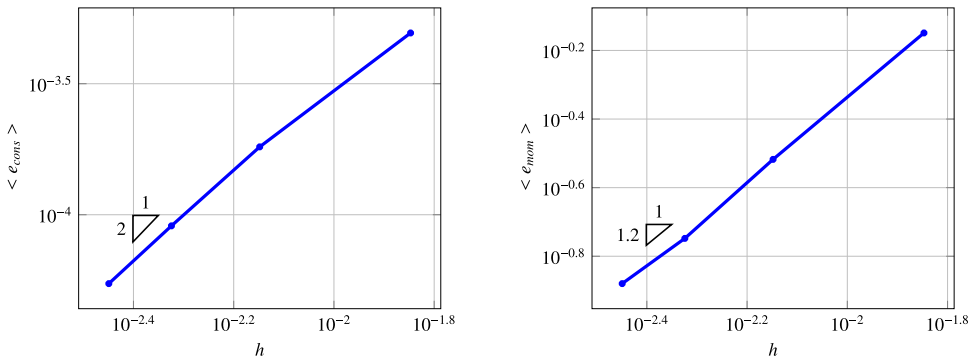


Fig. 12. Oscillating cylinder in a fluid at rest: Convergence under mesh refinement of the time-averages of the mass and momentum conservation errors. Left: Average mass conservation error $< e_{cons} >$. Right: Average momentum conservation error $< e_{mom} >$.

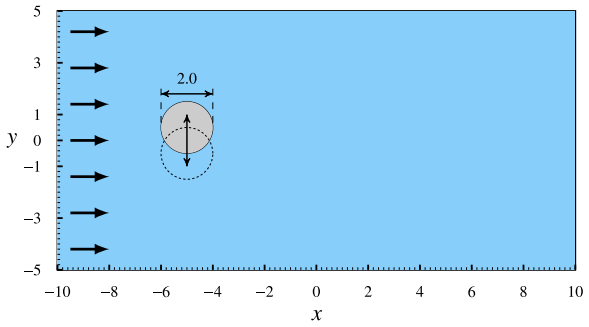
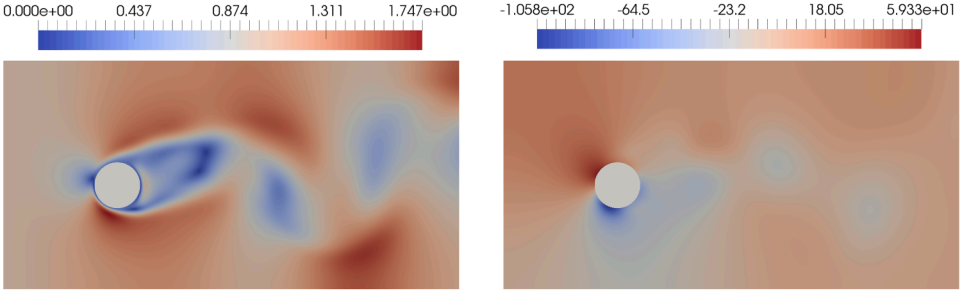


Fig. 13. Oscillating cylinder in cross flow: Setup. A vertically oscillating circular cylinder is subject to cross flow in a rectangular domain.



(a) Velocity magnitude at $t = 25.0$ s.

(b) Pressure at $t = 25.0$ s.

Fig. 14. Oscillating cylinder in cross flow: Velocity magnitude and pressure distribution at time $t = 25.0$ s.

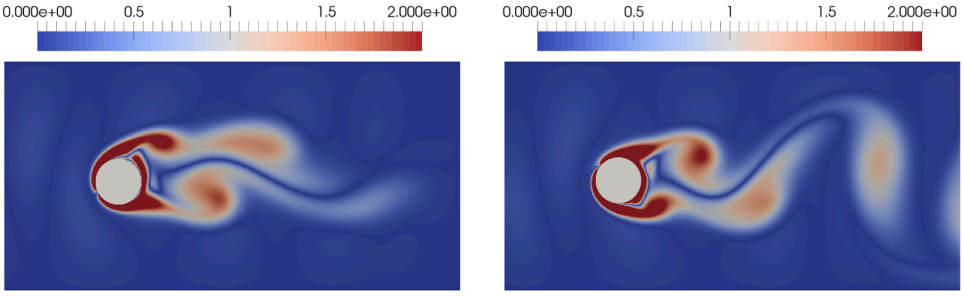
barycenter:

$$\boldsymbol{x}_c(t) = \left\{ \begin{array}{c} -5.0 \\ X_0 \cos(2\pi f_0 t) \end{array} \right\} \text{m}\,, \qquad \boldsymbol{u}_c(t) = \left\{ \begin{array}{c} 0 \\ -U_0 \sin(2\pi f_0 t) \end{array} \right\} \text{m/s}\,, \tag{47}$$

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 Reynolds number associated with the oscillatory motion is $Re_0 = \rho U_0 D/\mu = 70.3$.

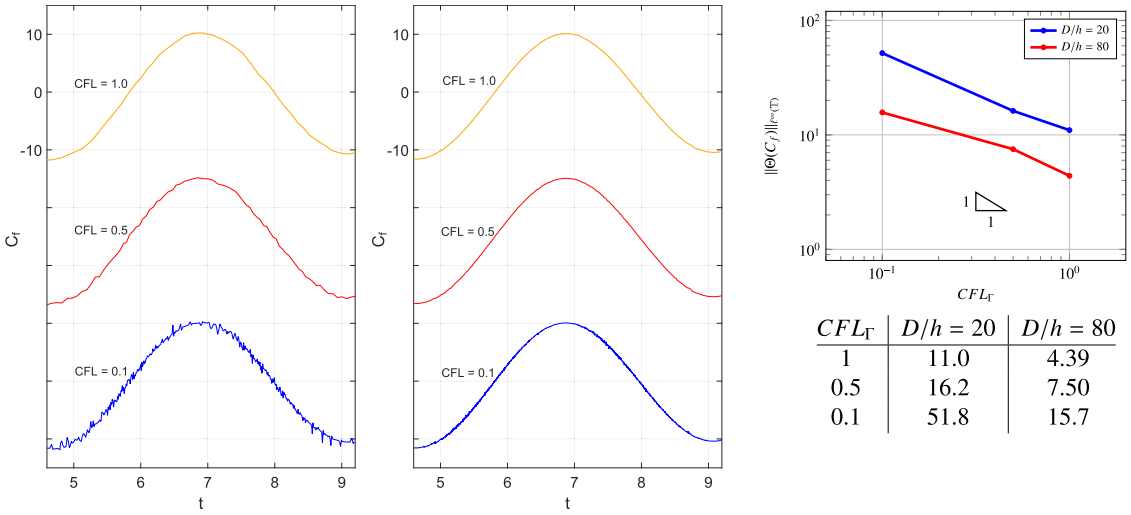
A slip boundary condition ($u_y = 0.0$ m/s) is strongly enforced at the upper and lower boundaries of the rectangular domain. An inflow boundary condition is enforced on the left side by means of the (strong) Dirichlet condition $\boldsymbol{U}_\infty = \{1.0, 0.0\}$ m/s. At the outflow boundary, on the right side of the rectangle, the pressure boundary condition $p_{out} = 0.0\,\text{N/m}^2$ is enforced. The Reynolds number associated with the free-stream flow is $Re_\infty = \rho U_\infty D/\mu = 100$. The computational meshes are obtained by subdividing the domain in squares of size h and then splitting each square into two sub-triangles.

Fig. 14 illustrates the contours of the velocity magnitude and the pressure at time $t = 25.0$ s, both of which exhibit smooth variations and are consistent with the theoretical expectations. The velocity flow patterns around the cylinder show large velocity gradients, associated with the interaction of the incoming flow with the oscillating motion of the cylinder. The pressure profile shows the interaction of the wake of the cylinder



(a) Vorticity at $t = 15.0$ s. (b) Vorticity at $t = 30.0$ s.

Fig. 15. Oscillating cylinder in cross flow: Vorticity at various time instants.



CFL_Γ	$D/h = 20$	$D/h = 80$
1	11.0	4.39
0.5	16.2	7.50
0.1	51.8	15.7

Fig. 16. Oscillating cylinder in cross flow: C_f time histories for different CFL_Γ values, for a grid of size $D/h = 20$ (left) and $D/h = 80$ (middle). On the right, variation of $\|\Theta(C_f)\|_{\ell^\infty(\mathbb{T})}$ as a function of CFL_Γ, for the grids of size $D/h = 20$ (blue) and $D/h = 80$ (red). Numerical values are shown on the table on the bottom-right corner. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

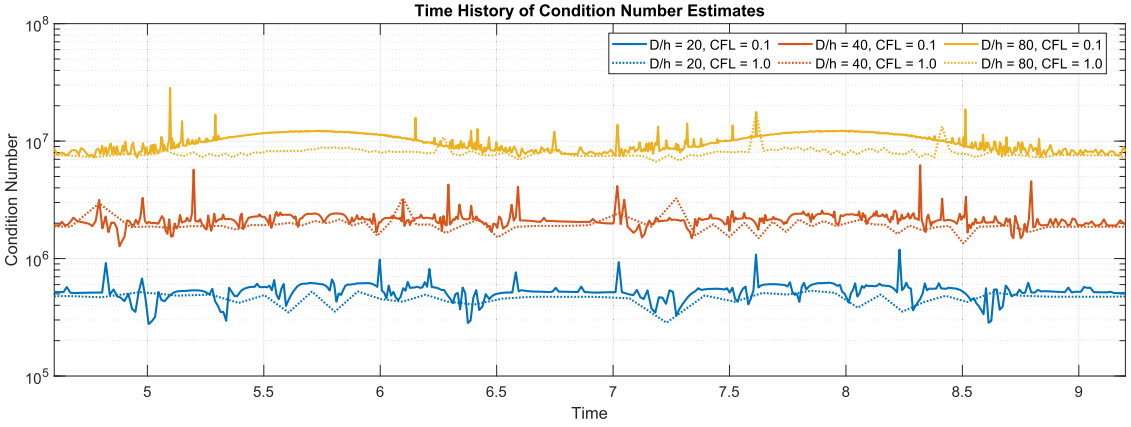


Fig. 17. Oscillating cylinder in cross flow: Time history of the (estimated) condition number for different mesh sizes ($D/h = 20, 40, 80$) and two CFL_Γ numbers (1.0 and 0.1). Solid lines correspond to $CFL_\Gamma = 0.1$, while dotted lines correspond to $CFL_\Gamma = 1.0$, with matching colors representing the same mesh size. Observe that the average condition numbers scale according to h^{-2}, where h is the grid size, and that the variations of condition number are on the order of at most 200–300 % of the average condition number, that is, the instantaneous condition number has the same order of magnitude of the average condition number, throughout the simulations.

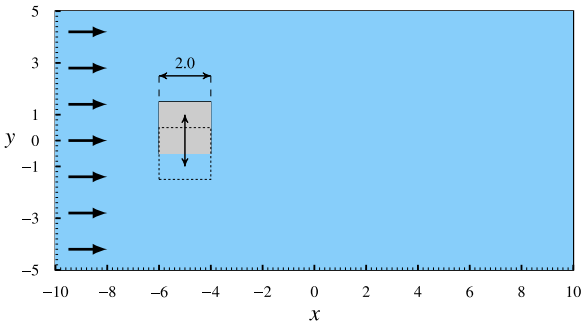
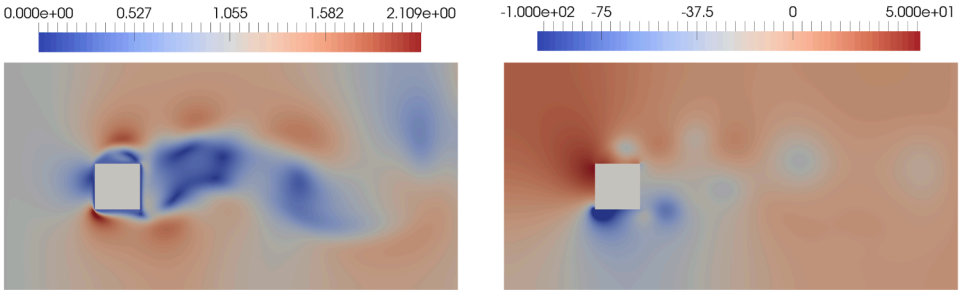


Fig. 18. Oscillating square in cross flow: Setup.



(a) Velocity magnitude at $t = 25.0$ s. (b) Pressure at $t = 25.0$ s.

Fig. 19. Oscillating square in cross flow: Velocity magnitude and pressure distribution at time $t = 25.0$ s.

with the cross flow around it, and gives indication about the expected forces acting on the cylinder. Visualizations of the vorticity at two instants in time are also presented in Fig. 15, to describe the vortex shedding past the cylinder.

Fig. 16 illustrates the time history of C_f with different choices of CFL_Γ numbers, for a course grid with $D/h = 20$ and finer grid with $D/h = 80$. In the computations of the coefficient C_f, we take the length scale ℓ equal to the diameter D of the cylinder. At $CFL_\Gamma = 1$, the oscillations are nearly absent for both grids, in the sense that only on the coarsest grid we observe very minor oscillations near the minimum of the C_f history. However, some oscillations start to creep in as the CFL_Γ approaches 0.1 in the case of the coarse grid. Still, the simulations with CFL_Γ in the range $[0.5, 1]$ seem to perform well from an engineering perspective, since the oscillations amount to just a few percents of the peak values of C_f. For the fine grid, instead, the oscillations seem minimal if non-existent for all values of CFL_Γ. Fig. 16 also shows the decay of pressure oscillations as the CFL_Γ is increased, for the two grid sizes considered. Note that $\|\Theta(C_f)\|_{\ell^\infty(\mathbb{T})}$ scales as $CFL_\Gamma^{-\zeta}$ with $\zeta < 1$, a fact that is actually positive, in the sense that the growth of the oscillations is less than what was reported with other methods in prior works.

Fig. 17 shows the time histories of the (estimated) condition numbers sampled at the last Newton iterate of each time step. Note that we also show results from simulations over an intermediate grid with $D/h = 40$. The time variations of condition number are not very large compared to the average condition number, that is the maximum condition number is on the order of 2–3 times the average. Also, comparing Fig. 17 with Fig. 16, there is no strong correlation between the spikes in the condition number and the pressure oscillations in time. The previous two observations seem to indicate that the condition number is not influenced by the presence of small-cut cells or the initialization issue described in Section 3.5.7.

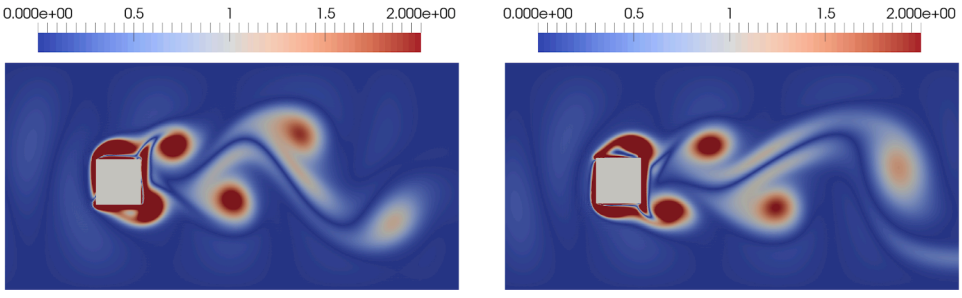
4.3.3. Oscillating square in cross flow

This final two-dimensional test is more demanding than the previous test. As shown in Fig. 18, the geometric configuration is very similar to the one of the benchmark described in Section 4.3.2, except for the fact that the cylinder has been replaced by a square of side $L = 2.0$ m. Consequently, in the definition (44) of C_f, the length-scale is taken as $\ell = L$. The suite of computational grids is identical to the one in the previous section.

The flow properties and grids utilized are the same as the ones used in Section 4.3.2, 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 (47).

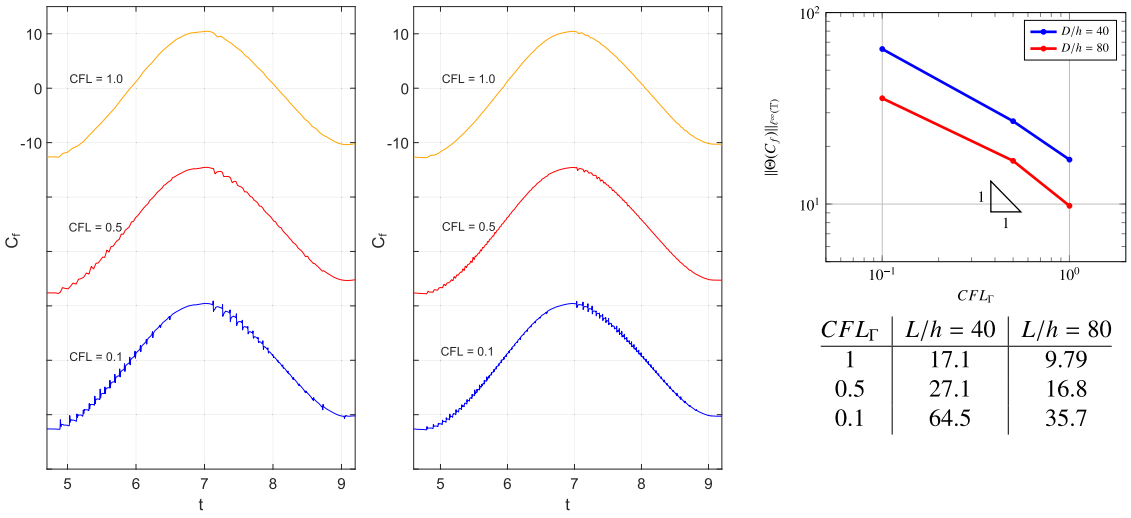
Fig. 19 shows the velocity and pressure fields at $T = 25.0$ s, and Fig. 20 shows the vorticity field at $T = 15.0$ s and $T = 30.0$ s. Fig. 21 plots the time history of C_f for different choices of CFL_Γ numbers and two grids, with $L/h = 40$ and $L/h = 80$, respectively. These figures should be compared with Fig. 16, relative to the oscillating cylinder in cross flow. Oscillations of C_f are more pronounced in Fig. 21, particularly for the coarse grid and with CFL_Γ is in the range $[0.1, 0.5]$. Oscillations are greatly reduced for the finer grid.

Fig. 21 also shows the decay of pressure oscillations as the CFL_Γ is increased, for the two grid sizes considered. Also in this case, $\|\Theta(C_f)\|_{\ell^\infty(\mathbb{T})}$ scales as $CFL_\Gamma^{-\zeta}$ with $\zeta < 1$ and, for CFL_Γ values in the range $[0.5, 1]$, the WSBM delivers results that have very mild oscillations, and can be used for predictive simulations of engineering applications. Note also the relatively rapid decay of oscillations under mesh refinement, since the finer grid is still under-resolving some of the flow features in this vortex-shedding problem.



(a) Vorticity at $t = 15.0$ s. (b) Vorticity at $t = 30.0$ s.

Fig. 20. Oscillating square in cross flow: Vorticity at various time instances.



CFL_Γ	$L/h = 40$	$L/h = 80$
1	17.1	9.79
0.5	27.1	16.8
0.1	64.5	35.7

Fig. 21. Oscillating square in cross flow: C_f time histories for different CFL_Γ values, for a grid of size $D/h = 40$ (left) and $D/h = 80$ (middle). On the right, variation of $\|\Theta(C_f)\|_{\ell^\infty(\mathbb{T})}$ as a function of CFL_Γ, for the grids of size $D/h = 40$ (blue) and $D/h = 80$ (red). Numerical values are shown on the table on the bottom-right corner. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 22 shows the time histories of the (estimated) condition numbers sampled at the last Newton iterate of each time step. Also in this case we have added results from simulations with an intermediate grid with $L/h = 40$. Observe first that the average condition numbers are smaller than for the cylinder in cross flow depicted in Fig. 17, but with larger spikes, in a relative sense. As the CFL_Γ is reduced ten-fold, from 1.0 to 0.1, the average condition numbers increase by a factor 10, for all grids: this typical in general fluid mechanics simulations, even on body-fitted grids. Specifically, we argue that if a small-cut cell effect on the condition number were present, then it will be localized in time, when the boundary motion forms the small cuts. About this point, the largest spikes of the condition number found in Fig. 22 have magnitude about 3–5 times the average, and they seem to occur at the instants in which the motion of the square reverses. This seems to indicate that the largest spikes are more likely produced by specific flow patterns rather than the occurrence of small cuts in the grid. In fact, small cuts most likely will occur in a random way, rather than having specific periodicity. Furthermore, there seems not to be a specific correlation of the large spikes with the pressure oscillations in time, if we compare Fig. 22 with Fig. 21. Instead, pressure oscillations in time seem to correlate with small oscillations of the condition number, which cannot significantly affect the algebraic solver. Again, all these observations seem to confirm that the possible presence of small cuts is not significantly affecting the matrix condition number produced by the WSBM.

4.4. Three-dimensional tests

We consider next a few three-dimensional tests that demonstrate the potential of the WSBM for moving boundary problems characterized by fairly complex geometries expressed in non-standard (STL) formats. Here we consider scenarios closer to the engineering practice, both in terms of the complexity of the geometric shapes and the size of the grids utilized.

In analogy with the previous two-dimensional tests, we study the behavior of pressures oscillations by tracking the time histories of the force (vector) coefficient $\boldsymbol{C}_f$ and the torque (vector) coefficient $\boldsymbol{C}_t$, calculated as

$$\boldsymbol{C}_f = \frac{\int_\Gamma [p\boldsymbol{n} - 2\mu\,\boldsymbol{\varepsilon}(\boldsymbol{u})\,\boldsymbol{n}]\,\mathrm{d}\Gamma}{(1/2)\,\rho\,\ell^3\,f_0^2} \approx \frac{\int_{\tilde{\Gamma}^h} (\tilde{\boldsymbol{n}}\cdot\boldsymbol{n})\,[(p + \nabla p\cdot\boldsymbol{d})\boldsymbol{n} - 2\mu\,\boldsymbol{\varepsilon}(\boldsymbol{u})\boldsymbol{n}]\,\mathrm{d}\tilde{\Gamma}^h}{(1/2)\,\rho\,\ell^3\,f_0^2}\,, \tag{48a}$$

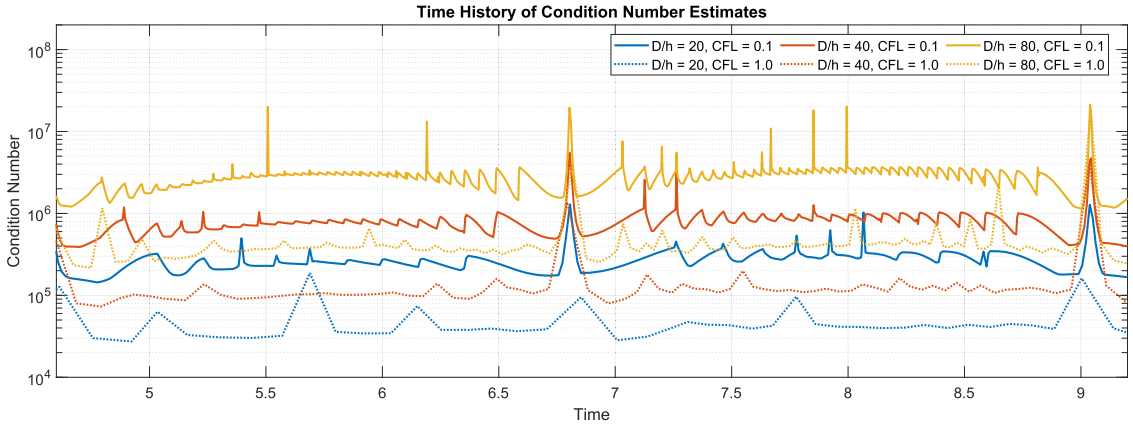


Fig. 22. Oscillating square in cross flow: Time history of the (estimated) condition number for different mesh sizes ($D/h = 20, 40, 80$) and CFL numbers (1.0, 0.1). Solid lines correspond to CFL = 0.1, while dotted lines correspond to CFL = 1.0, with matching colors representing the same mesh size. Observe that the average condition numbers scale according to h^{-2}, where h is the grid size.

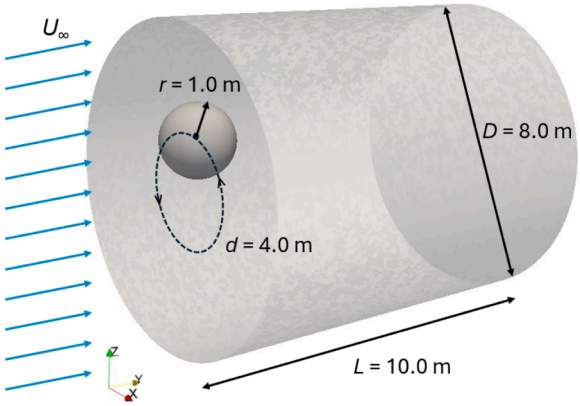


Fig. 23. Rotating sphere in cross flow: Geometry and setup.

Table 2
Description of the unstructured tetrahedral meshes used in the three-dimensional tests. The mesh size h is the target size of the mesh generator and approximately corresponds to the average length of the edges of the tetrahedra in the specific grid considered.

Tests	Mesh size (h)	Number of elements
Rotating sphere in cross flow	0.1 m	3,097,016
	0.05 m	21,080,603
Flow past a rotating gyroid geometry	0.02 m	2,529,367
	0.0125 m	7,662,509
Flow past a rotating Monkey Trefoil	0.1 m	3,097,016
	0.05 m	21,080,603

$$\boldsymbol{C}_t = \frac{\int_\Gamma \boldsymbol{x} \times [p\boldsymbol{n} - 2\mu\,\boldsymbol{\epsilon}(\boldsymbol{u})\,\boldsymbol{n}]\,\mathrm{d}\Gamma}{(1/2)\,\rho\,\ell^4\,f_0^2} \approx \frac{\int_{\tilde{\Gamma}^h} (\tilde{\boldsymbol{n}}\cdot\boldsymbol{n})\left[(\tilde{\boldsymbol{x}}+\boldsymbol{d})\times[(p+\nabla p\cdot\boldsymbol{d})\boldsymbol{n} - 2\mu\,\boldsymbol{\epsilon}(\boldsymbol{u})\,\boldsymbol{n}]\right]\mathrm{d}\tilde{\Gamma}^h}{(1/2)\,\rho\,\ell^4\,f_0^2}\,, \tag{48b}$$

where ℓ is a characteristic length of the problem and $\boldsymbol{x}$ is the position vector of a point on the true boundary Γ, which can also be expressed in terms of a corresponding point $\tilde{\boldsymbol{x}}$ on the surrogate boundary $\tilde{\Gamma}^h$ and the distance vector, as $\boldsymbol{x} = \tilde{\boldsymbol{x}} + \boldsymbol{d}$. Data about the type and size of the grids utilized in computations are reported in Table 2.

4.4.1. Rotating sphere in cross flow

This first three-dimensional test involves a rotating sphere of radius $r = 1.0$ m, immersed in a cylindrical region of diameter $D = 8.0$ m, length $L = 10.0$ m, and symmetry axis aligned with the y-axis of the Cartesian frame. In particular, the cylindrical region spans the interval $[-2.0, 8.0]$

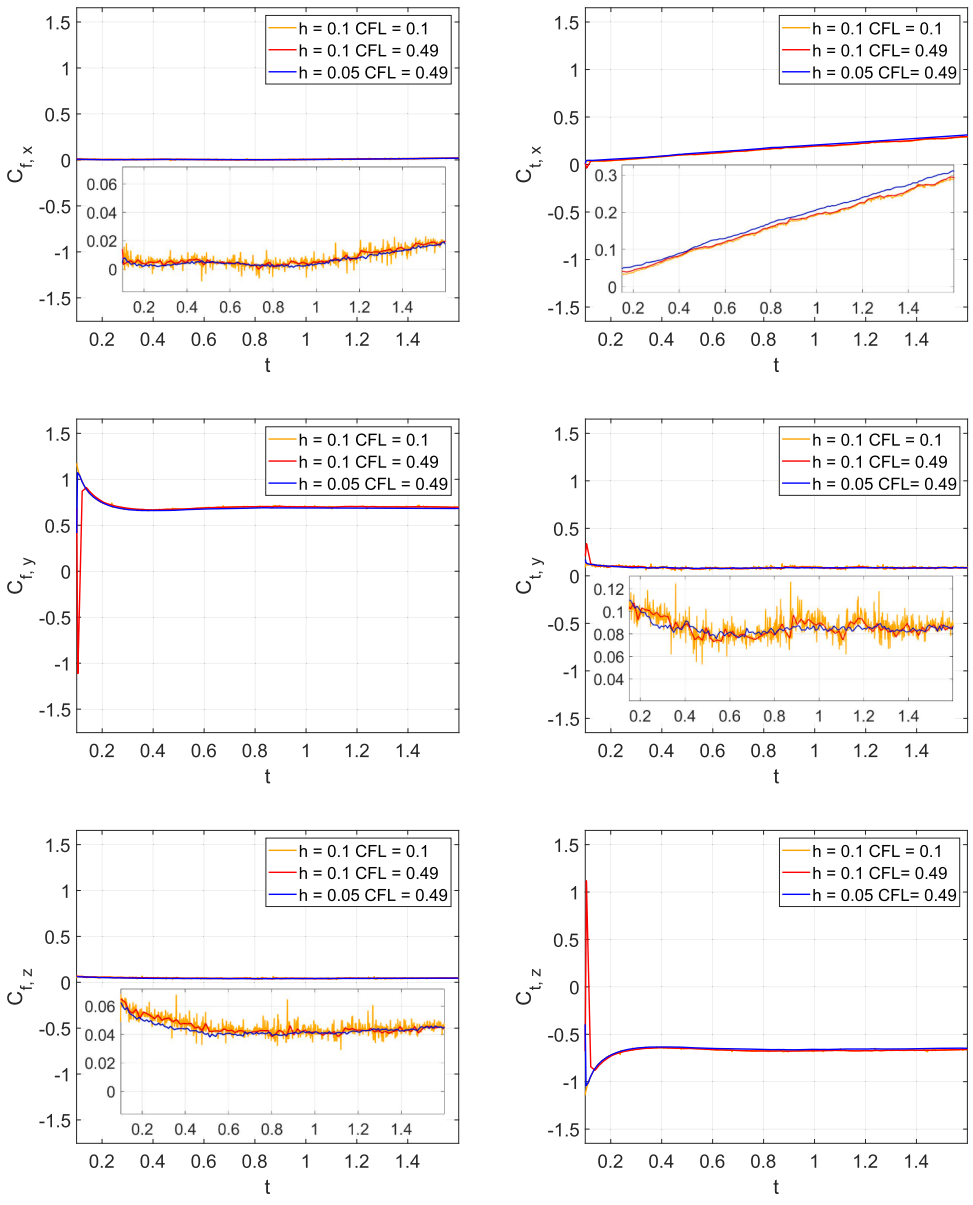


Fig. 24. Rotating sphere in cross flow: Time histories of the force coefficients $\boldsymbol{C}_f$ and $\boldsymbol{C}_t$ along the x, y, and z directions, for three different cases: $h = 0.1$ m and $CFL = 0.1$ (orange); $h = 0.1$ m and $CFL = 0.49$ (red); and $h = 0.05$ m and $CFL = 0.49$ (blue). Zoomed views of the results are also provided. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

along the y-axis. The sphere is initially centered at $\boldsymbol{x}_c = (-2.0, 0.0, 0.0)$ and follows a circular trajectory of diameter $d = 4.0$ m about the y-axis, with constant angular velocity $\boldsymbol{\omega} = -\frac{1}{10}\boldsymbol{e}_y$ rad/s ($\boldsymbol{e}_y$ is the unit vectors aligned with the y-axis). This angular velocity yields the frequency $f_0 = \frac{|\boldsymbol{\omega}|}{2\pi} = \frac{1}{20\pi}$ s^{-1}. A schematic representation of the computational setup is provided in Fig. 23.

The incoming flow has free-stream velocity of $U_\infty = 2.811$ m/s, which is weakly imposed at the inflow and lateral (curved) surface of the cylinder. An outflow boundary condition is imposed on the surface at $y = 8$ m, in the form of a zero-pressure condition. This is also implemented weakly, as a homogenous Neumann-type condition, in which the viscous part of the stress is set free. A no-slip boundary condition is applied on the surface of the sphere using the WSBM.

The fluid density is set to $\rho = 17.78\,\mathrm{kg/m^3}$, the dynamic viscosity is specified as $\mu = 1.0\,\mathrm{kg/(m \cdot s)}$, and the Reynolds number based on the free-stream velocity and the diameter of the sphere is $Re_\infty = \rho U_\infty (2r)/\mu \approx 100$. Observe that the speed of the center of the sphere induced by the rotation, that is $|\dot{\boldsymbol{x}}_c| = |\boldsymbol{\omega} \times \boldsymbol{x}_c| = 0.2$ m/s, is relatively small compared to the free-stream velocity U_∞.

Fig. 24 presents the time histories of the components along the three coordinate directions (x, y, and z) of the force coefficients $\boldsymbol{C}_f$ and torque coefficients $\boldsymbol{C}_t$, defined through equations (48a) and (48b), respectively. When computing $\boldsymbol{C}_f$ and $\boldsymbol{C}_t$, we have taken the length ℓ^* equal to the

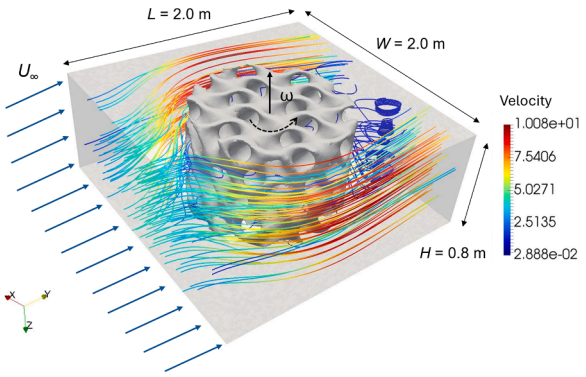


Fig. 25. Flow past a rotating gyroid: Setup and visualization of streamlines, colored by the magnitude of the flow velocity, at time $t = 0.7$ s. Observe the asymmetric pattern of the flow induced by the rotation of the gyroid shape.

diameter $2r$ of the sphere. Note that the force coefficient in the y-direction is larger than in the x and z-directions, by at least an order of magnitude. Fig. 24 also compares results obtained at $CFL = 0.49$, computed according to (40), with a relatively coarse mesh (average size $h = 0.1$ m, in red) and with a refined mesh ($h = 0.05$ m, in blue). While both meshes produce smooth results on the same scale, zoomed views of the x- and z-components of the results reveal oscillations in the calculations. By refining the mesh, these oscillations are significantly reduced, leading to more accurate results. These oscillations become instead more evident if the CFL is reduced to 0.1 for the coarse grid ($h = 0.1$ m), as shown in orange in Fig. 24. The nature of these oscillations was already discussed in the context of the two-dimensional tests presented in Section 4.3, and is related to pressure instabilities in time.

Observe however, that we are considering the zoomed view of the force coefficients along x and z, and the oscillations are negligible in magnitude for the simulations at $CFL = 0.49$, if compared to the y-component of the force coefficient(s), that is the leading order of the forces acting on the moving boundary. The same figure also illustrates the time histories of the components of the torque coefficients $\boldsymbol{C}_t$ defined through Eq. (48b). The components along y of the torque coefficients are comparatively small with respect to the ones along the other two directions and this is consistent with the fact that the sphere's rotation is about the y axis.

Conversely, the torque coefficient in the z-direction is the largest due to the dominant motion of the sphere along this axis during this time frame. The pressure oscillations in time have minimal effect on the torque time histories, for the case at $CFL = 0.49$, while they are about 2 % of the baseline solution, for the coarse grid at $CFL = 0.1$. These results for the torque coefficients are analogous to the results for the force coefficients.

In conclusion, the computational grids utilized in this test are sufficient to keep under control the oscillations of the pressure in time at $CFL = 0.49$. Observe that the levels of mesh refinement used here are not sufficient to capture all the fine features in the boundary layer, but seem sufficient to guarantee mesh convergence of the global force/torque calculations. Note also that from an engineering perspective, one would like to run at the largest CFL compatible with the dynamics of the problem and the time-integration procedure utilized ($CFL = 0.5$ in our case). Hence the proposed WSBM is effective in this test to keep the pressure oscillations under control and very small, for the typical resolutions and time steps that would be most likely utilized in practical computations.

4.4.2. Flow past a rotating gyroid geometry

This test case involves a rotating complex gyroid shape. Rotating bodies are particularly challenging to simulate, with regard to spurious pressure oscillations in time. This is because the classical CFL condition given by (40) is more likely attained at the periphery of a rotating body, while the elements swept by the moving boundary near the axis of rotation experience a relatively small (local) CFL_Γ, given by (39). This situation has potentially negative consequences on the spurious pressure oscillations in time.

The complete configuration is illustrated in Fig. 25, which also includes a visualization of flow streamlines. This flow visualization shows the flow meandering through the rotating gyroid and forming vortices in its wake. The flow velocity increases as the flow goes around the gyroid and shows a strong asymmetry, due to the rotation.

The computational domain is a rectangular prism centered at the origin of the coordinate axes and with length $L = 2$ m, width $W = 2$ m, and height $H = 0.8$ m. A gyroid shape is immersed in the computational domain and rotates about the z axis with angular velocity $\boldsymbol{\omega} = 1.0\,\boldsymbol{e}_z$ rad/s. This angular velocity yields the frequency $f_0 = \frac{|\omega|}{2\pi} = \frac{1}{2\pi}$ s^{-1}. The gyroid is confined in a cylinder of height H and diameter $d = 1.2$ m. The fluid density is set to $\rho = 17.78\,\mathrm{kg/m^3}$, and the dynamic viscosity is specified as $\mu = 1.0\,\mathrm{kg/(m \cdot s)}$. A nominal Reynolds number of the free-stream can be constructed as $Re_\infty = \rho U_\infty d/\mu \approx 60$.

The specific gyroid geometry used in this work was generated using the open-source software MaSMaker [129], with the following parameters: `structure` = TPMS, `TPMS structure` = GYROID, `radius` = 0.6, `height` = 0.8, `cell size` = 0.4, `volume fraction` = 50 %, `AM/3D printing process` = FDM, and `resolution` = 10.

A Hagen-Poiseuille flow of flow rate $Q = U_\infty\, H\, W$, with $U_\infty = 2.811$ m/s, is weakly imposed at the inlet, namely

$$\boldsymbol{u}(x,-1,z) = \left\{ \begin{array}{c} \frac{3}{2} U_\infty \left(1 - \left(\frac{2x}{W}\right)^2\right) \\ 0 \\ 0 \end{array} \right\} \mathrm{m/s}\,, \tag{49}$$

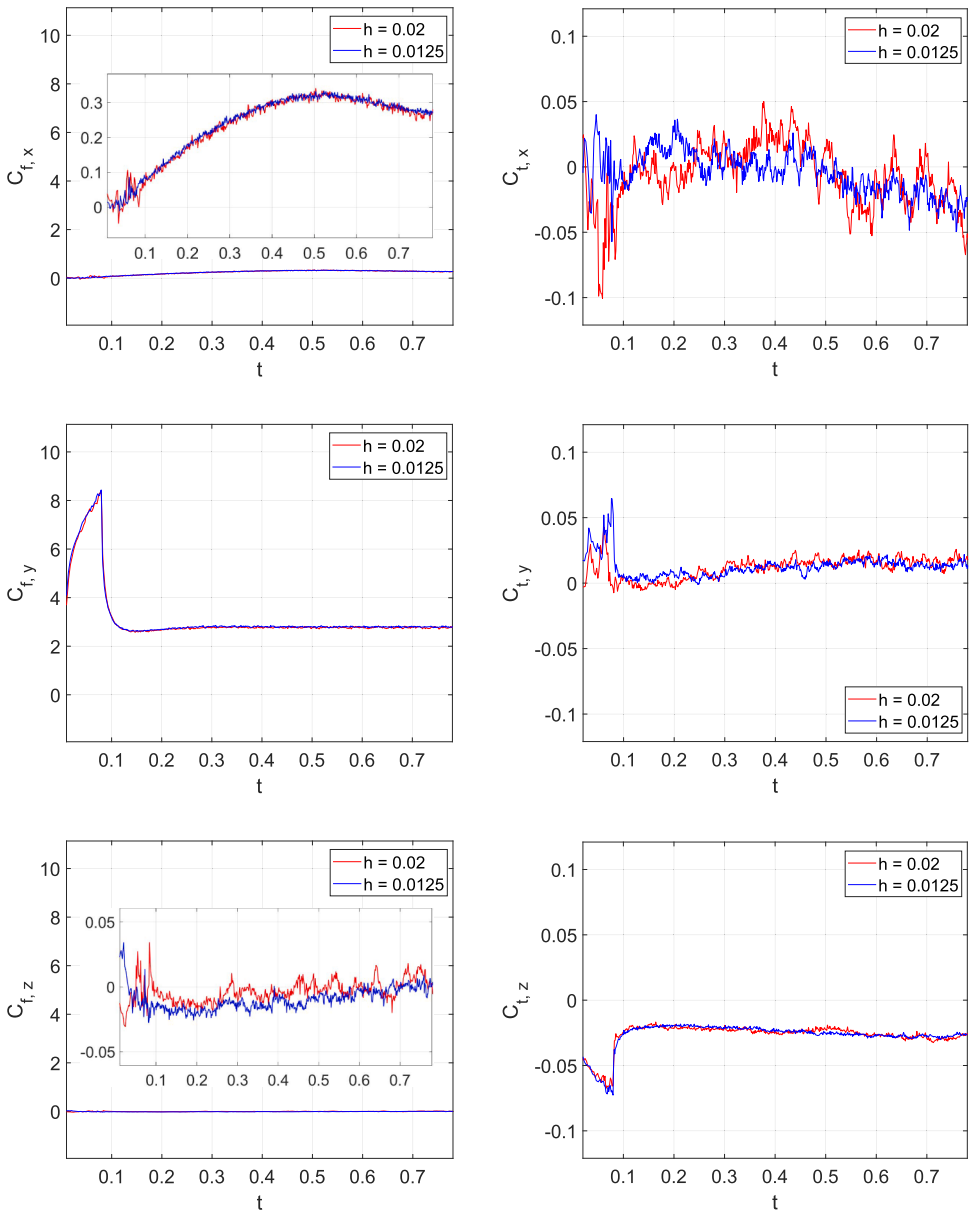


Fig. 26. Flow past a rotating gyroid: Time histories of the force coefficients $\boldsymbol{C}_f$ and $\boldsymbol{C}_t$ along the x, y, and z directions, computed with grids of two average sizes, $h = 0.02$ m and $h = 0.0125$ m, respectively. Zoomed views of the results are also provided.

a slip (i.e., no-penetration) boundary condition is imposed weakly at the top and bottom boundaries,

$$u_z(x, y, \pm 0.4) = 0\,\mathrm{m/s}\,, \tag{50}$$

while a no-slip boundary condition is imposed on the front and back boundaries,

$$\boldsymbol{u}(\pm 1, y, z) = \boldsymbol{0}\,\mathrm{m/s}\,. \tag{51}$$

The outflow condition is weakly imposed as a homogeneous pressure boundary condition, and a no-slip boundary condition is enforced on the surface of the gyroid using the WSBM. The boundary conditions and the angular velocity of rotation of the gyroid are progressively increased to the nominal value with a linear ramp transition until $t = 0.08$ s.

Fig. 26 illustrates the time histories of the components of the force coefficients $\boldsymbol{C}_f$ and torque coefficients $\boldsymbol{C}_t$. When computing $\boldsymbol{C}_f$ and $\boldsymbol{C}_t$, we have taken the length scale ℓ equal to the diameter d of the cylinder enclosing the gyroid: Other definitions are of course possible. The initial peak in the y-components of $\boldsymbol{C}_f$ is attributed to a transient initial phase. Observe that the x- and z-components of the force coefficients are about two orders of magnitude smaller than in the y-components. A zoomed view of the x- and z-components of the force coefficients, shown in Fig. 26,

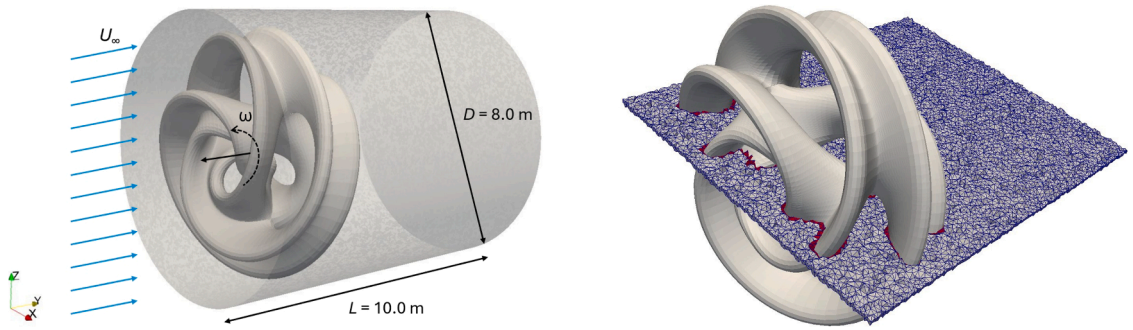


Fig. 27. Flow past a rotating Monkey Trefoil. Left: Problem setup. Right: A slice of the computational mesh around the Monkey Trefoil geometry, where elements marked in red belong to the set of intersected elements $\mathcal{T}_c^h$, elements marked in white belong to the set $\tilde{\mathcal{T}}^h$, and the inactive part of the computational grid is not visible. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

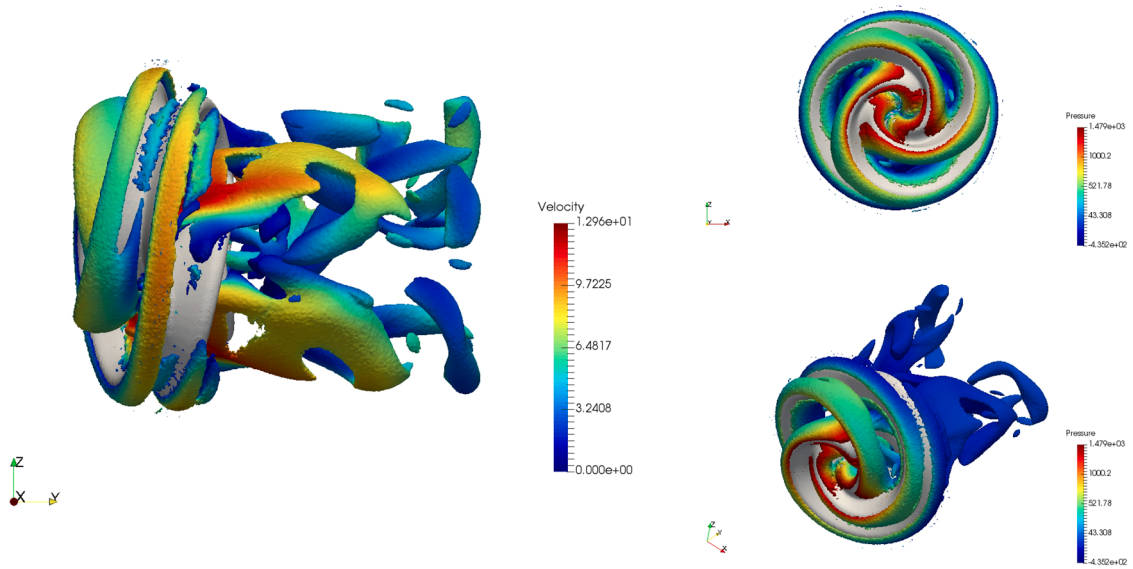


Fig. 28. Flow past a rotating Monkey Trefoil: Q-criterion of the rotating Monkey Trefoil at $t = 1.6$, from three different viewpoints. The color contours represent the magnitude of the Q-criterion, providing insight into the vortical flow structures around the trefoil. The results demonstrate the WSBM's capability capture complex vortex dynamics and intricate flow details.

reveals small-scale oscillations, which are significantly reduced by refining the initial grid, of average size $h = 0.02$ m, to a final grid, of average size $h = 0.0125$ m. We also note that these oscillations are typically on the order of 1 % of the magnitude of the largest force coefficient ($\boldsymbol{C}_{f,y}$), and are therefore minimal from an engineering perspective.

Fig. 26 also depicts the torque coefficients for this test case. While some oscillations are visible, it is important to note that these non-dimensional coefficients are scaled consistently with respect to the force coefficient, that is the torque is normalized as the force times the length ℓ. Within this consistent normalization, we have that the components of $\boldsymbol{C}_t$ are orders of magnitude smaller, respectively, than the components of $\boldsymbol{C}_f$. Hence, any oscillations seen in the torque coefficients are again negligible from an engineering perspective, with respect to the overall forces acting on the system, or, in other words, torques are negligible in this problem.

4.4.3. A complex-geometry mixer: Flow in a pipe past a rotating Monkey Trefoil

This test case features a pipe with a complex-geometry rotating object, with some analogy to a mixing device. The rotating object is the Monkey Trefoil shown in Fig. 27, a minimal, 3-fold symmetrical version of a 3-storey monkey-saddle Scherk tower, bent into a toroid [Séquin]. The specific Monkey Trefoil geometry used in this work was generated using the "Sculpture Generator 1" software, developed by Carlo H. Séquin at The University of California, Berkeley [Séquin], with the following parameters [Séquin]: `branches` = 3, `storeys` = 3, `height` = 2.0, `flange` = 1.5, `thickness` = 0.24, `rim_bulge` = 1.06, `warp` = 360, `twist` = 180, `azimuth` = 0, `texture_tiles` = 1, `detail` = 16, and scaling factors `scaleX` = `scaleY` = `scaleZ` = 1.0.

The computational domain is the same cylindrical region described in the rotating sphere test of Section 4.4.1. Approximately, the Monkey Trefoil is contained in the cylinder of diameter $d = 7.2$ m and length 3.42 m, centered at the origin. The Monkey Trefoil rotates with angular velocity $\boldsymbol{\omega} = -\frac{1}{2}\,\boldsymbol{e}_y$ rad/s, that is with frequency $f_0 = \frac{|\boldsymbol{\omega}|}{2\pi} = \frac{1}{4\pi}\ \mathrm{s}^{-1}$.

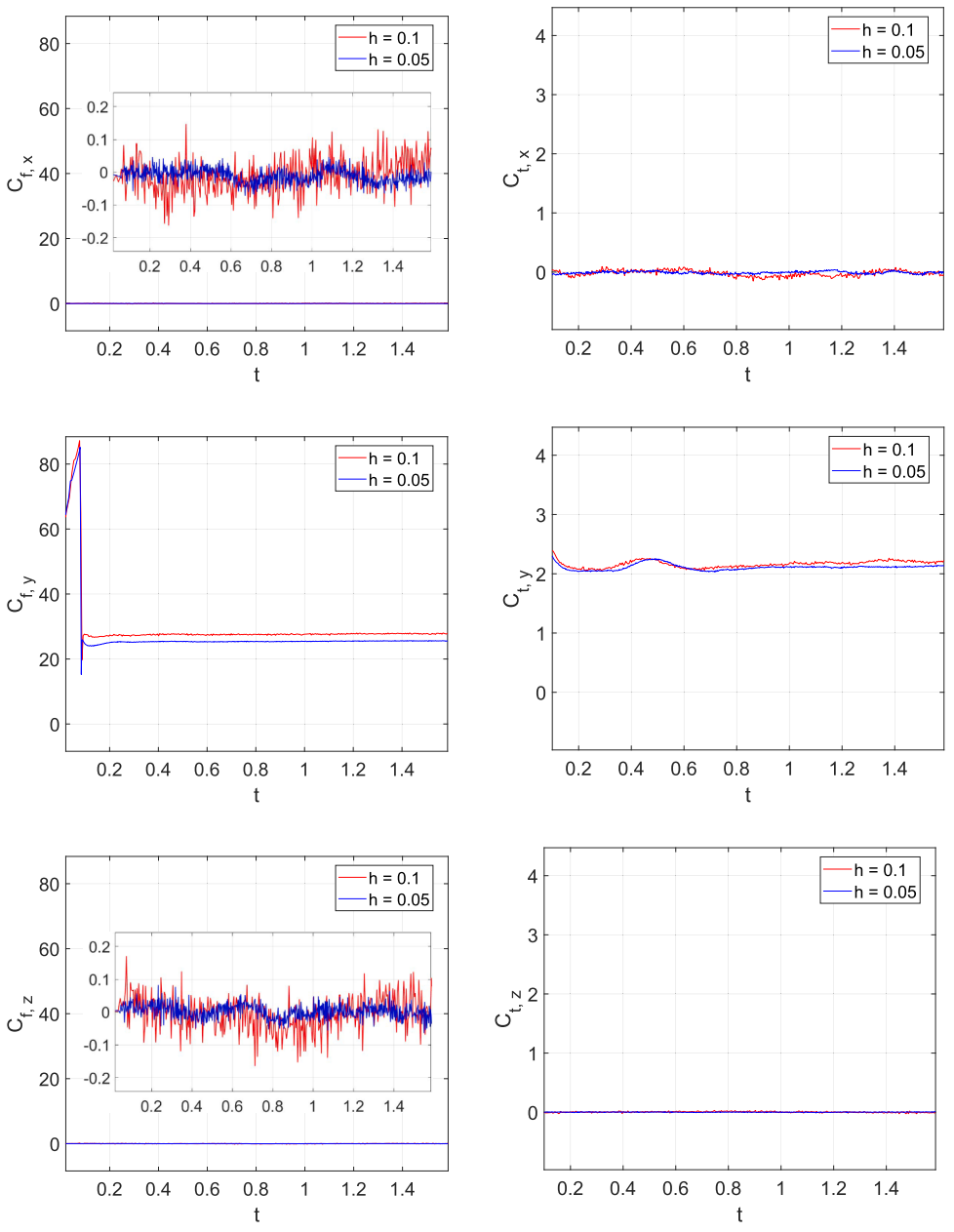


Fig. 29. Flow past a rotating Monkey Trefoil: Time histories of the force coefficients $\boldsymbol{C}_f$ and $\boldsymbol{C}_t$ along the x, y, and z directions, computed with grids of two average sizes, $h = 0.1$ m and $h = 0.005$ m, respectively. Zoomed views of the results are also provided.

A Hagen-Poiseuille flow is weakly imposed (by way of Nitsche's method) at the inlet:

$$\boldsymbol{u}(x,-2,z) = \left\{ \begin{array}{c} 0 \\ 2U_\infty\left(1-\frac{r(x,z)^2}{R^2}\right) \\ 0 \end{array} \right\} \mathrm{m/s}\,, \tag{52}$$

where $U_\infty = 2.811$ m/s, $R = D/2 = 4$ m, and $r(x,z) = \sqrt{x^2+z^2}$ m. No-slip boundary conditions are weakly imposed on the lateral wall of the cylinder. The outflow condition is imposed as a homogeneous traction boundary condition, while a no-slip boundary condition is applied on the surface of the Monkey Trefoil using the WSBM. The boundary conditions and the angular velocity of rotation of the Monkey Trefoil are progressively increased to the nominal value with a linear ramp transition until $t = 0.08$ s.

The fluid has density $\rho = 17.78\,\mathrm{kg/m^3}$ and dynamic viscosity $\mu = 1.0\,\mathrm{kg/(m\cdot s)}$, which yield a nominal free-stream Reynolds number $Re_\infty = \rho U_\infty d/\mu \approx 360$. Fig. 27 also illustrates a slice of the computational mesh: The intersected elements are highlighted in red, while the active

computational domain is shown in white, with edges marked in blue. This visualization helps in understanding the way WSBM captures the complex geometry of the Monkey Trefoil within the unfitted mesh framework. As the Monkey Trefoil rotates, the tracking algorithm dynamically identifies and updates the intersected and active elements at each time step, and recalculates the surrogate domain.

Fig. 28 presents the Q-criterion in the fluid around the rotating Monkey Trefoil at $t = 1.6$ s, from three viewing angles. The Q-criterion isosurfaces are colored by the pressure and velocity magnitude, revealing complex flow features, including the formation of spiral vortices in the wake of the Monkey Trefoil. These visualizations highlight the efficacy of the WSBM in resolving complex flow phenomena over complex geometries.

The force and torque coefficients for the rotating Monkey Trefoil are shown in Fig. 29. Consistent with expectations, the force is predominantly in the y-direction due to the alignment of the inflow, which directly impacts the Monkey Trefoil. The force coefficients in the x- and z-directions remain near zero, comprising only a minor fraction of the total drag. Zoomed views in this figure reveal small oscillations in the results, particularly for the x- and z-directions. These oscillations are minimal when compared to the leading order of the active forces, and are further mitigated by mesh refinement, as seen when comparing results from mesh sizes $h = 0.1$ m and $h = 0.05$ m.

Fig. 29 also presents the torque coefficients. The torque coefficients in the x- and z-directions are nearly zero, while the y-direction exhibits slightly larger values, on the order of unity. This outcome aligns with the dominant force contribution being in the y-direction, which significantly influences the torques. A comparison between the results from mesh sizes $h = 0.1$ m and $h = 0.05$ m demonstrates that mesh refinement effectively reduces torque oscillations, ensuring smoother and more accurate results. Also observe that, because of the consistent normalization, we can compare the orders of magnitude of forces and torques: the force in the y direction is at least an order of magnitude larger than the torque in the same direction, therefore the oscillations in the torque are rather small.

This test case highlights the robustness and versatility of the WSBM approach as an accurate tool to simulate vortex formation in separated flows. Furthermore, it is apparent that time-oscillations of the force and torque coefficients remain on the order of 1 % of the solution for relatively coarse grids and are effectively managed through mesh refinement. These results reinforce the conclusions that the WSBM is a viable option in practical engineering applications, where stability and accuracy are paramount.

5. Summary and future directions

We have presented a WSBM for the incompressible Navier-Stokes equations with moving (rigid) boundaries. The WSBM formulation is based on weighting the SBM formulation using the volume fraction of the active fluid, and like the SBM, does not require integration over cut cells, and associated complexities in the data structures.

We studied the accuracy and robustness of the proposed, approach, with particular attention to conservation errors and pressure oscillations occurring at very small time steps. We showed that although these pressure oscillations are not completely prevented by the WSBM, they are in general small (on the order of 1 % or less) for the grid resolutions required by engineering applications and are further reduced with mesh refinement. Hence, the simulations presented in this work indicate that the WSBM is viable in engineering applications involving complex moving three-dimensional geometries interacting with fluid flows.

Future research will be oriented toward attacking fluid-structure interaction (FSI) problems with the WSBM. For this purpose, a structural mechanics algorithm will need to be coupled with the WSBM fluid solver. For simple geometric shapes, this can be achieved discretizing the structure with a Lagrangian body-fitted grid and defining the active WSBM (Eulerian) domain with respect to the geometry of the structure. More advanced options for complex geometries would instead involve using an immersed approach for both the structure and the fluid, using a traditional SBM implementation for the structure and a WSBM implementation for the fluid. For example, a distance map can be defined to link the surrogate solid boundary and the corresponding true solid boundary. Then a second distance map can be defined, between the inner/outer surrogate boundaries of the fluid domain and the solid boundary. These are the key ingredients of the geometric setup of a typical FSI problem with a combination of SBM/WSBM.

CRediT authorship contribution statement

Danjie Xu: Writing - review & editing, Writing - original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation, Conceptualization; Oriol Colomés: Software, Methodology, Investigation, Formal analysis, Conceptualization; Alex Main: Visualization, Validation, Supervision, Investigation, Formal analysis, Conceptualization; Kangan Li: Visualization, Validation, Supervision, Software, Methodology, Investigation; Nabil M. Atallah: Software, Methodology, Investigation, Formal analysis, Conceptualization; Nabil Abboud: Supervision, Software, Methodology; Guglielmo Scovazzi: Writing - review & editing, Writing - original draft, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

Guglielmo Scovazzi reports financial support was provided by National Science Foundation. Guglielmo Scovazzi reports financial support was provided by Office of Naval Research. Guglielmo Scovazzi reports financial support was provided by ExxonMobil Research and Engineering Company. Guglielmo Scovazzi is an Associated Editor of the Journal of Computational Physics. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was partly supported by the National Science Foundation, under the grants DMS 2207164 and DMS 2409919 (Computational Mathematics Program, Division of Mathematical Sciences), by the U.S. Office of Naval Research under grant N00014-14-1-0311, and by ExxonMobil Upstream Research Company (Houston, TX). This support is gratefully acknowledged. This manuscript has been authored by Lawrence Livermore National Security, LLC under Contract No. DE-AC52-07NA27344 with the U.S. Department of Energy (LLNL-JRNL-2005992). The United States Government retains, and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

Appendix A. Approximation with stabilized finite elements

It is well known that the pure Galerkin formulation of the Navier-Stokes equations with finite element interpolation spaces of equal-order is not stable. Specifically, the Galerkin formulation outlined in Section 3.5 is not stable in the advection dominated limit and with respect to the incompressibility constraint (as the discretization of the associated Stokes flow problem does not satisfy the LBB inf-sup conditions). The numerical tests presented are then performed with a stabilized variational multiscale method [132,133]. Namely, we modify (18) as

Find $\boldsymbol{u} \in \boldsymbol{V}^h(\Omega^h)$ and $p \in Q^h(\Omega^h)$ such that, $\forall \boldsymbol{w}_\alpha \in \boldsymbol{V}^h_\alpha(\Omega^h)$ and $\forall q_\alpha \in Q^h_\alpha(\Omega^h)$,

$$0 = \mathbb{NS}(\boldsymbol{u}, p; \boldsymbol{w}_\alpha, q_\alpha) + \mathbb{STAB}(\boldsymbol{u}, p; \boldsymbol{w}_\alpha, q_\alpha) \,, \tag{A.1}$$

where $\mathbb{NS}(\boldsymbol{u}, p; \boldsymbol{w}_\alpha, q_\alpha)$ denotes the right hand side of (18) and

$$\mathbb{STAB}(\boldsymbol{u}, p; \boldsymbol{w}_\alpha, q_\alpha) = -(\,\rho\boldsymbol{u}\cdot\nabla\boldsymbol{w}_\alpha\,,\,\boldsymbol{u}'\,)_{I(\Omega^h)} + (\nabla q_\alpha, \boldsymbol{u}')_{I(\Omega^h)} - (\nabla\cdot\boldsymbol{w}_\alpha, p')_{I(\Omega^h)} + (\,\boldsymbol{w}_\alpha\,,\,\rho\boldsymbol{u}'\cdot\nabla\boldsymbol{u}\,)_{I(\Omega^h)} \,. \tag{A.2}$$

Here $I(\Omega^h) = \bigcup_{T\in\tilde{\mathcal{T}}^h\cup\mathcal{T}^h_c} \mathrm{int}(T)$ indicates the union of the interiors of the elements belonging to the surrogate domain and the following definitions hold:

$$\boldsymbol{u}' = -\tau_m \left(\rho\,\boldsymbol{u}_{,t} + \rho\,\boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p - \nabla\cdot(2\mu\,\boldsymbol{\epsilon}(\boldsymbol{u})) - \rho\,\boldsymbol{b}\right) \,, \tag{A.3}$$

$$p' = -\tau_c\,\nabla\cdot\boldsymbol{u} \,, \tag{A.4}$$

$$\tau_m = \left(\left(\frac{2\rho}{\Delta t}\right)^2 + \rho^2\boldsymbol{u}\cdot\boldsymbol{G}\boldsymbol{u} + 9\,(2\mu)^2\,\frac{\boldsymbol{G}:\boldsymbol{G}}{3}\right)^{-1/2} \,, \tag{A.5}$$

$$\tau_c = \frac{1}{8\,\mathrm{tr}(\boldsymbol{G})\,\tau_m} \,. \tag{A.6}$$

Here $\boldsymbol{G}$ is the metric tensor, that is $G_{ij} = \frac{\partial\xi_k}{\partial x_i}\frac{\partial\xi_k}{\partial x_j}$, where $\boldsymbol{\xi}$ is the position vector in the parent finite element domain, and $\boldsymbol{x}$ is the position vector in the spatial configuration. More details can be found in [132,133].

Remark 7. The proposed stabilization approach does not perturb the conservation statements discussed in Section 3.5.5. For example, if $q = 1$, that is $q_\alpha = \alpha$ and $\boldsymbol{w} = \boldsymbol{0}$, that is $\boldsymbol{w}_\alpha = 0$, we have $\mathbb{STAB}(\boldsymbol{u}, p; \boldsymbol{0}, \alpha) = 0$, that is the proposed stabilization operators do not change the global mass conservation balance. Similarly, if $\boldsymbol{w} = \boldsymbol{e}_i$, that is $\boldsymbol{w}_\alpha = \alpha\boldsymbol{e}_i$, and $q = -\rho\boldsymbol{u}\cdot\boldsymbol{e}_i$, that is $q_\alpha = -\alpha\rho\boldsymbol{u}\cdot\boldsymbol{e}_i$, yields

$$\begin{aligned}\mathbb{STAB}[\mu](\boldsymbol{u}, p; \alpha\boldsymbol{e}_i, -\alpha\rho\boldsymbol{u}\cdot\boldsymbol{e}_i) &= -(\,\alpha\rho\boldsymbol{u}\cdot\nabla\boldsymbol{e}_i\,,\,\boldsymbol{u}'\,)_{I(\Omega^h)} + (\alpha\nabla(-\rho\boldsymbol{u}\cdot\boldsymbol{e}_i), \boldsymbol{u}')_{I(\Omega^h)} \\ &\quad - (\alpha\nabla\cdot\boldsymbol{e}_i, p')_{I(\Omega^h)} + (\,\alpha\boldsymbol{e}_i\,,\,\rho\boldsymbol{u}'\cdot\nabla\boldsymbol{u}\,)_{I(\Omega^h)} \\ &= -(\alpha\rho\boldsymbol{u}'\cdot\nabla\boldsymbol{u}, \boldsymbol{e}_i)_{I(\Omega^h)} + (\,\alpha\boldsymbol{e}_i\,,\,\rho\boldsymbol{u}'\cdot\nabla\boldsymbol{u}\,)_{I(\Omega^h)} \\ &= 0 \,,\end{aligned} \tag{A.7}$$

that is the proposed stabilization operators do not change the global momentum conservation budget.

References

[1] W.F. Noh, CEL: A time-dependent, two-space-dimensional, coupled Eulerian-Lagrange code, Technical Report, Lawrence Radiation Lab., Univ. of California, Livermore, 1963.
[2] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (3) (1977) 220–252.
[3] C.S. Peskin, The immersed boundary method, Acta numerica 11 (2002) 479–517.
[4] R. Glowinski, T.-W. Pan, J. Periaux, Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies, Comput. Methods Appl. Mech. Eng. 151 (1–2) (1998) 181–194.
[5] D. Boffi, L. Gastaldi, A finite element approach for the immersed boundary method, Comput. Struct. 81 (8) (2003) 491–501. K.J Bathe 60th Anniversary Issue, http://www.sciencedirect.com/science/article/pii/S0045794902004042. https://doi.org/10.1016/S0045-7949(02)00404-2
[6] L. Zhang, A. Gerstenberger, X. Wang, W.K. Liu, Immersed finite element method, Comput. Methods Appl. Mech. Eng. 193 (21–22) (2004) 2051–2067.
[7] W.K. Liu, Y. Liu, D. Farrell, L. Zhang, X.S. Wang, Y. Fukui, N. Patankar, Y. Zhang, C. Bajaj, J. Lee, et al., Immersed finite element method and its applications to biological systems, Comput. Methods Appl. Mech. Eng. 195 (13–16) (2006) 1722–1749.
[8] L.T. Zhang, M. Gay, Immersed finite element method for fluid-structure interactions, J. Fluids Struct. 23 (6) (2007) 839–857.
[9] X. Wang, L.T. Zhang, Interpolation functions in the immersed boundary and finite element methods, Comput. Mech. 45 (4) (2010) 321–334.
[10] X. Wang, L.T. Zhang, Modified immersed finite element method for fully-coupled fluid–structure interactions, Comput. Methods Appl. Mech. Eng. 267 (2013) 150–169.
[11] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (1) (2000) 35–60.
[12] F. Sotiropoulos, X. Yang, Immersed boundary methods for simulating fluid–structure interaction, Prog. Aerosp. Sci. 65 (2014) 1–21.

[13] X. Zeng, C. Farhat, A systematic approach for constructing higher-order immersed boundary and ghost fluid methods for fluid–structure interaction problems, J. Comput. Phys. 231 (7) (2012) 2892–2923.
[14] J.H. Seo, R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, J. Comput. Phys. 230 (19) (2011) 7347–7363. https://www.sciencedirect.com/science/article/pii/S0021999111003524. https://doi.org/10.1016/j.jcp.2011.06.003
[15] D. Kim, H. Choi, Immersed boundary method for flow around an arbitrarily moving body, J. Comput. Phys. 212 (2) (2006) 662–680. https://www.sciencedirect.com/science/article/pii/S0021999105003505. https://doi.org/10.1016/j.jcp.2005.07.010
[16] J. Lee, J. Kim, H. Choi, K.-S. Yang, et al., Sources of spurious force oscillations from an immersed boundary method for moving-body problems, J. Comput. Phys. 230 (7) (2011) 2677–2695. https://www.sciencedirect.com/science/article/pii/S002199911100012X. https://doi.org/10.1016/j.jcp.2011.01.004
[17] C.-C. Liao, Y.-W. Chang, C.-A. Lin, J.M. McDonough, Simulating flows with moving rigid boundary using immersed-boundary method, Computers & Fluids 39 (1) (2010) 152–167. https://www.sciencedirect.com/science/article/pii/S0045793009001108. https://doi.org/10.1016/j.compfluid.2009.07.011
[18] J. Lee, D. You, An implicit ghost-cell immersed boundary method for simulations of moving body problems with control of spurious force oscillations, J. Comput. Phys. 233 (2013) 295–314.
[19] L. Schneiders, D. Hartmann, M. Meinke, W. Schröder, An accurate moving boundary formulation in cut-cell methods, J. Comput. Phys. 235 (2013) 786–809. https://www.sciencedirect.com/science/article/pii/S0021999112005839. https://doi.org/10.1016/j.jcp.2012.09.038
[20] Y.-H. Tseng, J.H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2) (2003) 593–623. https://www.sciencedirect.com/science/article/pii/S0021999103004108. https://doi.org/10.1016/j.jcp.2003.07.024
[21] J. Yang, E. Balaras, An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, J. Comput. Phys. 215 (1) (2006) 12–40. https://www.sciencedirect.com/science/article/pii/S0021999105004778. https://doi.org/10.1016/j.jcp.2005.10.035
[22] F. Gibou, L. Chen, D. Nguyen, S. Banerjee, A level set based sharp interface method for the multiphase incompressible Navier–Stokes equations with phase change, J. Comput. Phys. 222 (2) (2007) 536–555.
[23] W.D. Henshaw, A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids, J. Comput. Phys. 113 (1) (1994) 13–25.
[24] W. Henshaw, N.A. Petersson, A Split-Step Scheme for the Incompressible Navier-Stokes, Technical Report, Lawrence Livermore National Lab., CA (US), 2001.
[25] S.P. Domino, Towards verification of sliding mesh algorithms for complex applications using MMS, in: Proceedings of the Summer Program, 2010, p. 167.
[26] J.W. Banks, W.D. Henshaw, D.W. Schwendeman, An analysis of a new stable partitioned algorithm for FSI problems. part I: incompressible flow and elastic solids, J. Comput. Phys. 269 (2014) 108–137.
[27] Q. Tang, A stable partitioned FSI algorithm for rigid bodies and incompressible flow. part I: model problem analysis, J. Comput. Phys. 343 (2017) 432–468.
[28] Q. Tang, A stable partitioned FSI algorithm for rigid bodies and incompressible flow in three dimensions, J. Comput. Phys. 373 (2018) 455–492.
[29] K. Höllig, U. Reif, J. Wipper, Weighted extended B-spline approximation of Dirichlet problems, SIAM J. Numer. Anal. 39 (2) (2001) 442–462.
[30] A. Hansbo, P. Hansbo, An unfitted finite element method, based on Nitsche's method, for elliptic interface problems, Comput. Methods Appl. Mech. Eng. 191 (47) (2002) 5537–5552. http://www.sciencedirect.com/science/article/pii/S0045782502005248. https://doi.org/10.1016/S0045-7825(02)00524-8
[31] K. Hollig, Finite element methods with B-splines, 26, Siam, 2003.
[32] J. Parvizian, A. Düster, E. Rank, Finite cell method, Comput. Mech. 41 (1) (2007) 121–133.
[33] A. Düster, J. Parvizian, Z. Yang, E. Rank, The finite cell method for three-dimensional problems of solid mechanics, Comput. Methods Appl. Mech. Eng. 197 (45–48) (2008) 3768–3782.
[34] E. Burman, P. Hansbo, Fictitious domain finite element methods using cut elements: I. a stabilized Lagrange multiplier method, Comput. Methods Appl. Mech. Eng. 199 (41–44) (2010) 2680–2686.
[35] E. Burman, Ghost penalty, C.R. Math. 348 (21–22) (2010) 1217–1220.
[36] B. Cockburn, D. Gupta, F. Reitich, Boundary-conforming discontinuous Galerkin methods via extensions from subdomains, J. Sci. Comput. 42 (1) (2010) 144.
[37] E. Burman, P. Hansbo, Fictitious domain finite element methods using cut elements: II. a stabilized Nitsche method, Appl. Numer. Math. 62 (4) (2012) 328–341.
[38] D. Schillinger, L. Dede, M.A. Scott, J.A. Evans, M.J. Borden, E. Rank, T.J.R. Hughes, An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces, Comput. Methods Appl. Mech. Eng. 249–252 (2012) 116 – 150. http://www.sciencedirect.com/science/article/pii/S004578251200093X. https://doi.org/10.1016/j.cma.2012.03.017
[39] B. Cockburn, M. Solano, Solving Dirichlet boundary-value problems on curved domains by extensions from subdomains, SIAM Journal on Scientific Computing 34 (1) (2012) A497–A519.
[40] J. Baiges, R. Codina, F. Henke, S. Shahmiri, W.A. Wall, A symmetric method for weakly imposing Dirichlet boundary conditions in embedded finite element meshes, Int. J. Numer. Methods Eng. 90 (5) (2012) 636–658.
[41] T. Rüberg, F. Cirak, Subdivision-stabilised immersed B-spline finite elements for moving boundary flows, Comput. Methods Appl. Mech. Eng. 209 (2012) 266–283.
[42] T. Rüberg, F. Cirak, A fixed-grid B-spline finite element technique for fluid–structure interaction, Int. J. Numer Methods Fluids 74 (9) (2014) 623–660.
[43] E. Burman, M.A. Fernández, An unfitted Nitsche method for incompressible fluid–structure interaction using overlapping meshes, Comput. Methods Appl. Mech. Eng. 279 (2014) 497–514.
[44] B. Cockburn, W. Qiu, M. Solano, A priori error analysis for HDG methods using extensions from subdomains to achieve boundary conformity, Math. Comput. 83 (286) (2014) 665–699.
[45] A. Massing, M. Larson, A. Logg, M. Rognes, A Nitsche-based cut finite element method for a fluid-structure interaction problem, Comm. App. Math Comp. Sci. 10 (2) (2015) 97–120.
[46] B. Schott, U. Rasthofer, V. Gravemeier, W.A. Wall, A face-oriented stabilized Nitsche-type extended variational multiscale method for incompressible two-phase flow, Int. J. Numer. Methods Eng. 104 (7) (2015) 721–748.
[47] E. Burman, S. Claus, P. Hansbo, M.G. Larson, A. Massing, CutFEM: discretizing geometry and partial differential equations, Int. J. Numer. Methods Eng. 104 (7) (2015) 472–501.
[48] F. Xu, D. Schillinger, D. Kamensky, V. Varduhn, C. Wang, M.-C. Hsu, The tetrahedral finite cell method for fluids: immersogeometric analysis of turbulent flow around complex geometries, Comput. Fluids 141 (2016) 135–154.
[49] A. Lozinski, A new fictitious domain method: optimal convergence without cut elements, C.R. Math. 354 (7) (2016) 741–746.
[50] W.G. Dettmer, C. Kadapa, D. Perić, A stabilised immersed boundary method on hierarchical B-spline grids, Comput. Methods Appl. Mech. Eng. 311 (2016) 415–437.
[51] C. Kadapa, W.G. Dettmer, D. Perić, A stabilised immersed boundary method on hierarchical B-spline grids for fluid–rigid body interaction with solid–solid contact, Comput. Methods Appl. Mech. Eng. 318 (2017) 242–269.
[52] E. Burman, P. Hansbo, M.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, Springer, 2017, pp. 183–192.
[53] D. Kamensky, M.-C. Hsu, Y. Yu, J.A. Evans, M.S. Sacks, T.J.R. Hughes, Immersogeometric cardiovascular fluid–structure interaction analysis with divergence-conforming B-splines, Comput. Methods Appl. Mech. Eng. 314 (2017) 408–472.
[54] E. Burman, P. Hansbo, M. Larson, A cut finite element method with boundary value correction, Math. Comput. 87 (310) (2018) 633–657.
[55] E. Burman, D. Elfverson, P. Hansbo, M.G. Larson, K. Larsson, Shape optimization using the cut finite element method, Comput. Methods Appl. Mech. Eng. 328 (2018) 242–261.
[56] C. Kadapa, W.G. Dettmer, D. Perić, A stabilised immersed framework on hierarchical B-spline grids for fluid-flexible structure interaction with solid–solid contact, Comput. Methods Appl. Mech. Eng. 335 (2018) 472–489.
[57] E. Burman, P. Hansbo, M.G. Larson, Dirichlet Boundary Value Correction using Lagrange Multipliers, arXiv preprint arXiv:1903.07104 (2019).
[58] M. Núñez, I. López, J. Baiges, R. Rossi, An embedded approach for the solution of the full potential equation with finite elements, Comput. Methods Appl. Mech. Eng. 388 (2022) 114244.
[59] R. Zorrilla, A. Larese de Tetto, R. Rossi, A discontinuous Nitsche-based finite element formulation for the imposition of the Navier-slip condition over embedded volumeless geometries, Int. J. Numer. Methods Fluids 93 (9) (2021) 2968–3003.
[60] R. Zorrilla, E. Soudah, R. Rossi, Computational modeling of the fluid flow in type b aortic dissection using a modified finite element embedded formulation, Biomech. Model Mechanobiol. 19 (2020) 1565–1583.

[61] R. Zorrilla, R. Rossi, R. Wüchner, E. Oñate, An embedded finite element framework for the resolution of strongly coupled fluid–structure interaction problems. application to volumetric and membrane-like structures, Comput. Methods Appl. Mech. Eng. 368 (2020) 113179.
[62] S. Kang, A. Masud, A variational multiscale method with immersed boundary conditions for incompressible flows, Meccanica 56 (6) (2021) 1397–1422.
[63] S. Kang, J. Kwack, A. Masud, Variational coupling of non-matching discretizations across finitely deforming fluid–structure interfaces, Int. J. Numer. Methods Fluids 94 (6) (2022) 678–718.
[64] S. Badia, F. Verdugo, A.F. Martín, The aggregated unfitted finite element method for elliptic problems, Comput. Methods Appl. Mech. Eng. 336 (2018) 533–553.
[65] S. Badia, A.F. Martin, F. Verdugo, Mixed aggregated finite element methods for the unfitted discretization of the Stokes problem, SIAM journal on scientific computing 40 (6) (2018) B1541–B1576.
[66] F. Verdugo, A.F. Martín, S. Badia, Distributed-memory parallelization of the aggregated unfitted finite element method, Comput. Methods Appl. Mech. Eng. 357 (2019) 112583.
[67] S. Badia, A.F. Martín, E. Neiva, F. Verdugo, The aggregated unfitted finite element method on parallel tree-based adaptive meshes, SIAM Journal on Scientific Computing 43 (3) (2021) C203–C234.
[68] S. Badia, P.A. Martorell, F. Verdugo, Space–time unfitted finite elements on moving explicit geometry representations, Comput. Methods Appl. Mech. Eng. 428 (2024) 117091.
[69] A.J. Lew, M. Negri, Optimal convergence of a discontinuous-Galerkin-based immersed boundary method, ESAIM: Mathematical Modelling and Numerical Analysis 45 (4) (2011) 651–674.
[70] R. Rangarajan, A. Lew, G.C. Buscaglia, A discontinuous-Galerkin-based immersed boundary method with non-homogeneous boundary conditions and its application to elasticity, Comput. Methods Appl. Mech. Eng. 198 (17–20) (2009) 1513–1534.
[71] A.J. Lew, G.C. Buscaglia, A discontinuous-Galerkin-based immersed boundary method, Int. J. Numer. Methods Eng. 76 (4) (2008) 427–454.
[72] T. Lin, Y. Lin, X. Zhang, Partially penalized immersed finite element methods for elliptic interface problems, SIAM J. Numer. Anal. 53 (2) (2015) 1121–1144.
[73] Z. Li, T. Lin, Y. Lin, R.C. Rogers, An immersed finite element space and its approximation capability, Numerical Methods for Partial Differential Equations: An International Journal 20 (3) (2004) 338–367.
[74] R.E. Ewing, Z. Li, T. Lin, Y. Lin, The immersed finite volume element methods for the elliptic interface problems, Math. Comput. Simul. 50 (1–4) (1999) 63–76.
[75] X. He, T. Lin, Y. Lin, X. Zhang, Immersed finite element methods for parabolic equations with moving interface, Numer Methods Partial Differ Equ 29 (2) (2013) 619–646.
[76] S. Adjerid, N. Chaabane, T. Lin, An immersed discontinuous finite element method for Stokes interface problems, Comput. Methods Appl. Mech. Eng. 293 (2015) 170–190.
[77] Z. Li, T. Lin, X. Wu, New Cartesian grid methods for interface problems using the finite element formulation, Numerische Mathematik 96 (1) (2003) 61–98.
[78] J.A. Nitsche, Uber ein variationsprinzip zur losung Dirichlet-Problemen bei verwendung von teilraumen, die keinen randbedingungen unteworfen sind, Abh. Math. Sem. Univ., Hamburg 36 (1971) 9–15.
[79] E. Burman, P. Hansbo, Fictitious domain methods using cut elements: III. a stabilized Nitsche method for Stokes' problem, ESAIM: Mathematical Modelling and Numerical Analysis 48 (3) (2014) 859–874.
[80] A. Düster, O. Allix, Selective enrichment of moment fitting and application to cut finite elements and cells, Comput. Mech. 65 (2020) 429–450.
[81] L. Kudela, N. Zander, T. Bog, S. Kollmannsberger, E. Rank, Efficient and accurate numerical quadrature for immersed boundary methods, Advanced modeling and simulation in engineering sciences 2 (2015) 1–22.
[82] P. Antolin, T. Hirschler, Quadrature-free immersed isogeometric analysis, Eng. Comput. 38 (5) (2022) 4475–4499.
[83] Y. Sudhakar, A. Sommariva, M. Vianello, W.A. Wall, On the use of compressed polyhedral quadrature formulas in embedded interface methods, SIAM J. Sci. Comput. 39 (3) (2017) B571–B587.
[84] Y. Sudhakar, W.A. Wall, Quadrature schemes for arbitrary convex/concave volumes and integration of weak form in enriched partition of unity methods, Comput. Methods Appl. Mech. Eng. 258 (2013) 39–54.
[85] X. Wei, B. Marussig, P. Antolin, A. Buffa, Immersed boundary-conformal isogeometric method for linear elliptic problems, Comput Mech 68 (6) (2021) 1385–1405.
[86] R. Codina, G. Houzeaux, H. Coppola-Owen, J. Baiges, The fixed-mesh ALE approach for the numerical approximation of flows in moving domains, J. Comput. Phys. 228 (5) (2009) 1591–1611.
[87] J. Baiges, R. Codina, The fixed-mesh ALE approach applied to solid mechanics and fluid–structure interaction problems, Int. J. Numer. Methods Eng. 81 (12) (2010) 1529–1557.
[88] J. Baiges, R. Codina, H. Coppola-Owen, The fixed-Mesh ALE approach for the numerical simulation of floating solids, Int. J. Numer. Methods Fluids 67 (8) (2011) 1004–1023.
[89] J. Baiges, R. Codina, A. Pont, E. Castillo, An adaptive fixed-Mesh ALE method for free surface flows, Comput. Methods Appl. Mech. Eng. 313 (2017) 159–188.
[90] R. Rangarajan, A.J. Lew, Analysis of a method to parameterize planar curves immersed in triangulations, SIAM J Numer Anal 51 (3) (2013) 1392–1420.
[91] R. Rangarajan, A.J. Lew, Universal meshes: a method for triangulating planar curved domains immersed in nonconforming meshes, Int. J. Numer. Methods Eng. 98 (4) (2014) 236–264.
[92] E.S. Gawlik, A.J. Lew, High-order finite element methods for moving boundary problems with prescribed boundary evolution, Comput. Methods Appl. Mech. Eng. 278 (2014) 314–346.
[93] E.S. Gawlik, H. Kabaria, A.J. Lew, High-order methods for low Reynolds number flows around moving obstacles based on universal meshes, Int. J. Numer. Methods Eng. 104 (7) (2015) 513–538.
[94] A. Main, G. Scovazzi, The shifted boundary method for embedded domain computations. part I: Poisson and Stokes problems, J. Comput. Phys. 372 (2018) 972–995.
[95] A. Main, G. Scovazzi, The shifted boundary method for embedded domain computations. part II: linear advection–diffusion and incompressible Navier–Stokes equations, J. Comput. Phys. 372 (2018) 996–1026.
[96] T. Song, A. Main, G. Scovazzi, M. Ricchiuto, The shifted boundary method for hyperbolic systems: embedded domain computations of linear waves and shallow water flows, J. Comput. Phys. 369 (2018) 45–79.
[97] N.M. Atallah, C. Canuto, G. Scovazzi, The shifted boundary method for solid mechanics, Int. J. Numer. Methods Eng. 122 (20) (2021) 5935–5970.
[98] E.N. Karatzas, G. Stabile, N. Atallah, G. Scovazzi, G. 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, Springer, 2020, pp. 111–125.
[99] E.N. Karatzas, G. Stabile, L. Nouveau, G. Scovazzi, G. Rozza, A reduced basis approach for PDEs on parametrized geometries based on the shifted boundary finite element method and application to a Stokes flow, Comput. Methods Appl. Mech. Eng. 347 (2019) 568–587.
[100] E.N. Karatzas, G. Stabile, L. Nouveau, G. Scovazzi, G. Rozza, A Reduced-order shifted boundary method for parametrized incompressible Navier-Stokes equations, arXiv preprint arXiv:1907.10549 (2019).
[101] N.M. Atallah, C. Canuto, G. Scovazzi, Analysis of the shifted boundary method for the Stokes problem, Comput. Methods Appl. Mech. Eng. 358 (2020) 112609.
[102] N.M. Atallah, C. Canuto, G. Scovazzi, The second-generation shifted boundary method and its numerical analysis, Comput. Methods Appl. Mech. Eng. 372 (2020) 113341.
[103] N.M. Atallah, C. Canuto, G. Scovazzi, Analysis of the shifted boundary method for the poisson problem in domains with corners, Math. Comp. 90 (2021) 2041–2069.
[104] N.M. Atallah, C. Canuto, G. Scovazzi, The high-order shifted boundary method and its analysis, Comput. Methods Appl. Mech. Eng. 394 (2022) 114885.
[105] J.H. Collins, A. Lozinski, G. Scovazzi, A penalty-free shifted boundary method of arbitrary order, Comput. Methods Appl. Mech. Eng. (2023) 116301.
[106] C.-H. Yang, K. Saurabh, G. Scovazzi, C. Canuto, A. Krishnamurthy, B. Ganapathysubramanian, Optimal surrogate boundary selection and scalability studies for the shifted boundary method on octree meshes, Comput. Methods Appl. Mech. Eng. 419 (2024) 116686.
[107] C.-H. Yang, G. Scovazzi, A. Krishnamurthy, B. Ganapathysubramanian, Octree-based adaptive mesh refinement and the shifted boundary method for efficient fluid dynamics simulations, Advances in Computational Science and Engineering 4 (2025) 57–84.
[108] C.-H. Yang, G. Scovazzi, A. Krishnamurthy, B. Ganapathysubramanian, Simulating incompressible flows over complex geometries using the shifted boundary method with incomplete adaptive octree meshes, J. Comput. Phys. (2025). (published online, https://doi.org/10.1016/j.jcp.2025.114334).
[109] C.-H. Yang, G. Scovazzi, A. Krishnamurthy, B. Ganapathysubramanian, A shifted boundary method for thermal flows, J. Comput. Phys. (2025). (accepted, http://dx.doi.org/10.2139/ssrn.5197116).
[110] K. Li, N.M. Atallah, G.A. Main, G. Scovazzi, The shifted interface method: a flexible approach to embedded interface computations, Int. J. Numer. Methods Eng. 121 (3) (2020) 492–518. https://doi.org/10.1002/nme.6231
[111] K. Li, N.M. Atallah, A. Rodríguez-Ferran, D.M. Valiveti, G. Scovazzi, The shifted fracture method, Int. J. Numer. Methods Eng. 122 (22) (2021) 6641–6679.
[112] K. Li, A. Rodríguez-Ferran, G. Scovazzi, A blended shifted-Fracture/phase-Field framework for sharp/diffuse crack modeling, Int. J. Numer. Methods Eng. 124 (4) (2023) 998–1030.

[113] K. Li, A. Rodríguez-Ferran, G. Scovazzi, The simple shifted fracture method, Int. J. Numer. Methods Eng. 124 (12) (2023) 2837–2875.
[114] K. Li, A. Rodríguez-Ferran, G. Scovazzi, Crack branching and merging simulations with the shifted fracture method, Comput. Methods Appl. Mech. Eng. 433 (2025) 117528.
[115] O. Colomés, A. Main, L. Nouveau, G. Scovazzi, A weighted shifted boundary method for free surface flow problems, J. Comput. Phys. 424 (2021) 109837.
[116] D. Xu, O. Colomés, A. Main, K. Li, N.M. Atallah, N. Abboud, G. Scovazzi, A weighted shifted boundary method for immersed moving boundary simulations of Stokes flow, J. Comput. Phys. 510 (2024) 113095. https://www.sciencedirect.com/science/article/pii/S0021999124003449. https://doi.org/10.1016/j.jcp.2024.113095
[117] H. Dütsch, F. Durst, S. Becker, H. Lienhart, Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan-Carpenter numbers, J Fluid Mech 360 (1998) 249–271. https://doi.org/10.1017/S002211209800860X
[118] L. Wang, Z.L. Guo, J.C. Mi, Drafting, kissing and tumbling process of two particles with different sizes, Computers & Fluids 96 (2014) 20–34. https://www.sciencedirect.com/science/article/pii/S0045793014001042. https://doi.org/10.1016/j.compfluid.2014.03.005
[119] M. Olshanskii, H. von Wahl, Stability of instantaneous pressures in an Eulerian finite element method for moving boundary flow problems, arXiv preprint arXiv:2412.17657 (2024).
[120] N. Atallah, C. Canuto, G. Scovazzi, Analysis of the weighted shifted boundary method for the Poisson and Stokes problems, Computers & Mathematics with Applications (submitted, http://dx.doi.org/10.2139/ssrn.5183772) (2025).
[121] P. Dubey, P. Hanrahan, R. Fedkiw, M. Lentine, C. Schroeder, PhysBAM: physically based simulation, in: ACM SIGGRAPH 2011 Courses, ACM, 2011, p. 10.
[122] K. Wang, J. Grétarsson, A. Main, C. Farhat, Computational algorithms for tracking dynamic fluid–structure interfaces in embedded boundary methods, Int J Numer Methods Fluids 70 (4) (2012) 515–535.
[123] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Comput. Methods Appl. Mech. Eng. 197 (1–4) (2007) 173–201.
[124] O. Colomés, S. Badia, R. Codina, J. Principe, Assessment of variational multiscale models for the large eddy simulation of turbulent incompressible flows, Comput. Methods Appl. Mech. Eng. 285 (2015) 32–63.
[125] T.T.P. Team, The Trilinos Project Website, 2020.
[126] E.C. Cyr, J.N. Shadid, R.S. Tuminaro, Stabilization and scalable block preconditioning for the Navier–Stokes equations, J. Comput. Phys. 231 (2) (2012) 345–363.
[127] O. Colomés, S. Badia, J. Principe, Mixed finite element methods with convection stabilization for the large eddy simulation of incompressible turbulent flows, Comput. Methods Appl. Mech. Eng. 304 (2016) 294–318.
[128] O. Colomés, S. Badia, Segregated Runge–Kutta time integration of convection-stabilized mixed finite element schemes for wall-unresolved LES of incompressible flows, Comput. Methods Appl. Mech. Eng. 313 (2017) 189–215.
[129] J. Pérez-Barrera, A. Gómez-Ortega, M.I. Tenorio-Suárez, K. Corrales-Camacho, S. Piedra, C. Félix-Martínez, Version [2.0]- [MaSMaker: An open-source, portable software to create and integrate maze-like surfaces into arbitrary geometries], SoftwareX 26 (2024) 101683. https://www.sciencedirect.com/science/article/pii/S2352711024000542. https://doi.org/10.1016/j.softx.2024.101683
[Séquin] C. Séquin, Scherk-Collins sculpture generator, (https://people.eecs.berkeley.edu/~sequin/SCULPTS/scherk.html).
[Séquin] C. Séquin, Part Description and Specifications for "Monkey Trefoil", (https://people.eecs.berkeley.edu/~sequin/SFF/spec.monktref.html).
[132] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Eng. 32 (1–3) (1982) 199–259.
[133] T.J.R. Hughes, G. Scovazzi, L.P. Franca, Multiscale and stabilized methods, in: E. Stein, R. de Borst, T.J.R. Hughes (Eds.), Encyclopedia of Computational Mechanics, John Wiley & Sons, Ltd, 2004.