Computer Methods in Applied Mechanics and Engineering 452 (2026) 118748

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

The generalized shifted boundary method for geometry-parametric PDEs and time-dependent domains

Oriol Colomés [a,*], Jan Modderman [a], Guglielmo Scovazzi [b]

[a] Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, 2628 CN, The Netherlands
[b] Department of Civil and Environmental Engineering, Duke University, Durham, NC 27708, USA

ARTICLE INFO

Keywords:
Generalized shifted boundary method
Unfitted finite element method
Immersed boundary method
Moving geometry
Parametrized geometry

ABSTRACT

Many engineering and scientific problems require the solution of partial differential equations in complex geometries. Often, these problems involve parametrized geometries, e.g. design optimization, or moving domains, e.g. fluid-structure interaction problems. For such cases, traditional methods based on body-fitted grids require time-consuming mesh generation or re-meshing techniques. Unfitted finite element methods, e.g. CutFEM of AgFEM, are appealing techniques that address these challenges. However, they require ad-hoc integration methods and stabilization techniques to prevent instabilities for small cut cells. Recently, the Shifted Boundary Method (SBM), was introduced to prevent integration over cut cells and small cut-cell instabilities. An extension of the SBM was recently introduced, the Weighted Shifted Boundary Method (WSBM), where the variational form is weighted by the elemental active volume fraction, improving discrete mass/momentum conservation properties in simulations with moving domains. In this work we introduce the Generalized Shifted Boundary Method (GSBM), a geometry-agnostic generalization of the SBM and WSBM formulations that avoids the need of redefinition of integration domains and finite element spaces. The GSBM enables a unified formulation for problems with evolving geometries, supports gradient-based optimization of problems with varying geometries including topological changes, and unifies SBM, WSBM, and optimal-surrogate variants within a single framework. In this work we describe the formulation, and corresponding tests, for three model problems, namely: the Poisson problem, linear elasticity and transient Stokes flow.

1. Introduction

Many scientific and engineering problems deal with the solution of partial differential equations (PDEs) on domains with complex geometries, geometries that move or evolve in time, or with parametrized geometries. That is the case, for example, of blood flow simulations in patient-specific geometries [1], ice accretion on airplane wings [2], shape optimization in structural design [3], or hydrofoils moving under the effect of waves [4], just to name a few. In such applications, the use of immersed methods, also denoted as unfitted or embedded methods, is appealing. In such approaches, the need of generating geometry-fitted meshes is bypassed by embedding the solution domain into a non-fitting background grid encompassing the domain of interest. These types of methods have been successfully applied to a wide variety of problems, namely: flow around complex geometries [5–9], flow around moving

* Corresponding author.
E-mail addresses: j.o.colomesgene@tudelft.nl (O. Colomés), j.modderman@tudelft.nl (J. Modderman), guglielmo.scovazzi@duke.edu (G. Scovazzi).

https://doi.org/10.1016/j.cma.2026.118748
Received 8 August 2025; Received in revised form 29 October 2025; Accepted 8 January 2026
Available online 15 January 2026

domains [10–15], fluid-structure interaction problems [16–25], elastodynamic problems in complex geometries [26–30], fracture or interface problems [31–35], topology optimization [36–38] or additive manufacturing [39,40], among others.

The use of unfitted finite element (FE) methods is appealing for problems in complex and/or moving geometries, but it presents a number of challenges. Note that in this work we focus on unfitted FE methods, but similar challenges are faced when using other methodologies, e.g. immersed boundary methods in finite difference or finite volume schemes. The majority of unfitted FE methods rely on the evaluation of the weak form on cut cells. That is the case, for example, of the CutFEM approach [41], the Aggregated FE method (AgFEM) [42], or the Finite Cell Method (FCM). In such approaches, the integration on cut cells can lead to instabilities or ill-conditioning of the discrete system due to the small cut-cell problem [43]. Different strategies can be followed to prevent these instabilities; the CutFEM approach introduces ghost penalty stabilization [44], which consists of adding terms in the weak form that penalize the jump on normal gradients in the faces close to the embedded boundary. On the other hand, the AgFEM approach introduces an extension operator based on aggregation of cut cells with interior cells, providing stability of the solution without the need to modify the weak form resulting from a conformal mesh. It has been shown that both approaches are equivalent under a certain particular definition of the ghost penalty, see [45]. Both CutFEM and AgFEM methods have been successfully applied to a wide variety of problems and demonstrated their capabilities also in large-scale problems with parallelized implementations [46–49], space-time formulations [50,51] or using high-order finite element spaces [52–54]. Although these methods have been successfully used in many applications, they still rely on the integration on cut cells, which becomes a challenge. Different strategies can be pursued to integrate on cut cells: reconstruction of the cut cell via tessellation [55], adaptive quadrature rules [26], or moment fitted quadrature rules [56]. All these cut-cell integration methodologies introduce a significant extra computational cost. In addition, the AgFEM approach requires the definition of ad-hoc FE spaces with cell aggregates, which is both time-consuming and implementation-intensive. These challenges are accentuated when dealing with moving geometries or parametrized geometries, where all the geometrical preprocessing and integration data structure definition have to be done for every parametric evaluation or at every time step, or even nonlinear iteration if dealing with nonlinear problems.

In contrast with the described unfitted FE methods, the Shifted Boundary Method (SBM) [5,6] alleviates the need of having to integrate on cut-cells and/or define ad-hoc FE spaces. The SBM relies on the definition of a surrogate domain composed of full cells only, applying the boundary conditions on a *shifted* surrogate boundary composed of the closest non-intersected edges to the embedded boundary. Then, a modified boundary condition is enforced at the surrogate boundary through the use of a Taylor extrapolation. During the past years the SBM has been applied to a wide variety of problems, namely: Poisson and Stokes problems [5], advection-diffusion and Navier-Stokes [6], compressible Euler [57], shallow water equations [58], Lagrangian shock hydrodynamics [59] or elasticity problems [29,60,61]. A number of extensions to the SBM have been proposed in the literature, including efficient high-order formulations [62–64], applications in isogeometric analysis [65], definition of reduced order models [66,67], octree-based refinement strategies [68], or solution of interface, fracture and contact mechanics problems [35,69–72].

Despite being applied in several problems, one of the main issues of the SBM arises from the lack of mass conservation in problems with moving domains. This issue manifests in the results by the appearance of spurious pressure oscillations in time. This is a shared problem in sharp-interface immersed boundary methods [73,74] and unfitted FE methods for problems with moving domains. In a recent work, Olshanskii and Wahl [75] study this phenomenon and propose a remedy for the CutFEM approach. In this later work the authors propose a global definition of the ghost penalty, which introduces a consistency error, but preserves optimal order of accuracy and ensures pressure stability in time. Also, with the objective of reducing spurious pressure oscillations in time, a variant of the SBM approach was introduced in [76], denoted as Weighted Shifted Boundary Method (WSBM). The WSBM proposes a weighted variational formulation, where the weight factors are the volume fraction of fluid in every element. The resulting formulation does not require integration in cut cells and preserves the total volume of active fluid to a much higher degree of accuracy than the SBM, and, consequently, reduces the pressure oscillations in time. The WSBM method was initially proposed for the solution of free surface flows in [76], and recently extended to the solution of problems with moving domains in Stokes flow [77] and incompressible Navier-Stokes [78].

A common characteristic of all unfitted FE methods discussed until this point shares one additional challenge: the necessity to re-define degrees of freedom, associated FE spaces and data structures, and integration quadratures every time that the geometry changes. These considerations are particularly relevant in the case of moving geometries or shape/topology optimization. Our work precisely addresses this challenge, by introducing the Generalized Shifted Boundary Method (GSBM), which generalizes the WSBM [76].

In the GSBM, the formulation is applied over the entire background grid in which geometric shapes are immersed. This is a key novel point of the proposed approach, which enables the use of a unique FE space that does not depend on the geometry of the problem. The ghost-penalty stabilization is used to complement the standard weak form in regions where the governing equations are not defined. Because all nodes/elements of the background grid are active at all times in the simulation, the GSBM is a geometry-agnostic finite element method, and its main benefits are:

a) The GSBM provides a geometry-agnostic FE formulation that does not require re-definition of FE spaces, integration rules, or definition of geometrical entities. These features are novel with respect to the original SBM and WSBM formulations, where explicit definitions of surrogate boundaries and integration domains have to be identified a priori.
b) The GSBM formulation enables differentiation with respect to degrees of freedom, even for moving domains or evolving geometries with topological changes. Since the formulation uses a unique FE space defined on the background domain, independent of the embedded geometry, one can use differentiation techniques, e.g. automatic differentiation, to compute derivatives of certain quantities of interest with respect to the unknown. This feature is specially relevant for gradient-based topology optimization

[38] or reduced order models for variable parametric or non-parametric geometries [79,80]. However, specific research on the use of the differentiability of the proposed GSBM formulation is beyond the scope of this work and will be the object of future developments.

c) The same GSBM formulation enables the recovery of the original SBM formulation [5], the SBM formulation with optimal surrogate boundaries [68] or the WSBM formulation [77].

This paper is organized as follows: in Section 2 we define the problem setting, notation and general definitions that will be later used in the definition of the GSBM formulation. The GSBM formulation is later introduced in Section 3, where we discuss the equivalence with the original SBM and WSBM formulations and present the specific formulation for three model problems, namely Poisson, linear elasticity and transient Stokes problem. In Section 4 we present a set of numerical tests where we demonstrate the capabilities of the proposed GSBM formulation for manufactured solutions of the Poisson, linear elasticity and transient Stokes problems. We conclude the work with final remarks in Section 5.

2. Problem setting and definitions

In this section, we introduce the notation and we define the general problem setting that will be later used to develop the Generalized Shifted Boundary Method framework. Let us consider a PDE in a time-dependent domain $\Omega(t)$, also denoted as the physical domain hereinafter, defined by the following strong form:

$$\partial_t u - \nabla \cdot \boldsymbol{\sigma}(u) = f \qquad \text{in } \Omega(t), \tag{1a}$$

$$u = u_D \qquad \text{on } \Gamma_D(t), \tag{1b}$$

$$u = u_0 \qquad \text{on } \Omega(0), \tag{1c}$$

where $\boldsymbol{\sigma}$ is a given differential operator that depends on the physical problem at hand and u a set of scalar or vectorial unknown variables. In this work we assume that the solution to Eq. (1a) satisfies a Dirichlet boundary condition on the full domain boundary, *i.e.* $\Gamma_D(t) = \partial\Omega(t)$, with time-dependent boundary condition $u_D(t)$. We note that the formulation presented in this manuscript can be extended to a combined Dirichlet-Neumann problem or a full Neumann boundary condition problem. However, for the sake of simplicity we restrict ourselves to the pure Dirichlet case. Note also that the same approach applies to constant geometry and/or boundary conditions.

In this work we focus on the formulation of a problem in a time-dependent domain $\Omega(t)$. However, the same approach could be applied to a domain, $\Omega(\boldsymbol{\mu})$, parametrized by the set of parameters $\boldsymbol{\mu}$. Furthermore, the formulation could also be extended to time-dependent parametrized geometries, i.e. $\Omega(t, \boldsymbol{\mu})$. For the sake of simplicity, but without loss of generality, we will not consider the latter two cases.

For clarity, we consider equations that can be expressed as in (1a), such as in the Poisson problem where $\boldsymbol{\sigma}(u) \overset{\text{def}}{=} \nabla u$, the linear elasticity problem where $\boldsymbol{\sigma}(\boldsymbol{u}) \overset{\text{def}}{=} 2\mu\varepsilon(\boldsymbol{u}) + \lambda(\nabla \cdot \boldsymbol{u})\boldsymbol{I}$, or in the Stokes problem where $\boldsymbol{\sigma}(\boldsymbol{u}, p) \overset{\text{def}}{=} 2\mu\varepsilon(\boldsymbol{u}) - p\boldsymbol{I}$. However, more general definitions can also be considered with the proposed methodology, e.g. Navier-Stokes or nonlinear elasticity. The later are excluded from the scope of this work since the focus is on the definition of the SBM generalization rather than on the solution of complex systems.

We introduce next some notation that will be used in the definition of the proposed GSBM. Let us define a background closed domain D such that it entirely encloses the physical domain $\Omega(t)$ for all time instances t in a given time period $t \in [0, T]$. To simplify notation, in what comes we will drop the time dependency, referring to $\Omega \equiv \Omega(t)$, even if the domain changes in time. Using this notation, we have that Ω is fully immersed in D: $\text{clos}(\Omega) \subset D$, see Fig. 1a. We denote by $\mathcal{T}^h$ a conformal and shape-regular tesselation of the background domain D, fixed in time. Note that $\mathcal{T}^h$ can be a generic unstructured grid, but for efficiency purposes, it is often constructed by a regular Cartesian grid. We denote by $\mathcal{E}^h$ the set of element interfaces of dimension $d-1$ interior to the background tesselation $\mathcal{T}^h$, with d being the topological dimension of the domain D. In other words, $\mathcal{E}^h$ is the set of all mesh edges in a two-dimensional domain or the set of mesh faces in a three-dimensional domain, see Fig. 1b. We denote by Ω^h the discrete integration domain defined by all the elements in the tesselation $\mathcal{T}^h$, i.e. $\Omega^h \overset{\text{def}}{=} \text{int}\left(\cup_{K\in\mathcal{T}^h} K\right)$. Note that $\Omega^h \equiv D$ by construction.

We denote by $L^r(\Omega)$, $1 \leq r < \infty$, the spaces of functions such that their r-th power is Lebesgue-integrable in Ω. For the case in which $r = 2$, we have a Hilbert space with an inner product

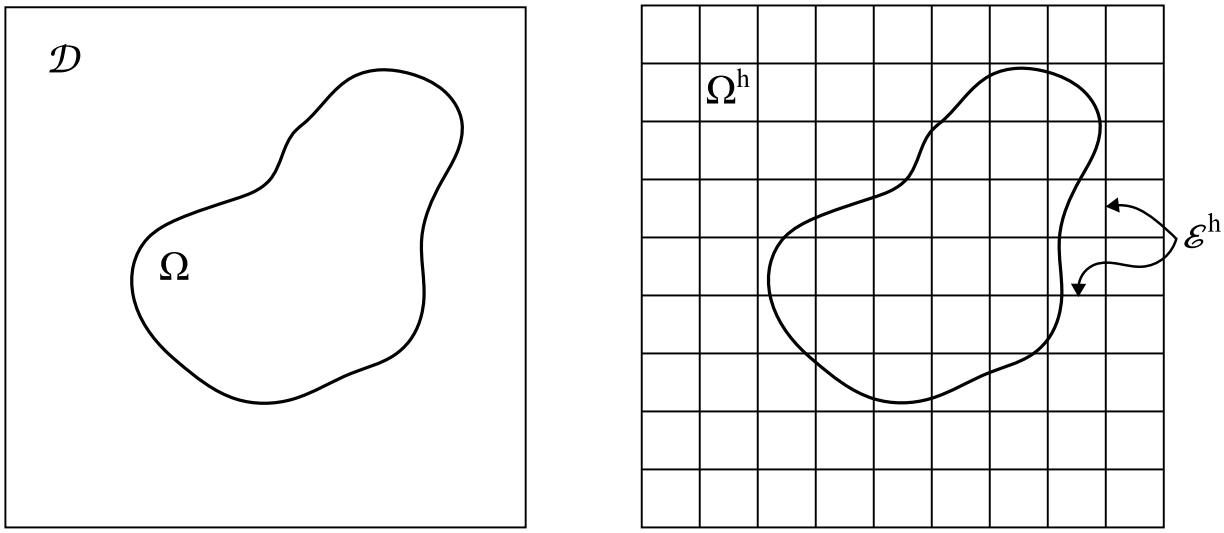
$$(u, v)_\Omega = \int_\Omega u\, v\, d\Omega, \tag{2}$$

and induced norm $\|u\|_{L^2(\Omega)} \overset{\text{def}}{=} \|u\|_\Omega = (u, u)_\Omega^{1/2}$. With a slight abuse of notation, (2) is used to define the integral of the product of two functions, even if they are not in $L^2(\Omega)$, and for both scalar and vector fields. The space of functions whose distributional derivatives up to order m are in $L^2(\Omega)$ are denoted by $H^m(\Omega)$. We will focus on the case of $m = 1$, which is also a Hilbert space.

In a discrete setting, for a given discrete domain Ω^h, the inner product equivalent to Eq. (2) will be given by

$$(u, v)_{\Omega^h} = \sum_{K\in\mathcal{T}^h} \left(\int_K u\, v\, dK \right). \tag{3}$$

With all these definitions, one can derive the weak form of the original problem in the continuous setting, using Nitsche's method [81] to enforce Dirichlet boundary conditions. With the notation defined previously, and assuming a scalar field as unknown u, the



(a) Background domain, $\mathcal{D}$, enclosing the physical domain, Ω.

(b) Discrete background domain, Ω^h, and set of interior edges, $\mathcal{E}^h$.

Fig. 1. Sketch of background domain, $\mathcal{D}$, and physical domain, Ω, (left) and background triangulation, Ω^h (right).

weak form of the model problem (1) reads: *find* $u \in \mathcal{V} \subset H^1(\Omega)$ *such that*

$$\left(v, \partial_t u\right)_\Omega + \left(\nabla v, \boldsymbol{\sigma}(u)\right)_\Omega - \left(v, \boldsymbol{\sigma}(u)\cdot\boldsymbol{n}\right)_{\Gamma_D} - \left(\boldsymbol{\sigma}(v)\cdot\boldsymbol{n}, u-u_D\right)_{\Gamma_D} + \left(\beta v, u-u_D\right)_{\Gamma_D} = \left(v, f\right)_\Omega \qquad \forall v \in \mathcal{V} \tag{4}$$

almost everywhere in time and satisfying the initial condition (1c) in a weak sense. In Eq. (4), β is a constant that in the discrete setting typically depends on the element size.

3. The generalized shifted boundary method

3.1. Shifted boundary and weighted finite element space

Before defining the Generalized Shifted Boundary (GSBM) formulation in Section 3.2, we introduce some ingredients that will be used later on. That is, the concepts of *shifted boundary* condition and *weighted finite element space*.

The *shifted boundary* condition, initially proposed in [82], is summarized next. Let us denote by $\boldsymbol{d}$ the distance vector that maps any point in the background domain $\boldsymbol{x} \in D$ to the nearest point of the boundary $\boldsymbol{x}_\Gamma \in \Gamma_D$.

$$\boldsymbol{d}(\boldsymbol{x}) \overset{\text{def}}{=} \boldsymbol{x}_\Gamma - \boldsymbol{x} = \left[\mathcal{M}_\Gamma - \mathcal{I}\right](\boldsymbol{x}), \tag{5}$$

where $\mathcal{M}_\Gamma$ is the closest projection map from any point in D to Γ_D and $\mathcal{I}$ the identity map. In particular,

$$\begin{aligned} \mathcal{M}_\Gamma : D &\longrightarrow \Gamma_D, \\ \boldsymbol{x} &\mapsto \boldsymbol{x}_\Gamma, \end{aligned} \tag{6}$$

and we are interested in the discrete version of this map, denoted by $\mathcal{M}_\Gamma^h(\boldsymbol{x})$, defined for all points in the background triangulation $\boldsymbol{x} \in \mathcal{T}^h$. Computational geometry algorithms are available to directly evaluate $\boldsymbol{d}$ by closest point projection. Otherwise, if the physical domain Ω is defined by a level set function $\phi(\boldsymbol{x})$, the distance $\boldsymbol{d}$ can be computed as

$$\boldsymbol{d} = \phi \frac{\nabla\phi}{\|\nabla\phi\|}. \tag{7}$$

In practice, it is enough to define the distance only on the facets where $[\![\alpha_h]\!] > 0$, see Eq. (12) for the definition of α and Eq. (16) for the definition of $[\![\cdot]\!]$. This set of facets is in a narrow band of elements in $\mathcal{T}^h$ that intersect with Γ_D. Once the distance vector $\boldsymbol{d}$ is known, we can define a *shifted boundary* condition at a given point $\boldsymbol{x}_0$ close to the Dirichlet boundary Γ_D as

$$S_{\boldsymbol{d}}(u)\big|_{\boldsymbol{x}_0} = u_D, \tag{8}$$

with

$$S_{\boldsymbol{d}}(u)\big|_{\boldsymbol{x}_0} \overset{\text{def}}{=} u(\boldsymbol{x}_0) + \nabla u(\boldsymbol{x}_0)\cdot\boldsymbol{d}(\boldsymbol{x}_0) \tag{9}$$

being the truncated Taylor expansion of the solution u along the distance vector $\boldsymbol{d}$. When using higher-order FE spaces, this expansion should include higher-order terms, see [63].

The concept of *weighted finite element space*, initially proposed in [76], is described next. Let us define the FE space $\mathcal{V}_h$ as the space of continuous piece-wise functions defined by elemental Lagrange polynomials of order r in quadrilateral/hexahedral elements, $\mathbb{Q}^r$, that is

$$\mathcal{V}_h \overset{\text{def}}{=} \left\{ v \in C^0(\Omega^h) : v_K \in \mathbb{Q}^r(K),\ \forall K \in \mathcal{T}^h \right\}. \tag{10}$$

In this paper we work with quadrilateral/hexahedral elements, but the same formulation can be directly applied to triangular/tetrahedral elements, i.e. $\mathbb{P}^r$. Let us also define a Heaviside function as

$$H(\boldsymbol{x}) \overset{\text{def}}{=} \begin{cases} 1.0 & \text{if } \boldsymbol{x} \in \Omega, \\ 0.0 & \text{otherwise.} \end{cases} \tag{11}$$

Using the Heaviside function (11), we can define a piece-wise constant field, α_h, in the triangulation $\mathcal{T}^h$ given by

$$\alpha_h \overset{\text{def}}{=} \left\{ \left. \frac{\int_K H(\boldsymbol{x})\, dK}{|K|} \right|_K , \forall K \in \mathcal{T}^h \right\}. \tag{12}$$

Where $|K|$ is the measure of a given element K of the triangulation $\mathcal{T}^h$. The field $\alpha_h \in [0, 1]$ can be interpreted as the active volume fraction of each element in the triangulation, equivalent to the element fraction of $K \in \mathcal{T}^h$ belonging to the physical domain Ω. Note also that $\sum_{K \in \mathcal{T}^h} \alpha_h |K| \approx |\Omega|$, that is, the sum of weighted elemental volumes approximates the total measure of the physical domain Ω.

Remark 1. The proposed formulation does not require the elemental integral $\int_K H(\boldsymbol{x})\, dK$ appearing in Eq. (12) to be computed exactly, avoiding the need for cell tessellation operations or high-order quadrature rules. In practice, we just need a good enough approximation that does not pollute the intrinsic numerical error associated with the choice of FE spaces.

We now can define the *α-weighted finite element space* $\mathcal{V}_h^\alpha$ as the space of discontinuous piece-wise functions defined by the set of functions in $\mathcal{V}_h$ weighted by the piece-wise constant field α_h as defined in Eq. (12):

$$\mathcal{V}_h^\alpha \overset{\text{def}}{=} \left\{ v_\alpha \in L^2(\Omega^h) : v_\alpha = \alpha_h v_h,\ v_h \in \mathcal{V}_h \right\}. \tag{13}$$

3.2. The GSBM formulation

To derive the weak form of the GSBM, we multiply the strong form (1a) by a test function $v_\alpha \in \mathcal{V}_h^\alpha$ and enforce a *shifted* boundary condition (8) following Nitsche's approach, adding a ghost penalty term [44]. This results in the following Petrov-Galerkin form: *find* $u_h \in \mathcal{V}_h$ *such that*

$$\begin{aligned} &\left(v_\alpha, \partial_t u_h\right)_{\Omega^h} + \left(\nabla v_\alpha, \boldsymbol{\sigma}(u_h)\right)_{\Omega^h} - \left([\![v_\alpha]\!], \{\!\{ \boldsymbol{\sigma}(u_h) \}\!\}_\gamma \right)_{\mathcal{E}^h} - \left(\{\!\{ \boldsymbol{\sigma}(v_h) \}\!\}_\gamma, [\![\alpha_h]\!] \{\!\{ S_{\boldsymbol{d}}(u_h) - u_D \}\!\}_\gamma \right)_{\mathcal{E}^h} + \\ &\left(\frac{\beta_1}{h} |[\![\alpha_h]\!]| \{\!\{ S_{\boldsymbol{d}}(v_h) \}\!\}_\gamma, \{\!\{ S_{\boldsymbol{d}}(u_h) - u_D \}\!\}_\gamma \right)_{\mathcal{E}^h} + j(u_h, v_h) = (v_\alpha, f)_{\Omega^h} \qquad \forall v_\alpha \in \mathcal{V}_h^\alpha. \end{aligned} \tag{14}$$

The term $j(u_h, v_h)$ is the ghost penalty term, defined as

$$j(u_h, v_h) \overset{\text{def}}{=} \sum_{l=1}^{r} \left(\beta_2(\alpha_h) h^{2l-1} [\![D_{\boldsymbol{n}}^l v_h]\!], [\![D_{\boldsymbol{n}}^l u_h]\!] \right)_{\mathcal{E}^h}, \tag{15}$$

with $D_{\boldsymbol{n}}^l$ the lth partial derivative along the face normal direction $\boldsymbol{n}$, for $l = 1, \dots, r$ where r is the polynomial order of the finite element space.

Since the space of test functions $\mathcal{V}_h^\alpha$ is discontinuous, we have a Petrov-Galerkin formulation with discontinuous fields across facets. This leads to the presence of facet integrals containing jump and mean operators, see the third to sixth terms in Eq. (14). Here we use the following definitions:

$$[\![\cdot]\!] \overset{\text{def}}{=} (\cdot)^+ \boldsymbol{n}^+ + (\cdot)^- \boldsymbol{n}^-, \qquad \{\!\{ \cdot \}\!\}_\gamma \overset{\text{def}}{=} \gamma (\cdot)^+ + (1 - \gamma)(\cdot)^-, \tag{16}$$

where $\boldsymbol{n}$ is the normal to a given facet of the triangulation skeleton $\mathcal{E}^h$ and

$$\gamma \overset{\text{def}}{=} \alpha_h^+ / \max(\alpha_h^+ + \alpha_h^-, tol).$$

Here $tol > 0$ is an arbitrarily small number used to prevent $\gamma \to \infty$ when $\alpha_h^+ = \alpha_h^- = 0.0$. In Eq. (14), h is the characteristic element size, $\beta_1 > 0$ is an algorithmic constant used to tune the penalty term in the Nitsche's method and $\beta_2(\alpha)$ a parameter for the ghost penalty term [44] defined as

$$\beta_2(\alpha) \overset{\text{def}}{=} \begin{cases} \kappa & \text{if} \{\!\{ \alpha \}\!\} < 1.0 \text{and} |[\![\alpha]\!]| < 1.0, \\ 0.0 & \text{otherwise.} \end{cases} \tag{17}$$

With $\kappa > 0$ an algorithmic constant, which is set to $\kappa = 0.5$ in this work, or otherwise stated.

Remark 2. Note that β_1 and the algorithmic constant κ appearing in Eq. (17) might also depend on the problem properties; we refer to Section 3.5 for specific definitions of these parameters.

Here it is important to highlight that the last term in (14), $j(u_h, v_h)$, acts as a ghost penalty in the edges belonging to the cut elements and as an extension operator that extends the solution from Ω to $D \setminus \Omega$. The need for the ghost penalty term arises from the fact that the weighting field α_h can be arbitrary small, leading to vanishing integrals in cells with small active volume. Thus, we introduce the ghost penalty stabilization [44] to prevent instabilities and ill-conditioning of the system, and use it as an extension operator into the inactive region. We refer the reader to [83] for the formal analysis of the original WSBM method. It is also important to note that the presented formulation assumes $r = 1$ in (10). For higher-order polynomials, higher-order derivatives should be penalized, see [44].

Remark 3. The definition of β_2 proposed in Eq. (17) ensures that the ghost penalty term is not enforced in facets between two fully active cells, i.e. $\alpha_h^+ = \alpha_h^- = 1.0$. An alternative definition could be

$$\beta_2(\alpha) \overset{\text{def}}{=} \kappa, \qquad \forall \alpha,$$

leading to a global ghost penalty approach as proposed by Olshanskii in [75] et al. This definition might be beneficial to avoid spurious pressure oscillations in incompressible flow problems around moving domains. See also Section 4.3.2.

The geometry-parametric weak form. The GSBM formulation described in (14) defines a geometry-parametric finite element discretization for arbitrary geometries, parametrized by the fields $\boldsymbol{d}$ and α. Therefore, we can rewrite the GSBM problem as: *for a given domain $\Omega \subset \Omega^h$ characterized by the active volume fraction, α_h defined in Eq. (12), and closest distance to the boundary $\partial\Omega$, $\boldsymbol{d}$, find the solution $u_h \in \mathcal{V}_h$ such that:*

$$B_h\big(\alpha_h, \boldsymbol{d}; (u_h, v_h)\big) = L_h\big((\alpha_h, \boldsymbol{d}; v_h\big) \qquad \forall v_h \in \mathcal{V}_h. \tag{18}$$

The geometry-parametric bilinear form $B_h\big(\alpha_h, \boldsymbol{d}; (u_h, v_h)\big)$ and the linear form $L_h\big(\alpha_h, \boldsymbol{d}; v_h\big)$ are given by

$$\begin{aligned} B_h\big(\alpha_h, \boldsymbol{d}; (u_h, v_h)\big) =& \big(v_\alpha, \partial_t u_h\big)_{\Omega^h} + \big(\nabla v_\alpha, \boldsymbol{\sigma}(u_h)\big)_{\Omega^h} - \Big(\llbracket v_\alpha \rrbracket, \big\{\!\!\big\{ \boldsymbol{\sigma}(u_h) \big\}\!\!\big\}_\gamma\Big)_{\mathcal{E}^h} - \\ & \Big(\big\{\!\!\big\{ \boldsymbol{\sigma}(v_h) \big\}\!\!\big\}_\gamma, \llbracket \alpha_h \rrbracket \big\{\!\!\big\{ S_{\boldsymbol{d}}(u_h) - u_D \big\}\!\!\big\}_\gamma\Big)_{\mathcal{E}^h} + \\ & \left(\frac{\beta_1}{h} |\llbracket \alpha_h \rrbracket| \big\{\!\!\big\{ S_{\boldsymbol{d}}(v_h) \big\}\!\!\big\}_\gamma, \big\{\!\!\big\{ S_{\boldsymbol{d}}(u_h) - u_D \big\}\!\!\big\}_\gamma\right)_{\mathcal{E}^h} + j(u_h, v_h) \end{aligned} \tag{19}$$

$$L_h\big(\alpha_h, \boldsymbol{d}; v_h\big) = (v_\alpha, f)_{\Omega^h}, \tag{20}$$

where we recall that $v_\alpha = \alpha_h v_h$ and $j(u_h, v_h)$ is given by Eq. (15).

Observe that the solution to problem (18) is defined in a FE space, $\mathcal{V}_h$, that does not depend on the geometry of the domain, Ω. The geometry and boundary conditions are implicitly represented in the formulation by the fields α_h and $\boldsymbol{d}$. This leads to a geometry-parametric formulation where the degrees of freedom are fixed given a unique background grid. Since the degrees of freedom of the problem are fixed for any embedded geometry in the background domain, $\Omega \subset D$, one can derive quantities of interest with respect to the degrees of freedom, even when the domain changes in time.

3.3. Equivalence with the original shifted boundary method

Let us define an alternative definition of the weighting field α_h, denoted as $\tilde{\alpha}_h$, given by

$$\tilde{\alpha}_h = \left\{ \text{floor}\left(\frac{\int_K H(\boldsymbol{x})\, dK}{|K|} \right)\Bigg|_K, \forall K \in \mathcal{T}^h \right\}. \tag{21}$$

That is, $\tilde{\alpha}_h|_K = 1.0$ if $\int_K H(\boldsymbol{x})\, dK = |K|$ and $\tilde{\alpha}_h|_K = 0.0$ otherwise, for all $K \in \mathcal{T}^h$. In other words, we consider $\tilde{\alpha}_h|_K = 1.0$ if a cell is interior to Ω and not cut by $\partial\Omega$, and $\tilde{\alpha}_h|_K = 0$ otherwise. Later in the results section, we will denote this case as *binary* definition of α or *SBM* equivalent. The original Shifted Boundary Method (SBM) formulation is defined only on the set of elements where $\tilde{\alpha}_h = 1.0$, which defines a surrogate interior domain $\tilde{\Omega}^h$ and the corresponding surrogate boundary $\tilde{\Gamma}^h \equiv \partial\tilde{\Omega}^h$, see Fig. 2a. That is,

$$\tilde{\mathcal{T}}^h \overset{\text{def}}{=} \big\{ K \in \mathcal{T}^h : K \subset \text{clos}(\Omega) \big\}, \qquad \tilde{\Omega}^h \overset{\text{def}}{=} \text{int}\big(\cup_{K \in \tilde{\mathcal{T}}^h} K\big) \subseteq \Omega.$$

Let us define the FE space on the surrogate domain $\tilde{\Omega}^h$ as

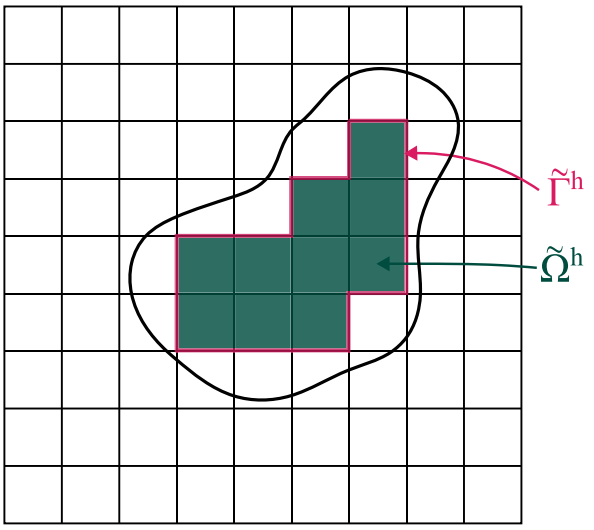
$$\tilde{\mathcal{V}}_h = \big\{ v \in C^0(\tilde{\Omega}^h) : v_K \in \mathbb{Q}^r(K),\ \forall K \in \tilde{\mathcal{T}}^h \big\}. \tag{22}$$

With these definitions, the SBM formulation for a pure Dirichlet boundary problem reads: *find $u_h \in \tilde{\mathcal{V}}_h$ such that*

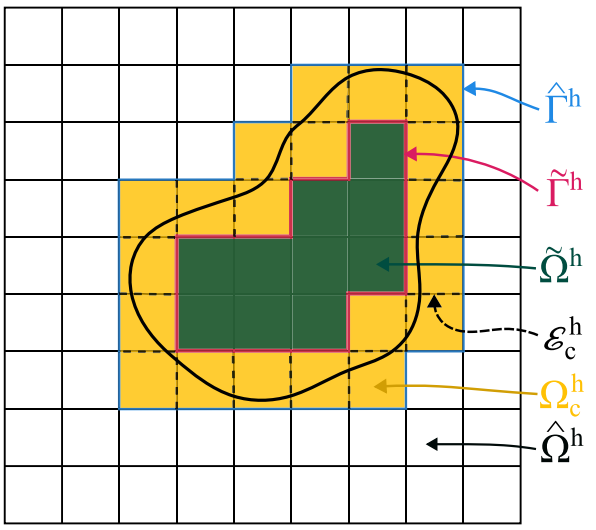
$$\begin{aligned} & \big(v_h, \partial_t u_h\big)_{\tilde{\Omega}^h} + \big(\nabla v_h, \boldsymbol{\sigma}(u_h)\big)_{\tilde{\Omega}^h} - \big(v_h, \boldsymbol{\sigma}(u_h) \cdot \boldsymbol{n}_{\tilde{\Gamma}^h}\big)_{\tilde{\Gamma}^h} - \big(\boldsymbol{\sigma}(v_h) \cdot \boldsymbol{n}_{\tilde{\Gamma}^h}, S_{\boldsymbol{d}}(u_h) - u_D\big)_{\tilde{\Gamma}^h} + \\ & \left(\frac{\beta_1}{h} S_{\boldsymbol{d}}(v_h), S_{\boldsymbol{d}}(u_h) - u_D\right)_{\tilde{\Gamma}^h} = (v_h, f)_{\tilde{\Omega}^h} \qquad \forall v_h \in \tilde{\mathcal{V}}_h. \end{aligned} \tag{23}$$

In Eq. (23), the symbol $\boldsymbol{n}_{\tilde{\Gamma}^h}$ denotes the outward pointing normal to the surrogate boundary $\tilde{\Gamma}^h$.

Proposition 1 (equivalence with SBM). *The GSBM formulation (14) reduces to the original SBM formulation (23) coupled with an extension operator into the extension region, $\Omega^h \setminus \tilde{\Omega}^h$, when the alternative weighting field $\tilde{\alpha}_h$ defined in (21) is used in (14).*



(a) Surrogate domain $\tilde{\Omega}^h$ and surrogate boundary $\tilde{\Gamma}^h$ used in the SBM formulation.



(b) Surrogate domain $\tilde{\Omega}^h$, cut domain Ω_c^h, and inactive domain $\hat{\Omega}^h$, the cut facets $\mathcal{E}_c^h$ and surrogate boundaries $\tilde{\Gamma}^h$ and $\hat{\Gamma}^h$.

Fig. 2. Sketch of the surrogate domain and boundaries for the SBM (left) and WSBM (right).

Proof. Let us expand each term in Eq. (14) when $\alpha_h = \tilde{\alpha}_h$, noting that $[\![\tilde{\alpha}_h]\!] = 1.0$ on $\tilde{\Gamma}^h$ and $[\![\tilde{\alpha}_h]\!] = 0.0$ on $\mathcal{E}^h \setminus \tilde{\Gamma}^h$, and $\{\!\{\cdot\}\!\}_\gamma = (\cdot)^+$ on Γ_D, being $(\cdot)^+$ the contribution from the active element in $\tilde{\Omega}^h$. For the sake of simplicity and without loss of generality, in this proof we consider the case of $r = 1$.

$$\left(v_\alpha, \partial_t u_h\right)_{\Omega^h} = \left(\tilde{\alpha}_h v_h, \partial_t u_h\right)_{\tilde{\Omega}^h} = \left(v_h, \partial_t u_h\right)_{\tilde{\Omega}^h}, \tag{24a}$$

$$\left(\nabla v_\alpha, \boldsymbol{\sigma}(u_h)\right)_{\Omega^h} = \left(\tilde{\alpha}_h \nabla v_h, \boldsymbol{\sigma}(u_h)\right)_{\Omega^h} = \left(\nabla v_h, \boldsymbol{\sigma}(u_h)\right)_{\tilde{\Omega}^h}, \tag{24b}$$

$$\left([\![v_\alpha]\!], \{\!\{\boldsymbol{\sigma}(u_h)\}\!\}_\gamma\right)_{\mathcal{E}^h} = \left([\![\tilde{\alpha}_h]\!]\{\!\{v_h\}\!\}, \{\!\{\boldsymbol{\sigma}(u_h)\}\!\}_\gamma\right)_{\mathcal{E}^h} = \left(v_h, \boldsymbol{\sigma}(u_h)\cdot \boldsymbol{n}_{\tilde{\Gamma}^h}\right)_{\tilde{\Gamma}^h}, \tag{24c}$$

$$\left(\{\!\{\boldsymbol{\sigma}(v_h)\}\!\}_\gamma, [\![\alpha_h]\!]\{\!\{S_{\boldsymbol{d}}(u_h) - u_D\}\!\}_\gamma\right)_{\mathcal{E}^h} = \left(\boldsymbol{\sigma}(v_h), S_{\boldsymbol{d}}(u_h) - u_D\right)_{\tilde{\Gamma}^h}, \tag{24d}$$

$$\left(\frac{\beta_1}{h}|[\![\alpha_h]\!]|\{\!\{S_{\boldsymbol{d}}(v_h)\}\!\}_\gamma, \{\!\{S_{\boldsymbol{d}}(u_h) - u_D\}\!\}_\gamma\right)_{\mathcal{E}^h} = \left(\frac{\beta_1}{h} S_{\boldsymbol{d}}(v_h), S_{\boldsymbol{d}}(u_h) - u_D\right)_{\tilde{\Gamma}^h}, \tag{24e}$$

$$\left(\beta_2(\alpha_h) h[\![\nabla v_h]\!], [\![\nabla u_h]\!]\right)_{\mathcal{E}^h} = \left(\kappa h[\![\nabla v_h]\!], [\![\nabla u_h]\!]\right)_{\mathcal{E}^h \setminus \{\tilde{\xi}\in\tilde{\Omega}^h\}} + \left(\kappa h[\![\nabla v_h]\!], [\![\nabla u_h]\!]\right)_{\tilde{\Gamma}^h}, \tag{24f}$$

$$\left(v_\alpha, f\right)_{\Omega^h} = \left(\alpha_h v_h, f\right)_{\Omega^h} = \left(v_h, f\right)_{\tilde{\Omega}^h}. \tag{24g}$$

Grouping Eqs. (24) and noting that the last term in (24f) enforces continuity of gradients in a weak sense on the surrogate boundary $\tilde{\Gamma}^h$, we end up with the SBM formulation as defined in (23) coupled with an extension problem on the domain $\hat{\Omega}^h \equiv \Omega^h \setminus \tilde{\Omega}^h$ that extends the solution u_h from the surrogate boundary $\tilde{\Gamma}^h$ satisfying the extension problem: *find* $u_h \in \hat{\mathcal{V}}_{h,D}$ *such that*

$$\left(\kappa h[\![\nabla v_h]\!], [\![\nabla u_h]\!]\right)_{\mathcal{E}^h \setminus \{\tilde{\xi}\in\tilde{\Omega}^h\}} + \left(\kappa h[\![\nabla v_h]\!], [\![\nabla u_h]\!]\right)_{\tilde{\Gamma}^h} = 0 \qquad \forall v_h \in \hat{\mathcal{V}}_{h,0}. \tag{25}$$

With $\hat{\mathcal{V}}_h \overset{\text{def}}{=} \left\{ v \in C^0(\hat{\Omega}^h) : v_K \in \mathbb{Q}^r(K),\ \forall K \in \mathcal{T}^h \setminus \tilde{\mathcal{T}}^h \right\}$, $\hat{\mathcal{V}}_{h,0} \overset{\text{def}}{=} \left\{ v_h \in \hat{\mathcal{V}}_h : v_h = 0 \text{ on } \tilde{\Gamma}^h \right\}$ and $\hat{\mathcal{V}}_{h,D} \overset{\text{def}}{=} \left\{ v_h \in \hat{\mathcal{V}}_h : v_h = u_h^* \text{ on } \tilde{\Gamma}^h \right\}$, where u_h^* is the solution of the SBM problem (23). □

Note that the extension operator, i.e. ghost penalty term, it is not needed in the original SBM formulation where the solution is only defined in the interior surrogate domain, i.e. where $\tilde{\alpha}_h = 1.0$. In such a case, there is no small cut-cell and the system is well-posed, see [84]. In this work we use the ghost penalty term as extension operator of the problem solution into the full background domain $\mathcal{D}$. The key point of adding the extension operator in that case is that it enables an smooth extension of the solution into the inactive part, and this is one of the crucial points to allow differentiation for arbitrary geometries.

Remark 4. An equivalent proof can be obtained for the case where the surrogate boundary is optimally selected as proposed in [68]. In that case, the surrogate domain contains the cut cells whose active volume fraction is larger than a certain threshold λ. Optimal results have been found for $\lambda = 0.5$. For this case, we define an alternative expression for the volume weighting given by

$$\hat{\alpha}_h = \begin{cases} 1.0, & \text{if } \alpha_h \geq \lambda, \\ 0.0, & \text{otherwise.} \end{cases} \tag{26}$$

The equivalent formulation to [68] can then be obtained by using $\hat{\alpha}_h$ in Eq. (14). Note that this case will be denoted as *fraction* in the results section or *O-SBM* equivalent, where we use $\lambda = 0.5$.

3.4. Equivalence with the original weighted shifted boundary method

Let us define the set of cells cut by the boundary Γ, $\mathcal{T}_c^h$, and corresponding domain, Ω_c^h, as depicted in Fig. 2b.

$$\mathcal{T}_c^h \overset{\text{def}}{=} \left\{K \in \mathcal{T}^h : K \cap \Gamma \neq \emptyset\right\}, \qquad \Omega_c^h \overset{\text{def}}{=} \text{int}\left(\cup_{K\in\mathcal{T}_c^h} K\right).$$

We also define the tesselation exterior to the physical domain, $\hat{\mathcal{T}}^h$, and corresponding domain $\hat{\Omega}^h$ as

$$\hat{\mathcal{T}}^h \overset{\text{def}}{=} \left\{K \in \mathcal{T}^h : K \subset \text{clos}(D \setminus \Omega)\right\}, \qquad \hat{\Omega}^h \overset{\text{def}}{=} \text{int}\left(\cup_{K\in\hat{\mathcal{T}}^h} K\right) \subseteq D \setminus \Omega.$$

We denote by $\hat{\Gamma}^h$ the outer surrogate boundary composed of the set of non-cut facets of topological dimension $d-1$ that do not belong to $\tilde{\Omega}^h$, and by $\mathcal{E}_c^h$, the set of facets that are intersected by Γ, see Fig. 2b. With these definitions, the active region of the discrete domain can be defined as $\Omega_{\text{act}}^h \overset{\text{def}}{=} \tilde{\Omega}^h \cup \Omega_c^h$ and the extension region as $\hat{\Omega}^h \equiv \Omega^h \setminus \Omega_{\text{act}}^h$. Note that in the original WSBM works [76–78], the active domain is denoted by Ω^h. Since in this work we do not distinguish between active and extensions domains in the integration, we use Ω^h to denote the discrete background domain.

Let us define the FE space on the active surrogate domain Ω_{act}^h as

$$\tilde{\mathcal{V}}_{h,c} = \left\{v \in C^0(\Omega_{\text{act}}^h) : v_K \in \mathbb{Q}^r(K),\ \forall K \in \tilde{\mathcal{T}}^h \cup \mathcal{T}_c^h\right\}. \tag{27}$$

The original WSBM formulation, with the equivalent formulation as proposed in [76] applied to problem (1a)-(1b), reads: *find* $u_h \in \tilde{\mathcal{V}}_{h,c}$ *such that:*

$$\begin{aligned}
&\left(\alpha_h v_h, \partial_t u_h\right)_{\Omega_{\text{act}}^h} + \left(\alpha_h \nabla v_h, \boldsymbol{\sigma}(u_h)\right)_{\Omega_{\text{act}}^h} - \left(v_h, \left\{\!\!\left\{\boldsymbol{\sigma}(u_h)\right\}\!\!\right\}_\gamma \cdot [\![\alpha_h]\!]\right)_{\tilde{\Gamma}^h\cup\mathcal{E}_c^h} - \left(v_h, \boldsymbol{\sigma}(u_h)\cdot\alpha_h \boldsymbol{n}\right)_{\hat{\Gamma}^h} - \\
&\left(\left\{\!\!\left\{\boldsymbol{\sigma}(v_h)\right\}\!\!\right\}_\gamma \cdot [\![\alpha_h]\!], \left\{\!\!\left\{S_{\boldsymbol{d}}(u_h) - u_D\right\}\!\!\right\}_\gamma\right)_{\tilde{\Gamma}^h\cup\mathcal{E}_c^h} - \left(\boldsymbol{\sigma}(v_h)\cdot\alpha_h\boldsymbol{n}, S_{\boldsymbol{d}}(u_h) - u_D\right)_{\hat{\Gamma}^h} + \\
&\left(\frac{\beta_1}{h}|[\![\alpha_h]\!]|\left\{\!\!\left\{S_{\boldsymbol{d}}(v_h)\right\}\!\!\right\}_\gamma, \left\{\!\!\left\{S_{\boldsymbol{d}}(u_h) - u_D\right\}\!\!\right\}_\gamma\right)_{\tilde{\Gamma}^h\cup\mathcal{E}_c^h} + \left(\frac{\beta_1}{h}\alpha_h S_{\boldsymbol{d}}(v_h), S_{\boldsymbol{d}}(u_h) - u_D\right)_{\hat{\Gamma}^h} + \\
&\sum_{l=1}^{r}\left(\beta_2(\alpha_h) h^{2l-1}[\![D_{\boldsymbol{n}}^l v_h]\!], [\![D_{\boldsymbol{n}}^l u_h]\!]\right)_{\tilde{\Gamma}^h\cup\mathcal{E}_c^h} = \left(\alpha v_h, f\right)_{\Omega_{\text{act}}^h} \qquad \forall v_h \in \tilde{\mathcal{V}}_{h,c}.
\end{aligned} \tag{28}$$

Note that in Eq. (28) we use the average operator $\{\!\{\cdot\}\!\}_\gamma$, as introduced in [77,78]. However, an equivalent version with $\{\!\{\cdot\}\!\} \overset{\text{def}}{=} \{\!\{\cdot\}\!\}_{\frac{1}{2}}$ was used in the first definition of the WSBM in [76].

Proposition 2 (equivalence with WSBM)**.** *The GSBM formulation reduces to the original WSBM formulation (28) coupled with an extension problem into the extension region* $\hat{\Omega}^h$.

Proof. Let us expand each term in Eq. (14), again restricting ourselves to the case of $r = 1$, noting that same arguments apply for arbitrary polynomial order r. Noting that $\alpha_h = 0$ in $\hat{\Omega}^h$ and $[\![\alpha_h]\!] = 0.0$ on $\mathcal{E}^h \setminus (\tilde{\Gamma}^h \cup \mathcal{E}_c^h \cup \hat{\Gamma}^h)$, we have that

$$\left(v_\alpha, \partial_t u_h\right)_{\Omega^h} = \left(\alpha_h v_h, \partial_t u_h\right)_{\Omega^h} = \left(\alpha_h v_h, \partial_t u_h\right)_{\Omega_{\text{act}}^h}, \tag{29a}$$

$$\left(\nabla v_\alpha, \boldsymbol{\sigma}(u_h)\right)_{\Omega^h} = \left(\alpha_h \nabla v_h, \boldsymbol{\sigma}(u_h)\right)_{\Omega^h} = \left(\alpha_h \nabla v_h, \boldsymbol{\sigma}(u_h)\right)_{\Omega_{\text{act}}^h}, \tag{29b}$$

$$\begin{aligned}
&\left([\![v_\alpha]\!], \left\{\!\!\left\{\boldsymbol{\sigma}(u_h)\right\}\!\!\right\}_\gamma\right)_{\mathcal{E}^h} = \left([\![\alpha_h]\!]\left\{\!\!\left\{v_h\right\}\!\!\right\}_\gamma, \left\{\!\!\left\{\boldsymbol{\sigma}(u_h)\right\}\!\!\right\}_\gamma\right)_{\mathcal{E}^h} = \\
&\quad \left(v_h, \left\{\!\!\left\{\boldsymbol{\sigma}(u_h)\right\}\!\!\right\}_\gamma \cdot [\![\alpha_h]\!]\right)_{\tilde{\Gamma}^h\cup\mathcal{E}_c^h} + \left(v_h, \boldsymbol{\sigma}(u_h)\cdot\alpha_h\boldsymbol{n}\right)_{\hat{\Gamma}^h},
\end{aligned} \tag{29c}$$

$$\begin{aligned}
&\left(\left\{\!\!\left\{\boldsymbol{\sigma}(v_h)\right\}\!\!\right\}_\gamma, [\![\alpha_h]\!]\left\{\!\!\left\{S_{\boldsymbol{d}}(u_h) - u_D\right\}\!\!\right\}_\gamma\right)_{\mathcal{E}^h} = \left(\left\{\!\!\left\{\boldsymbol{\sigma}(v_h)\right\}\!\!\right\}_\gamma \cdot [\![\alpha_h]\!], \left\{\!\!\left\{S_{\boldsymbol{d}}(u_h) - u_D\right\}\!\!\right\}_\gamma\right)_{\tilde{\Gamma}^h\cup\mathcal{E}_c^h} + \\
&\quad \left(\boldsymbol{\sigma}(v_h)\cdot\alpha_h\boldsymbol{n}, S_{\boldsymbol{d}}(u_h) - u_D\right)_{\hat{\Gamma}^h},
\end{aligned} \tag{29d}$$

$$\begin{aligned}
&\left(\frac{\beta_1}{h}|[\![\alpha_h]\!]|\left\{\!\!\left\{S_{\boldsymbol{d}}(v_h)\right\}\!\!\right\}_\gamma, \left\{\!\!\left\{S_{\boldsymbol{d}}(u_h) - u_D\right\}\!\!\right\}_\gamma\right)_{\mathcal{E}^h} = \left(\frac{\beta_1}{h}|[\![\alpha_h]\!]|\left\{\!\!\left\{S_{\boldsymbol{d}}(v_h)\right\}\!\!\right\}_\gamma, \left\{\!\!\left\{S_{\boldsymbol{d}}(u_h) - u_D\right\}\!\!\right\}_\gamma\right)_{\tilde{\Gamma}^h\cup\mathcal{E}_c^h} + \\
&\quad \left(\frac{\beta_1}{h}\alpha_h S_{\boldsymbol{d}}(v_h), S_{\boldsymbol{d}}(u_h) - u_D\right)_{\hat{\Gamma}^h},
\end{aligned} \tag{29e}$$

$$\begin{aligned}
&\left(\beta_2(\alpha_h) h[\![\nabla v_h]\!], [\![\nabla u_h]\!]\right)_{\mathcal{E}^h} = \left(\kappa h[\![\nabla v_h]\!], [\![\nabla u_h]\!]\right)_{\mathcal{E}^h\setminus\{\xi\in\tilde{\Omega}^h\}} + \left(\kappa h[\![\nabla v_h]\!], [\![\nabla u_h]\!]\right)_{\tilde{\Gamma}^h\cup\mathcal{E}_c^h} + \\
&\left(\kappa h[\![\nabla v_h]\!], [\![\nabla u_h]\!]\right)_{\hat{\Gamma}^h},
\end{aligned} \tag{29f}$$

$$\left(v_\alpha, f\right)_{\Omega^h} = \left(\alpha v_h, f\right)_{\Omega_{\text{act}}^h}. \tag{29g}$$

Grouping Eqs. (29) and noting that the term (29f) leads to the ghost penalty term in the original WSBM formulation together with the extension operator on the inactive domain weakly coupled to the solution in Ω_{act}^h through the gradient jump on the surrogate

boundary $\hat{\Gamma}^h$. Thus, we end up with the WSBM formulation as defined in (28) supplemented with an extension problem on the domain $\hat{\Omega}^h$ that extends the solution u_h from the surrogate boundary $\hat{\Gamma}^h$, satisfying the extension problem: *find* $u_h \in \hat{\mathcal{V}}_{h,D}$ *such that*

$$\left(\kappa h[\![\nabla v_h]\!], [\![\nabla u_h]\!]\right)_{\mathcal{E}^h \setminus \{\xi \in \hat{\Omega}^h\}} + \left(\kappa h[\![\nabla v_h]\!], [\![\nabla u_h]\!]\right)_{\hat{\Gamma}^h} = 0 \qquad \forall v_h \in \hat{\mathcal{V}}_{h,0}. \tag{30}$$

With $\hat{\mathcal{V}}_h = \left\{ v \in C^0(\hat{\Omega}^h) : v_K \in \mathbb{Q}^r(K),\ \forall K \in \hat{\mathcal{T}}^h \right\}$, $\hat{\mathcal{V}}_{h,0} = \left\{ v_h \in \hat{\mathcal{V}}_h : v_h = 0 \text{ on } \hat{\Gamma}^h \right\}$ and $\hat{\mathcal{V}}_{h,D} = \left\{ v_h \in \hat{\mathcal{V}}_h : v_h = u_h^* \text{ on } \hat{\Gamma}^h \right\}$, where u_h^* is the solution of the WSBM problem (28). □

We would like to note that Propositions 1 and 2 establish the equivalence of the proposed GSBM formulation with the original SBM and WSBM methods. We refer the reader to [85] and [83] for a formal analysis of the well-posedness of these formulations, respectively.

3.5. The GSBM formulation for three model problems

In this work, we consider three model problems: an elliptic problem for scalar fields, the Poisson problem, an elliptic problem for vectorial fields, i.e. the linear elasticity problem, and the transient Stokes problem to showcase the applicability of the proposed approach for multi-field problems with moving domains.

3.5.1. The Poisson problem

Let us consider the static Poisson problem with Dirichlet boundary condition on the full boundary $\Gamma_D \equiv \partial\Omega$, given by the following strong form

$$-\Delta u = f \qquad \text{in } \Omega, \tag{31a}$$

$$u = u_D \qquad \text{on } \Gamma_D. \tag{31b}$$

In this case, the equivalent operator $\boldsymbol{\sigma}(u)$ can be defined as $\boldsymbol{\sigma}(u) \overset{\text{def}}{=} \nabla u$. Replacing this definition in Eq. (14), the GSBM for the static Poisson problem reads: *find* $u_h \in \mathcal{V}_h$ *such that:*

$$\begin{aligned} &\left(\nabla v_\alpha, \nabla u_h\right)_{\Omega^h} - \left([\![v_\alpha]\!], \{\!\{\nabla u_h\}\!\}_\gamma\right)_{\mathcal{E}^h} - \left(\{\!\{\nabla v_h\}\!\}_\gamma, [\![\alpha_h]\!]\{\!\{S_{\boldsymbol{d}}(u_h) - u_D\}\!\}_\gamma\right)_{\mathcal{E}^h} + \\ &\left(\frac{\beta_1}{h}|[\![\alpha_h]\!]|\{\!\{S_{\boldsymbol{d}}(v_h)\}\!\}_\gamma, \{\!\{S_{\boldsymbol{d}}(u_h) - u_D\}\!\}_\gamma\right)_{\mathcal{E}^h} + \sum_{l=1}^{r} \left(\beta_2(\alpha_h) h^{2l-1} [\![D^l_{\boldsymbol{n}} v_h]\!], [\![D^l_{\boldsymbol{n}} u_h]\!]\right)_{\mathcal{E}^h} = \\ &(v_\alpha, f)_{\Omega^h} \qquad \forall v_\alpha \in \mathcal{V}_h^\alpha. \end{aligned} \tag{32}$$

In Eq. (32) the coefficient β_1 is defined as $\beta_1 = \chi(r+1)^2$, where χ is an algorithmic constant and r the interpolation order. In this work we select $\chi = 20$, or otherwise stated, in line with the choice made in [82]. The coefficient β_2 depends on the volume fraction, as defined in Eq. (17).

3.5.2. The linear elasticity problem

The strong form of the static linear elastic problem reads:

$$-\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}) = \boldsymbol{f} \qquad \text{in } \Omega, \tag{33}$$

$$\boldsymbol{u} = \boldsymbol{u}_D \qquad \text{on } \Gamma_D. \tag{34}$$

With $\boldsymbol{\sigma}(\boldsymbol{u}) \overset{\text{def}}{=} 2\mu\varepsilon(\boldsymbol{u}) + \lambda(\nabla \cdot \boldsymbol{u})\boldsymbol{I}$, where μ and λ are the Lamé constants. The equivalent GSBM formulation for the linear elasticity problem reads: *find* $\boldsymbol{u}_h \in \left(\mathcal{V}_h\right)^d$ *such that*

$$\begin{aligned} &\left(\varepsilon(\boldsymbol{v}_\alpha), \boldsymbol{\sigma}(\boldsymbol{u}_h)\right)_{\Omega^h} - \left([\![v_\alpha]\!], \{\!\{\boldsymbol{\sigma}(\boldsymbol{u}_h)\}\!\}_\gamma\right)_{\mathcal{E}^h} - \left(\{\!\{\boldsymbol{\sigma}(\boldsymbol{v}_h)\}\!\}_\gamma, [\![\alpha_h]\!]\{\!\{S_{\boldsymbol{d}}(\boldsymbol{u}_h) - \boldsymbol{u}_D\}\!\}_\gamma\right)_{\mathcal{E}^h} + \\ &\left(\frac{\beta_1}{h}|[\![\alpha_h]\!]|\{\!\{S_{\boldsymbol{d}}(\boldsymbol{v}_h)\}\!\}_\gamma, \{\!\{S_{\boldsymbol{d}}(\boldsymbol{u}_h) - \boldsymbol{u}_D\}\!\}_\gamma\right)_{\mathcal{E}^h} + \sum_{l=1}^{r} \left(\beta_2(\alpha_h)(2\mu+\lambda) h^{2l-1} [\![D^l_{\boldsymbol{n}} \boldsymbol{v}_h]\!], [\![D^l_{\boldsymbol{n}} \boldsymbol{u}_h]\!]\right)_{\mathcal{E}^h} = \\ &\left(\boldsymbol{v}_\alpha, f\right)_{\Omega^h} \qquad \forall \boldsymbol{v}_\alpha \in \left(\mathcal{V}_h^\alpha\right)^d. \end{aligned} \tag{35}$$

In Eq. (35) the coefficient β_1 is defined as $\beta_1 \overset{\text{def}}{=} \chi(r+1)^2(2\mu+\lambda)$, with the same default value for χ as given in Section 3.5.1.

3.5.3. The transient Stokes problem

Finally, the third model problem is the transient Stokes flow. The strong form of which reads:

$$\partial_t \boldsymbol{u} - \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}, p) = \boldsymbol{f} \qquad \text{in } \Omega(t), \tag{36}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega(t), \tag{37}$$

$$\boldsymbol{u} = \boldsymbol{u}_D \qquad \text{on } \Gamma_D, \tag{38}$$

$$\boldsymbol{u}(0) = \boldsymbol{u}_0 \qquad \text{in } \Omega(0). \tag{39}$$

With $\boldsymbol{\sigma}(\boldsymbol{u},p) \overset{\text{def}}{=} 2\nu\boldsymbol{\varepsilon}(\boldsymbol{u}) - p\boldsymbol{I}$, where ν is the kinematic viscosity. The equivalent semi-discrete GSBM formulation for the transient Stokes problem reads: *find* $[\boldsymbol{u}_h, p_h] \in \left(\mathcal{V}_h\right)^d \times \mathcal{Q}_h$ *such that*

$$
\begin{aligned}
&\left(\boldsymbol{v}_\alpha, \partial_t \boldsymbol{u}_h\right)_{\Omega^h} + \left(\boldsymbol{\varepsilon}(\boldsymbol{v}_\alpha), 2\nu\boldsymbol{\varepsilon}(\boldsymbol{u}_h)\right)_{\Omega^h} - \left(\nabla\cdot\boldsymbol{v}_\alpha, p_h\right)_{\Omega^h} + \left(q_\alpha, \nabla\cdot\boldsymbol{u}_h\right)_{\Omega^h} - \left(\left[\!\left[\boldsymbol{v}_\alpha\right]\!\right], \left\{\!\left\{\boldsymbol{\sigma}(\boldsymbol{u}_h,p_h)\right\}\!\right\}_\gamma\right)_{\mathcal{E}^h} - \\
&\left(\left\{\!\left\{2\nu\boldsymbol{\varepsilon}(\boldsymbol{v}_\alpha) + q_\alpha\boldsymbol{I}\right\}\!\right\}_\gamma, \left[\!\left[\alpha_h\right]\!\right]\left\{\!\left\{S_{\boldsymbol{d}}(\boldsymbol{u}_h) - \boldsymbol{u}_D\right\}\!\right\}_\gamma\right)_{\mathcal{E}^h} + \left(\frac{\beta_1}{h}\left|\left[\!\left[\alpha_h\right]\!\right]\right|\left\{\!\left\{S^1_{\boldsymbol{d}}(\boldsymbol{v}_h)\right\}\!\right\}_\gamma, \left\{\!\left\{S_{\boldsymbol{d}}(\boldsymbol{u}_h) - \boldsymbol{u}_D\right\}\!\right\}_\gamma\right)_{\mathcal{E}^h} + \\
&\sum_{l=1}^{r_u}\left(\beta_2^u(\alpha_h)\nu h^{2l-1}\left[\!\left[D^l_{\boldsymbol{n},s}\boldsymbol{v}_h\right]\!\right], \left[\!\left[D^l_{\boldsymbol{n},s}\boldsymbol{u}_h\right]\!\right]\right)_{\mathcal{E}^h} + \sum_{l=1}^{r_p}\left(\frac{\beta_2^p(\alpha_h)}{\nu+\beta_{\text{div}}}h^{2l+1}\left[\!\left[D^l_{\boldsymbol{n}}q_h\right]\!\right], \left[\!\left[D^l_{\boldsymbol{n}}p_h\right]\!\right]\right)_{\mathcal{E}^h} + \\
&\left(\beta_{\text{div}}\nabla\cdot\boldsymbol{v}_h, \nabla\cdot\boldsymbol{u}_h\right)_{\Omega^h} = \left(v_\alpha, f\right)_{\Omega^h} \qquad \forall [v_\alpha, q_\alpha] \in \left(\mathcal{V}^\alpha_h\right)^d \times \mathcal{Q}^\alpha_h.
\end{aligned}
\tag{40}
$$

With $D^l_{\boldsymbol{n},s}$ the lth partial symmetric directional derivative along the face normal.

Note that a few key aspects have been introduced in Eq. (40) in comparison to the equivalent formulations for the Poisson and linear elasticity problems. First, here we have multiple fields with compatible finite element spaces. In particular, here we use an inf-sup stable velocity pressure pair $\mathbb{Q}^2 - \mathbb{Q}^1$, that is spaces composed of second-order Lagrange polynomials in quadrilaterals for the velocity FE space and first-order Lagrange polynomials in quadrilaterals for the pressure FE space. Both spaces are $C^0(\Omega^h)$, that is, continuous across elements. To be precise,

$$
\mathcal{V}_h \overset{\text{def}}{=} \left\{v \in C^0(\Omega^h) \,:\, v_K \in \mathbb{Q}^2(K),\ \forall K \in \mathcal{T}^h\right\}, \tag{41}
$$

$$
\mathcal{Q}_h \overset{\text{def}}{=} \left\{q \in C^0(\Omega^h) \,:\, q_K \in \mathbb{Q}^1(K),\ \forall K \in \mathcal{T}^h\right\}. \tag{42}
$$

With the equivalent $\mathcal{V}^\alpha_h$ and $\mathcal{Q}^\alpha_h$ given by Eq. (13).

To keep the accuracy of the method when using second-order polynomials for the velocity FE space, the extrapolation operator $S_{\boldsymbol{d}}(\boldsymbol{u}_h)$ has to include the second-order Taylor expansion, i.e.

$$
S_{\boldsymbol{d}}(u)\big|_{\boldsymbol{x}_0} \overset{\text{def}}{=} u(\boldsymbol{x}_0) + \nabla u(\boldsymbol{x}_0)\cdot\boldsymbol{d}(\boldsymbol{x}_0) + \frac{1}{2}\nabla\nabla u(\boldsymbol{x}_0) \,:\, \left(\boldsymbol{d}(\boldsymbol{x}_0)\otimes\boldsymbol{d}(\boldsymbol{x}_0)\right), \tag{43}
$$

and we denote by $S^1_{\boldsymbol{d}}(v)$ the linear expansion as described in Eq. (9). Similarly, the ghost penalty stabilization for the velocity field includes now a higher-order term, see the second term of the third row in Eq. (40).

An additional term appearing in (40) is the so-called *grad-div* stabilization, i.e. $\left(\beta_{\text{div}}\nabla\cdot\boldsymbol{v}_h, \nabla\cdot\boldsymbol{u}_h\right)_{\Omega^h}$. This term is consistent with the conservation of mass equation and, thus, penalizes the divergence of the velocity field. Note that for this term we do not use the weighted function space. The addition of this term has been shown to reduce spurious pressure oscillations in time for incompressible flow problems in moving domains, see [75,77,78]. This term is also related to the pressure subscale in a variational multiscale stabilization context [86,87], also shown to be relevant even in the case of inf-sup stable formulations [88,89].

In Eq. (40) the coefficient β_1 is defined as $\beta_1 \overset{\text{def}}{=} \chi(r+1)^2\nu$ with $\chi = 1.0$. The coefficients β_2^u and β_2^p are defined in Eq. (17) and $\beta_{\text{div}} > 0$ is a viscosity-dependent algorithmic coefficient chosen as $\beta_{\text{div}} = 1.0e2\nu$ or, otherwise stated. We also refer the reader to [77,88,89] for alternative definitions in the context of variational multiscale stabilization methods.

4. Numerical results

In what follows, we present the numerical results for the three model problems described in Section 3.5. For all cases, we demonstrate that the formulation described in Section 3 converges with the prescribed accuracy by the underlying FE space, and we show how the proposed approach can be applied to a variety of complex geometries and moving domains. We would like to highlight that the proposed formulation is tested for a set of geometries defined by parametric level-set functions, see Eqs. (44)–(46), where different parametrizations are used, as well as explicit geometry representations, see Section 4.1.4. The same formulation is used for the different domain geometries, without modifiying the definition of integration rules, finite element spaces or identification of geometrical entities where different terms have to be integrated. Furthermore, in Section 4.1.3 we present a case with a parametric domain where changes on the parameter value lead to topology changes.

4.1. Poisson problem results

4.1.1. Convergence analysis

The first example is the solution of the Poisson problem for three different geometries given by an analytical level set function. We use a circle, ϕ_{circle}, a flower-shape, ϕ_{flower}, and a parallelogram with sharp corners and sides not aligned with the background grid, ϕ_{parall}. The solution is obtained in a background domain of size $[0,1]\times[0,1]$.

$$
\phi_{\text{circle}}(x,y) = 0.3 - \left((x-0.5)^2 + (y-0.5)^2\right)^{1/2}, \tag{44}
$$

$$
\phi_{\text{flower}}(x,y) = R(x,y) - \left(\mathbf{w}(x,y)\cdot\mathbf{w}(x,y)\right)^{1/2}, \tag{45}
$$

$$
\phi_{\text{parall}}(x,y) = \min\left\{d_1(x,y), d_2(x,y), d_3(x,y), d_4(x,y)\right\}. \tag{46}
$$

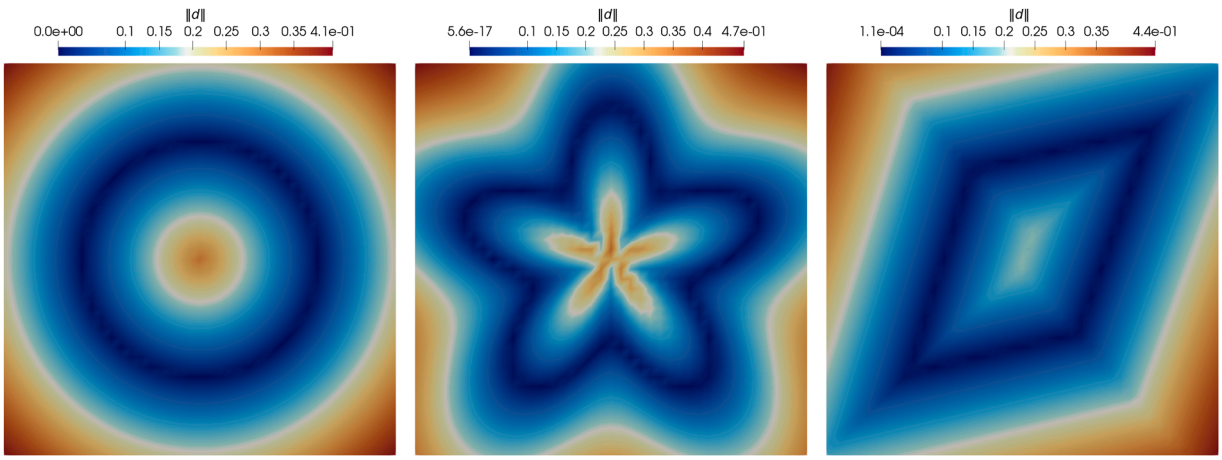


Fig. 3. Distance function magnitude for the circle (left), flower-shape (center) and parallelogram (right).

For the flower shape, the following parameters are used: $\mathbf{w}(x,y) = [x-0.5, y-0.5]^T$, $t(x,y) = \arg\left(w_1(x,y) + i\, w_2(x,y)\right)$ and $R(x,y) = 0.3(1+0.3\sin(5\,t(x,y)))$. For the parallelogram, the vertices are located at $v_1 = (0.21, 0.21)$, $v_2 = (0.64, 0.31)$, $v_3 = (0.83, 0.83)$, $v_4 = (0.37, 0.73)$. Denoting the vertex coordinates (x_i, y_i) for each vertex v_i, the distances used in Eq. (46) can be computed as:

$$d_1(\mathbf{x}) = \frac{l_1(x,y)}{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}},$$
$$d_2(\mathbf{x}) = \frac{l_2(x,y)}{\sqrt{(x_3-x_2)^2+(y_3-y_2)^2}}.$$
$$d_3(\mathbf{x}) = \frac{l_3(x,y)}{\sqrt{(x_4-x_3)^2+(y_4-y_3)^2}},$$
$$d_4(\mathbf{x}) = \frac{l_4(x,y)}{\sqrt{(x_1-x_4)^2+(y_1-y_4)^2}},$$

with

$$l_1(x,y) = (y-y_1)(x_2-x_1) - (x-x_1)(y_2-y_1),$$
$$l_2(x,y) = (y-y_2)(x_3-x_2) - (x-x_2)(y_3-y_2),$$
$$l_3(x,y) = (y-y_3)(x_4-x_3) - (x-x_3)(y_4-y_3),$$
$$l_4(x,y) = (y-y_4)(x_1-x_4) - (x-x_4)(y_1-y_4).$$

In Fig. 3 we depict the magnitude of the three distance functions. The isocontour $|\boldsymbol{d}| = 0$ defines the domain boundary $\partial\Omega$ for each case. In Fig. 4 we show the volume fraction field α_h for the circle geometry using a 40 by 40 elements mesh, for the generalized weighted case, Eq. (12), for the SBM equivalent (21) and for the optimal SBM surrogate, Eq. (26).

In this subsection, we solve a problem with a manufactured analytical solution. That is, we enforce the force term and boundary condition to satisfy Eq. (31a) for the given analytical solution

$$u(x,y) = \sin(15\pi x)\sin(15\pi y). \tag{47}$$

In Fig. 5 we depict the solution of the problem, where we see that the analytical solution is enforced in the interior of the domain Ω and extended through a smooth extension operator outside the active domain.

In Fig. 6 we show the error for the equivalent case to WSBM, where α_h is computed using Eq. (12). We see that for this case the largest errors are located at the vertices located outside the domain. This is expected, as the solution outside the domain does not satisfy the prescribed problem.

To demonstrate that the proposed formulation retains the optimal convergence rates, in Fig. 7 we depict the L^2-norm of the solution error for the different cases: *WSBM* equivalent using α_h as in Eq. (12), *SBM* equivalent using $\tilde{\alpha}_h$ as in Eq. (21), and *O-SBM* equivalent to SBM with optimal surrogate with $\hat{\alpha}_h$ as defined in Eq. (26). The L^2-norm of the error is computed using the respective weights:

$$\|e\|_\Omega \approx \left(\int_{\Omega^h} \alpha_i (u-u_h)^2 d\Omega\right)^{1/2}, \tag{48}$$

with $\alpha_i = \{\alpha_h, \tilde{\alpha}_h, \hat{\alpha}_h\}$, depending on the case. In Fig. 7 we see that all three variants of the proposed GSBM formulation result in errors with the prescribed second-order accuracy for the three geometries considered here, i.e. circle (left), flower shape (center) and

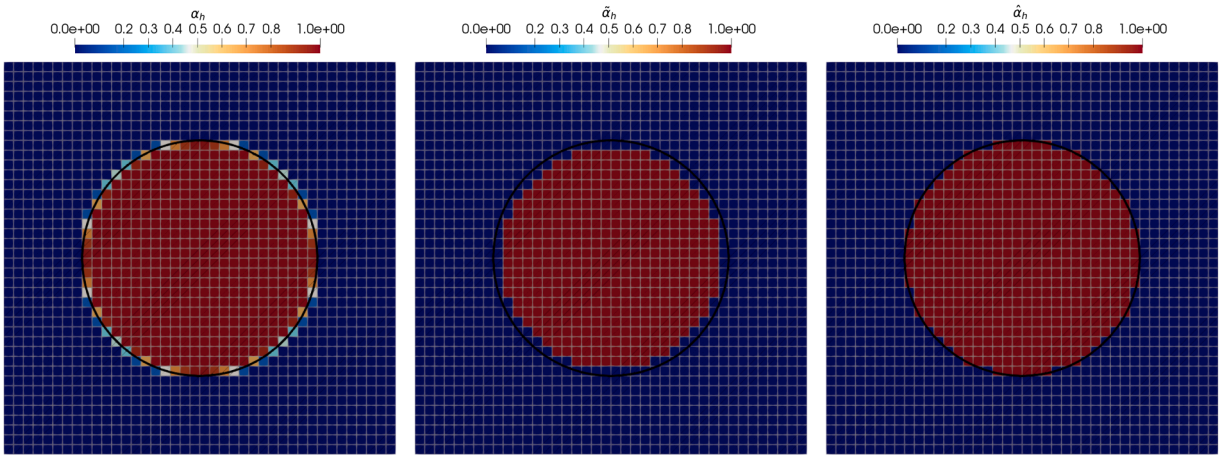


Fig. 4. Volume fraction weights α_h (left), $\tilde{\alpha}_h$ (center) and $\hat{\alpha}_h$ (right).

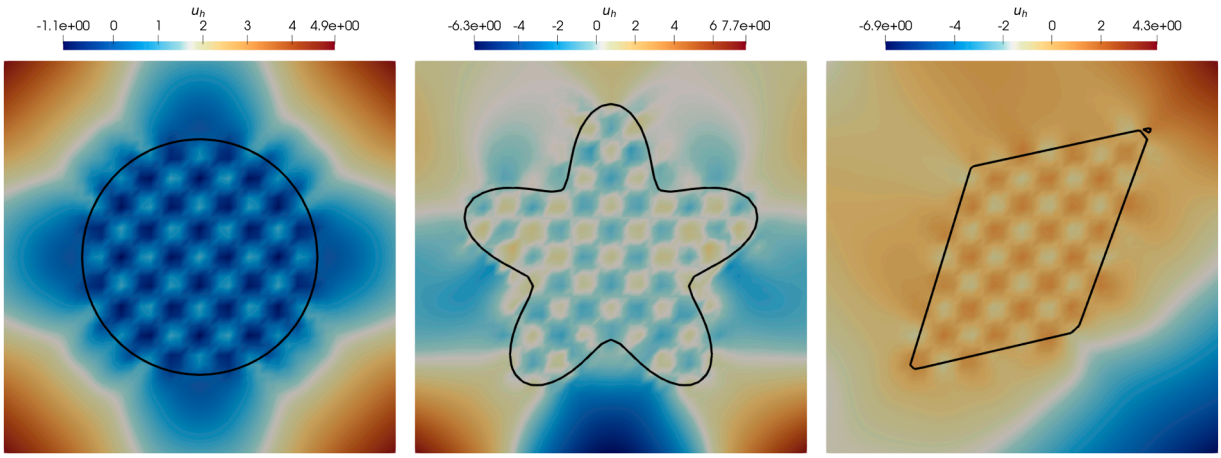


Fig. 5. Computed solution using the WSBM equivalent in a 40 by 40 background grid. Circle level set function (left), flower-shape level set function (center) and parallelogram level set function (right).

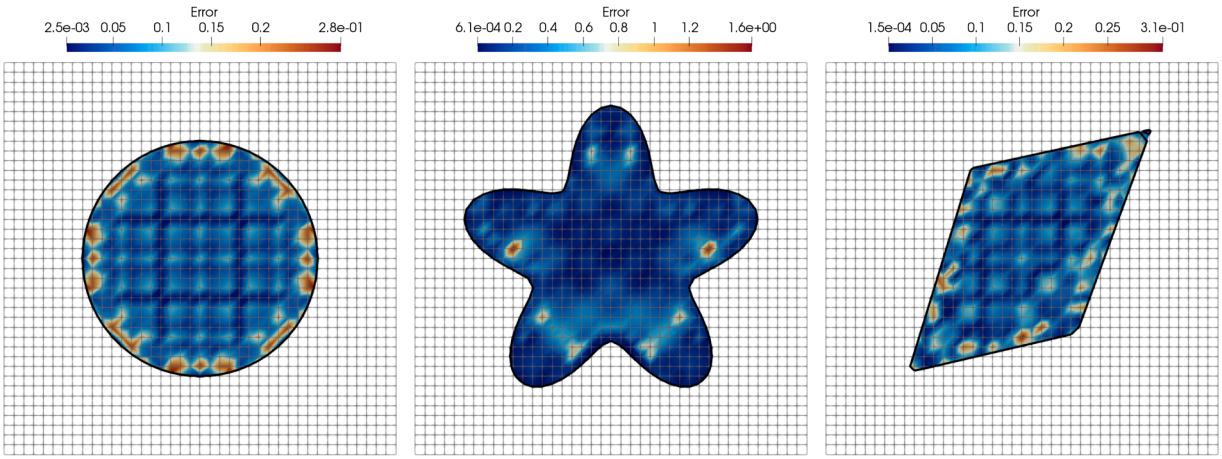


Fig. 6. Absolute value of the error of for the solution computed using the WSBM equivalent in a 40 by 40 background grid. The error is depicted only in the interior of the physical domain (Ω). Circle level set function (left), flower-shape level set function (center) and parallelogram level set function (right).

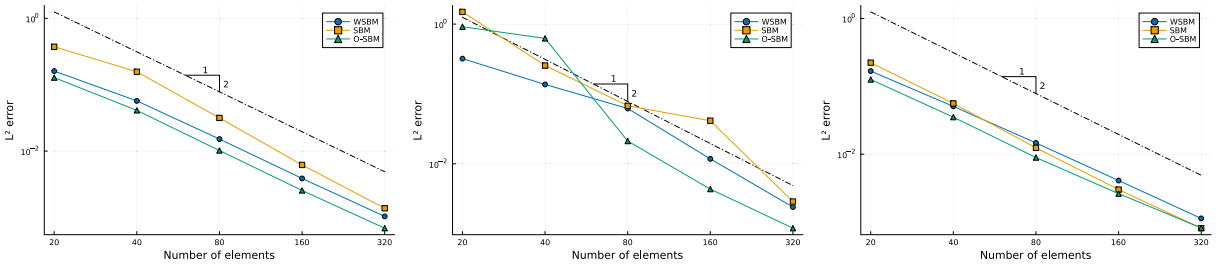


Fig. 7. Convergence of the L2-norm using the different weight definitions for the circle level set function (left), the flower-shape level set function (center) and the parallelogram level set function (right).

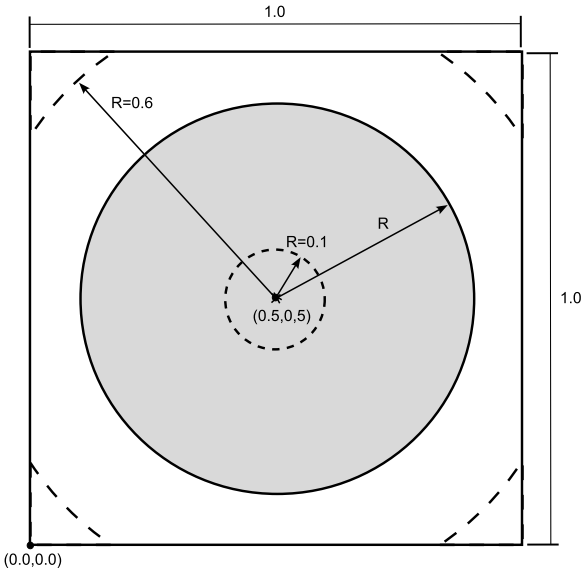


Fig. 8. Sketch of the geometry used in the condition number analysis.

parallelogram (right). Here we see that using the weights definition $\hat{\alpha}_h$, equivalent to SBM with optimal surrogate boundary, results in smaller errors in general.

4.1.2. Condition number analysis

The proposed formulation uses a smooth 4th order extension operator outside the active domain, which results in a condition number of the system matrix, $\kappa(A)$, that scales with $\kappa(A) \propto h^{-4}$ instead of the standard $\kappa(A) \propto h^{-2}$ rate expected for the 2nd order Laplacian operator. In this subsection, we evaluate the influence of the increased matrix condition number for various cases modifying the ratio of active domain over background domain area. Here we use the approach equivalent to *WSBM*, with α_h.

To perform this analysis we consider a circle centered at $(0.5, 0.5)$ with variable radius in the range $R \in [0.1, 0.6]$ embedded in a square background grid of size $[0, 1] \times [0, 1]$, see Fig. 8. Note that for the cases where $R > 0.5$ the physical domain Ω is not contained in the background grid Ω^h. In such cases, we consider as physical domain the intersection of both domains, $\Omega \cap \Omega^h$, enforcing strong Dirichlet boundary conditions on the boundaries of Ω^h that belong to Ω. Again, here we solve enforcing a manufactured analytical solution as in the previous case, with the same analytical function defined in Eq. (47).

The results are shown in Fig. 9, where we plot the L^2-norm of the solution error (left) and the condition number of the resulting system matrix using the 2-norm (right). We see that the L^2-norm of the error converges with the prescribed rate for all the cases. Note that for the case $R = 0.1$, the coarsest meshes cannot capture well the geometry and, thus, the error plot shows a lower rate for these points. Looking at the condition number, i.e. Fig. 9 (right), we see that for small circle radius the condition number is governed by the extension operator. This is expected as we have a larger ratio of extension domain to physical domain. When the radius is increased, the condition number rate tends to h^{-2}, as expected for the Laplacian operator. We can conclude that depending on the background to active domain ratio, the GSBM formulation can lead to worse condition numbers than the original SBM and WSBM formulations, where there is no coupling with an extension operator. However, more studies need to be performed to have a complete picture of the condition number behavior of the GSBM.

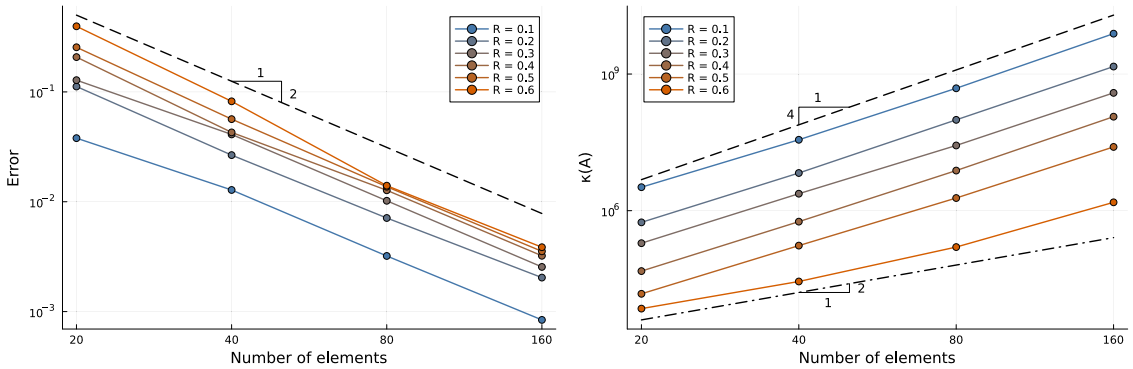


Fig. 9. L^2-norm of the solution error and condition number of the resulting matrix for varying radius of the circle.

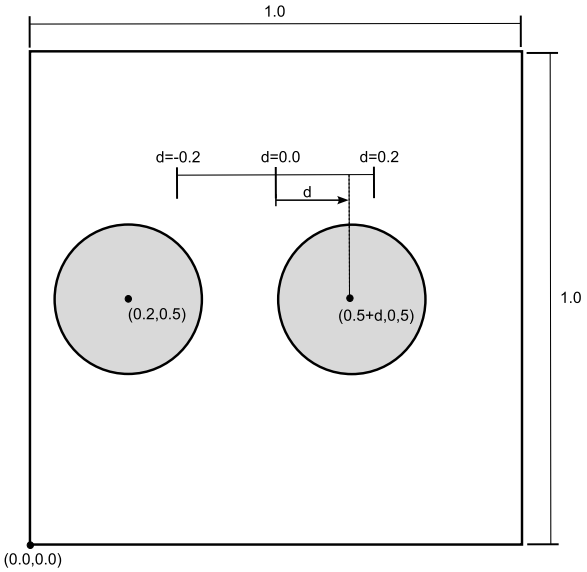


Fig. 10. Sketch of the parametric geometry problem.

4.1.3. Parametric domain with topology changes

One of the distinct features of the proposed GSBM formulation is the ability to define geometry-parametric formulations without having to define new FE spaces and integration measures, see Proposition 3.2. To showcase this ability, we consider a test with a parametrized geometry that includes a change in topology. Here, we consider two circles of radius $R = 0.15$, one with a fixed center at $(0.2, 0.5)$ and the other at $(0.5 + d, 0.5)$, with d a parameter that is in the range $d \in [-0.2, 0.2]$. The two circles are embedded in a background grid of size $[0, 1] \times [0, 1]$, see Fig. 10.

For this case, we solve the problem enforcing a manufactured analytical solution given by Eq. (47). Here we fix the background grid to 100 elements per direction.

In Fig. 11 we depict the L^2-norm of the solution error for varying parameter d using the three variants of the GSBM approach. We can see that for all cases the error stays relatively constant within certain margins with fluctuations that depend on the formulation variant. When we use the *SBM* equivalent, the error has strong fluctuations due to the change of the surrogate domain for varying d. For the optimal surrogate boundary *O-SBM*, there error is in general lower, but still has some fluctuations in the error. For the *WSBM* equivalent, the error fluctuations for varying d are smoother, as expected due to the weighting of the cut cells. Note that the error for all cases does not show significant change when going through the topology change at $d = 0.0$, highlighting the capability of the formulation to handle parametric geometries with eventual topology changes.

In Fig. 12 we show the error of the solution for the case equivalent to the *SBM* formulations, i.e. using $\tilde{\alpha}_h$, for three different values of the parameter d, $d = -0.1$ (left), $d = 0.0$ (center) and $d = 0.1$ (right).

4.1.4. Explicit geometrical boundary representation in three dimensions

In contrast to prior case studies, which addressed implicit geometrical representations via level sets, the example detailed in this section involves the Low Poly Stanford Bunny geometry represented using an .STL file format. Consequently, an explicit geometrical representation is employed. The STL file is accessible online in the Thingi10k library under model ID 293137. The computation of

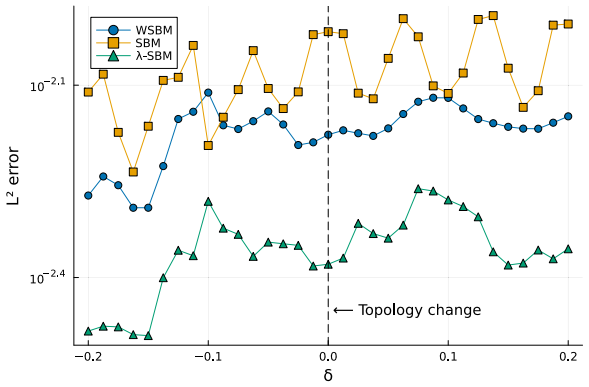


Fig. 11. L^2-norm of the solution error for varying position of the circles.

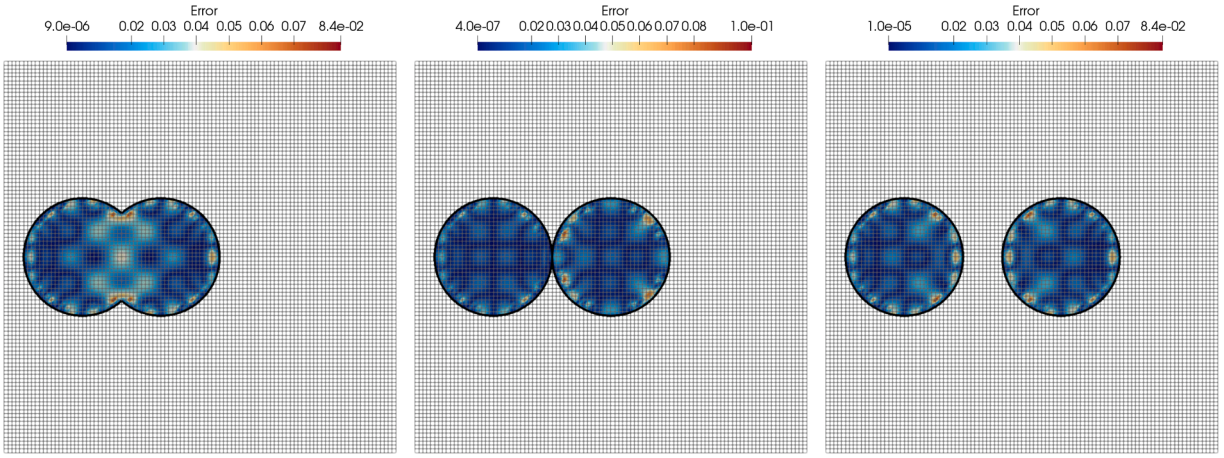


Fig. 12. Solution error at different circles position: $d = -0.1$ (left), $d = 0.0$ (center) and $d = 0.1$ (right). The error is depicted only in the interior of the physical domain (Ω).

the distance function necessary for the SBM-type methodologies is achieved using the *STLCutters.jl* package, developed by Martorell and Badia [90]. This package implements a geometrical intersection and parametric distance computation algorithm, as explained in Chapter 3 by Badia et al. [55]. Furthermore, volume fractions for each cell are obtained by constructing an integration space through a subtriangulation algorithm and subsequently computing the ratio of this approximate volume to the total cell volume on a cell-wise basis. The weights assigned to each cell for every SBM scenario are presented in Fig. 13. The WSBM on the left exhibits variable weights, while the O-SBM utilizes binary weights, with certain elements permitted to intersect the geometry. On the right, the SBM is illustrated with binary weights without any intersections with the geometry. It is important to note that only the active elements are depicted, although the complete solution extends over the entire background grid, which is represented by the black outline. The size of this background grid is $\mathcal{D} = [0, 1] \times [0, 1] \times [0, 1]$ with a partition of 30 elements in each direction. The Stanford Bunny geometry has been shrunk to fit completely into the background grid; the bounding box of the geometry is [0.0875305, 0.898451; 0.171241, 0.820928; 0.119404, 0.923852].

In the context of this case study, a manufactured analytical solution is utilized to impose the boundary conditions and the forcing term. This analytical solution is specified as follows:

$$u(x, y, z) = z. \tag{49}$$

Utilizing this analytical equation, we determine the error associated with each SBM type as illustrated in Fig. 14. For the purposes of comparison, the error is picked relative to the largest error observed in the case exhibiting the smallest error. Consequently, it might appear that the WSBM and O-SBM (left and center) exhibit significant error; however, the magnitude of this error is on the order of 10^{-10}. In summary, it seems that the approximation approaches machine precision.

Looking at the convergence of the error, shown in Fig. 17, we see that all three variants of the GSBM formulation result in optimal convergence rates. Noting that the *O-SBM* variant generally results in lower errors, following the similar trend observed for the Poisson case.

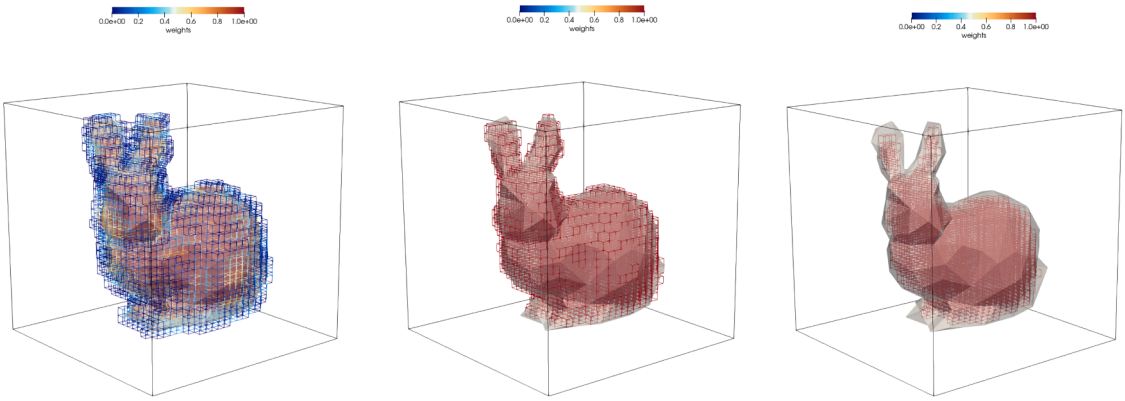


Fig. 13. Geometrical representation of the Low Poly Stanford Bunny in $30 \times 30 \times 30$ background grid. The weights are only depicted in active elements ($\alpha_h > 0$). WSBM (left), O-SBM (center) and SBM (right).

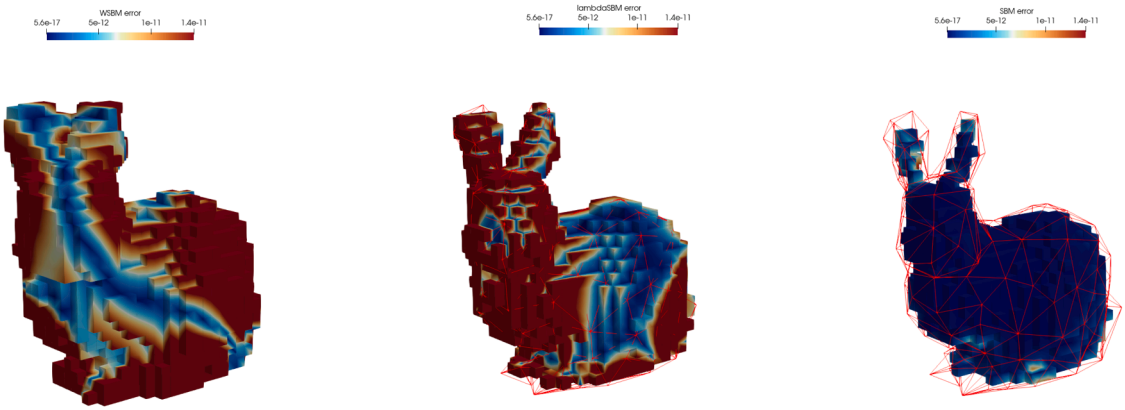


Fig. 14. Error of the method of manufactured solution for the geometrical case of the Low Poly Stanford Bunny in $30 \times 30 \times 30$ background grid. The geometry is outlined using a red wireframe. Error only depicted in active elements ($\alpha_h > 0$). WSBM (left), O-SBM (center) and SBM (right). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

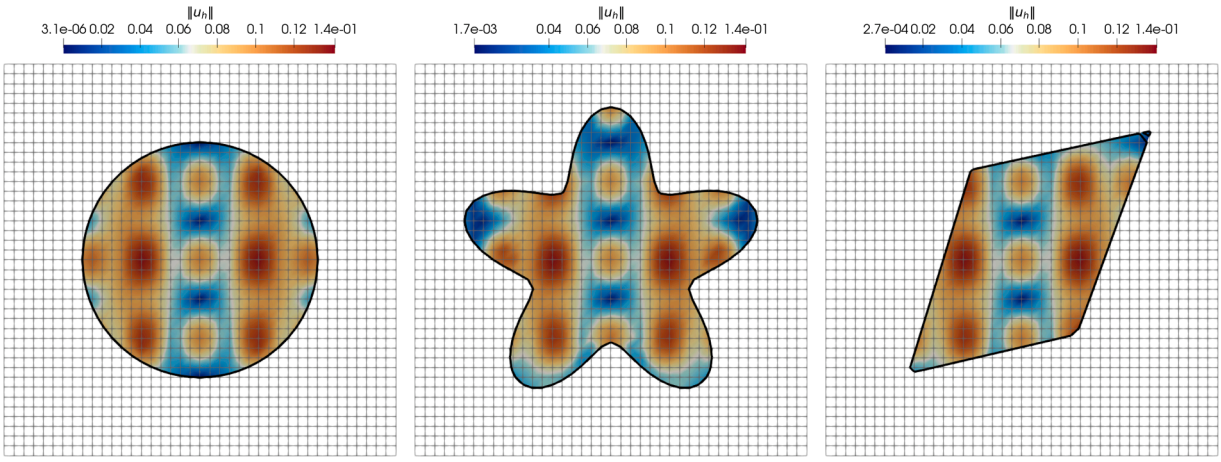


Fig. 15. Computed displacement absolute value using the optimal SBM equivalent in a 40 by 40 background grid. Circle level set function (left), flower-shape level set function (center) and parallelogram level set function (right). The solution is depicted only in the interior of the physical domain (Ω).

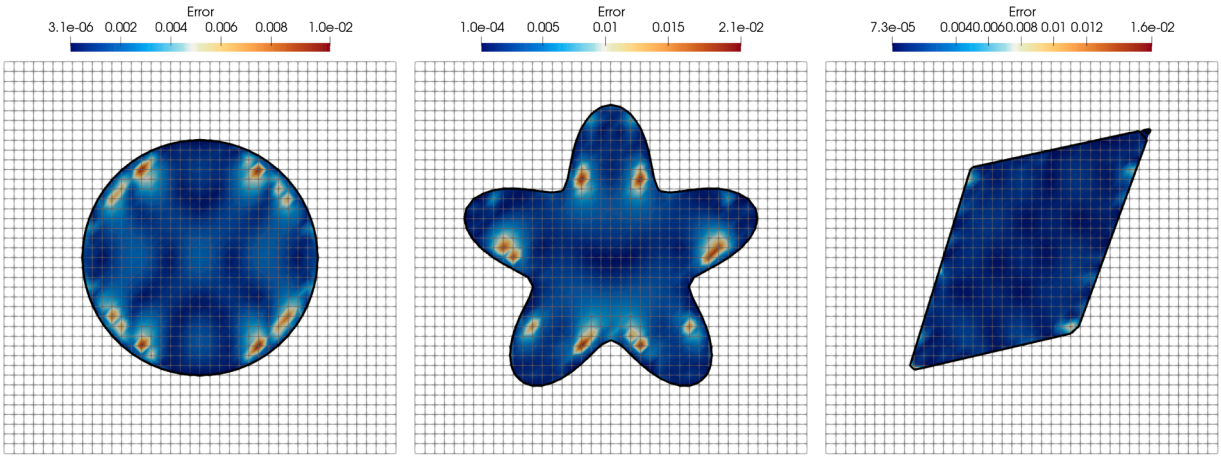


Fig. 16. Absolute value of the displacement error of for the solution computed using the optimal SBM equivalent in a 40 by 40 background grid. Error only depicted in active elements ($\alpha_h > 0$). Circle level set function (left), flower-shape level set function (center) and parallelogram level set function (right). The error is depicted only in the interior of the physical domain (Ω).

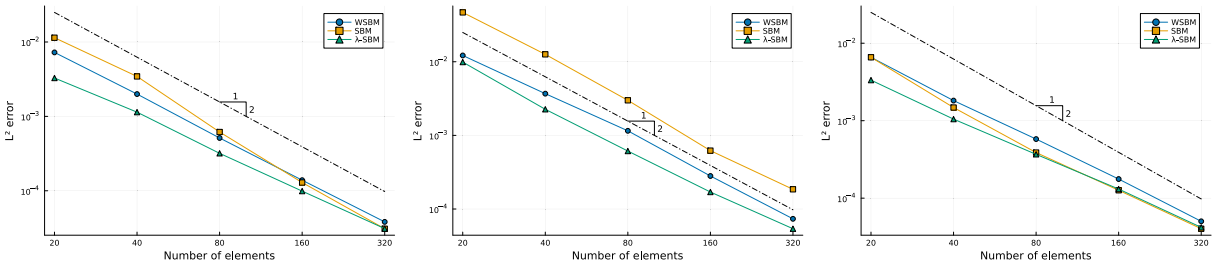


Fig. 17. L^2-norm of error convergence: circle shape (left), flower shape (center) and quadrilateral (right).

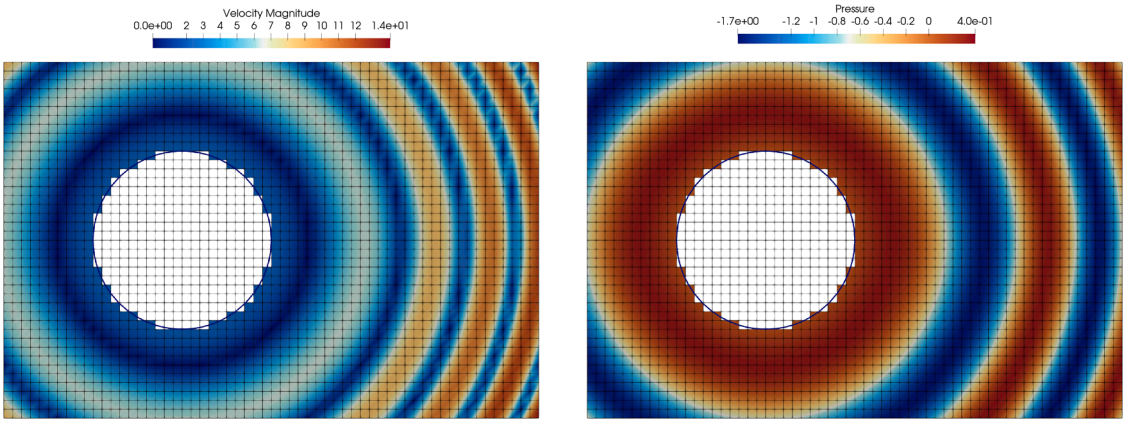


Fig. 18. Computed velocity magnitude (left) and pressure (right) using the optimal SBM equivalent in a 60 by 40 background grid using the flower-shape level set function. Results are shown only in elements with positive α.

In Fig. 20 we plot the L^2-norm of the velocity (left) and pressure (right) fields. We see that the WSBM and O-SBM variants of the proposed method attain the optimal convergence rates for both velocity and pressure. The SBM variant has a slightly lower rate for the velocity. However, all methods have very similar pressure errors.

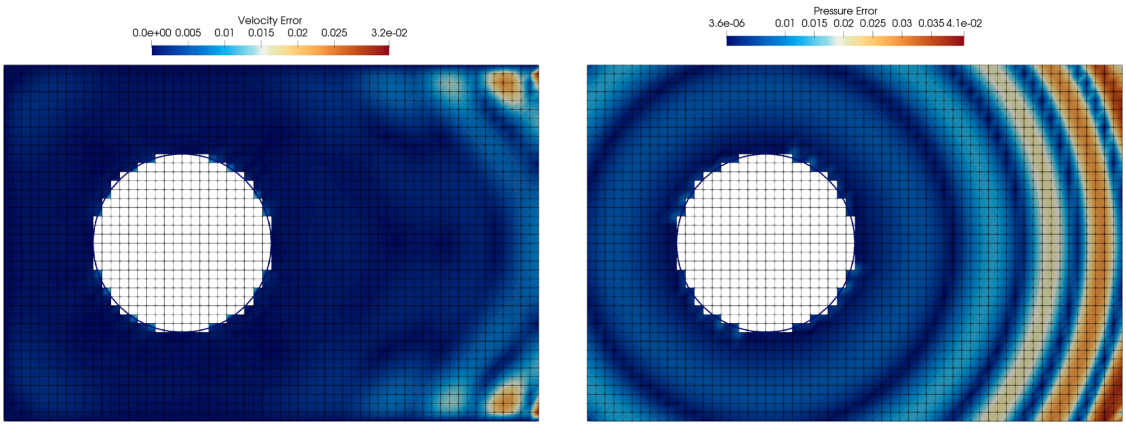


Fig. 19. Absolute value of the velocity error (left) and pressure error (right) for the solution computed using the optimal SBM equivalent in a 60 by 40 background grid in a flower-shape level set function. Results are shown only in elements with positive α.

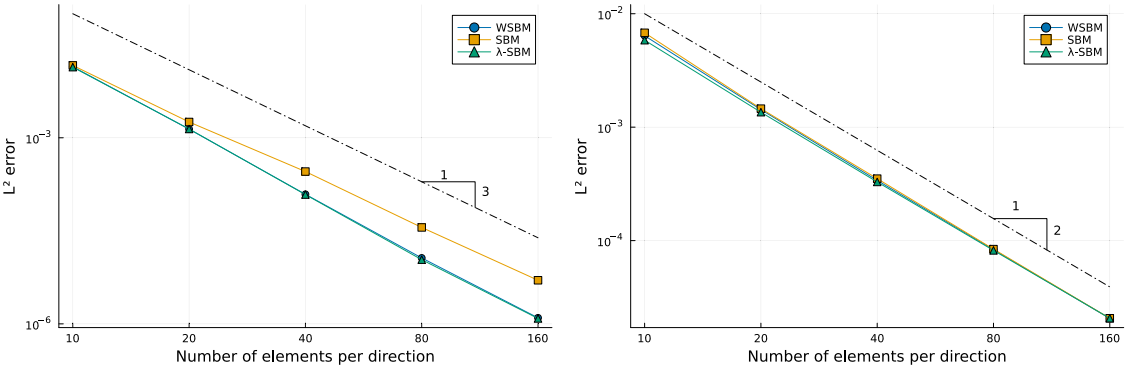


Fig. 20. L^2-norm of the velocity (left) and pressure (right) error convergence.

4.2. Linear elasticity problem results

4.2.1. Convergence analysis

In this section we consider the linear elasticity problem formulation defined in Eq. (35), with Lamé coefficients $\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}$ and $\mu = \frac{E}{2(1+\nu)}$, where $E = 200$ GPa and $\nu = 0.3$. Here we solve the problem enforcing a manufactured solution that takes the following expression:

$$\boldsymbol{u}(x, y) = \begin{bmatrix} -0.1\cos(3\pi x)\sin(\pi y) \\ 0.1\cos(7\pi x)\sin(5\pi y) \end{bmatrix}. \tag{50}$$

Again, we consider the three different geometries presented in Section 4.1.1, with the three formulation variants, namely *WSBM*, *SBM* and *O-SBM*, using the definitions of the weighting factor given by Eqs. (12), (21) and (26), respectively. In Fig. 15 we depict the computed solution for the three shapes using the optimal SBM approach (O-SBM), computed using a background grid of 40×40 cells. In Fig. 16 we show the respective L^2-norm of the error.

4.3. Stokes problem results

4.3.1. Manufactured solution for the static Stokes problem

In this final results subsection, we test the performance of the proposed GSBM variants for the Stokes problem. We first start analysing the static case, i.e. non-moving geometry without considering the dynamic term (first term) in Eq. (40). Here we use the static version of the manufactured solution defined in [91], given by:

$$\boldsymbol{u}(x, y) = \begin{bmatrix} 2\pi y\cos(\pi(x^2 + y^2)) \\ -2\pi x\cos(\pi(x^2 + y^2)) \end{bmatrix}, \tag{51a}$$

$$p(x, y) = \sin(\pi(x^2 + y^2)) - \frac{2}{\pi}. \tag{51b}$$

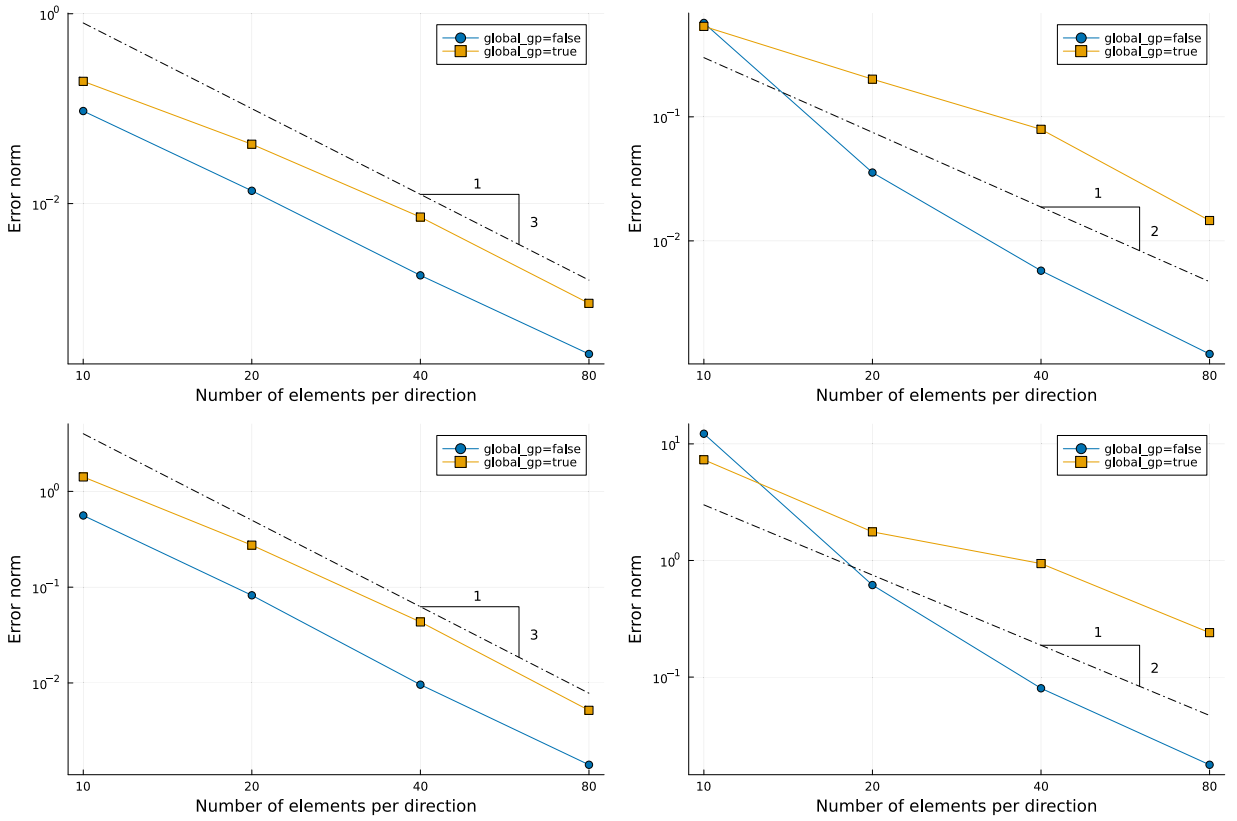


Fig. 21. Convergence in space of the manufactured solution with and without global ghost penalty. L^2-norm (top) and L^∞-norm (bottom) in time of the velocity (left) and pressure (right) L^2 error norm.

We solve the problem in a background domain $D = [-2, 1] \times [-1, 1]$, outside of an embedded circle of radius $R = 0.5$. At the circle boundary, we prescribe a Dirichlet boundary condition weakly enforcing the manufactured solution given in Eq. (51a). For the Stokes case, unlike the previous two problems, we solve the Stokes problem in the exterior part of the geometries. That is, we use the negative of the level-set geometries defined in Eqs. (44)–(46). Since we solve on the exterior of the geometry, we define strong Dirichlet boundary conditions at the left, top, and bottom boundaries of the background domain Ω^h, while we define a Neumann boundary condition at the right boundary. The strong Dirichlet boundary condition is prescribed by the velocity field $\boldsymbol{u}$ as defined in Eq. (51a), and the Neumann boundary condition is defined as: $\mathbf{g} = \boldsymbol{\sigma}(\boldsymbol{u}, p) \cdot \boldsymbol{n}$, with $\boldsymbol{u}$ and p given by Eqs. (51a) and (51b), respectively. For this test, we set the viscosity to $\nu = 0.01$. In this test, we use the algorithmic constant κ appearing in Eq. (17) as $\kappa = 1.0$ in β_2^u and $\kappa = 0.1$ in β_2^p.

As in the other cases, we solve the problem considering the three formulation variants, namely *WSBM*, *SBM* and *O-SBM*, using the definitions of the weighting factor given by Eqs. (12), (21) and (26), respectively. In Fig. 18 we depict the computed velocity and pressure solution using the optimal SBM approach (O-SBM), computed using a background grid of 60×40 cells. In Fig. 19 we show the respective L^2-norm of the error. We see that for both cases, velocity and pressure, the solution error is not concentrated at the boundaries.

4.3.2. Manufactured solution for the transient Stokes problem in a time-dependent domain

One of the main downsides of the SBM is the lack of mass conservation for incompressible flow problems in moving domains, resulting in pressure oscillations in time. To mitigate this issue the Weighted SBM variant was introduced in [76] and later expanded in [77,78]. Therefore, in this section we will only focus on the WSBM variant of the proposed GSBM approach, i.e. we only use the weighted function α_h as defined in Eq. (12).

Here we solve a problem with a time-dependent manufactured analytical solution, similar to the previous case, given by

$$\boldsymbol{u}(x, y) = \begin{bmatrix} 2\pi y \cos(\pi((x-t)^2 + y^2)) \\ -2\pi(x-t)\cos(\pi((x-t)^2 + y^2)) \end{bmatrix}, \tag{52a}$$

$$p(x, y) = \sin(\pi((x-t)^2 + y^2)) - \frac{2}{\pi}. \tag{52b}$$

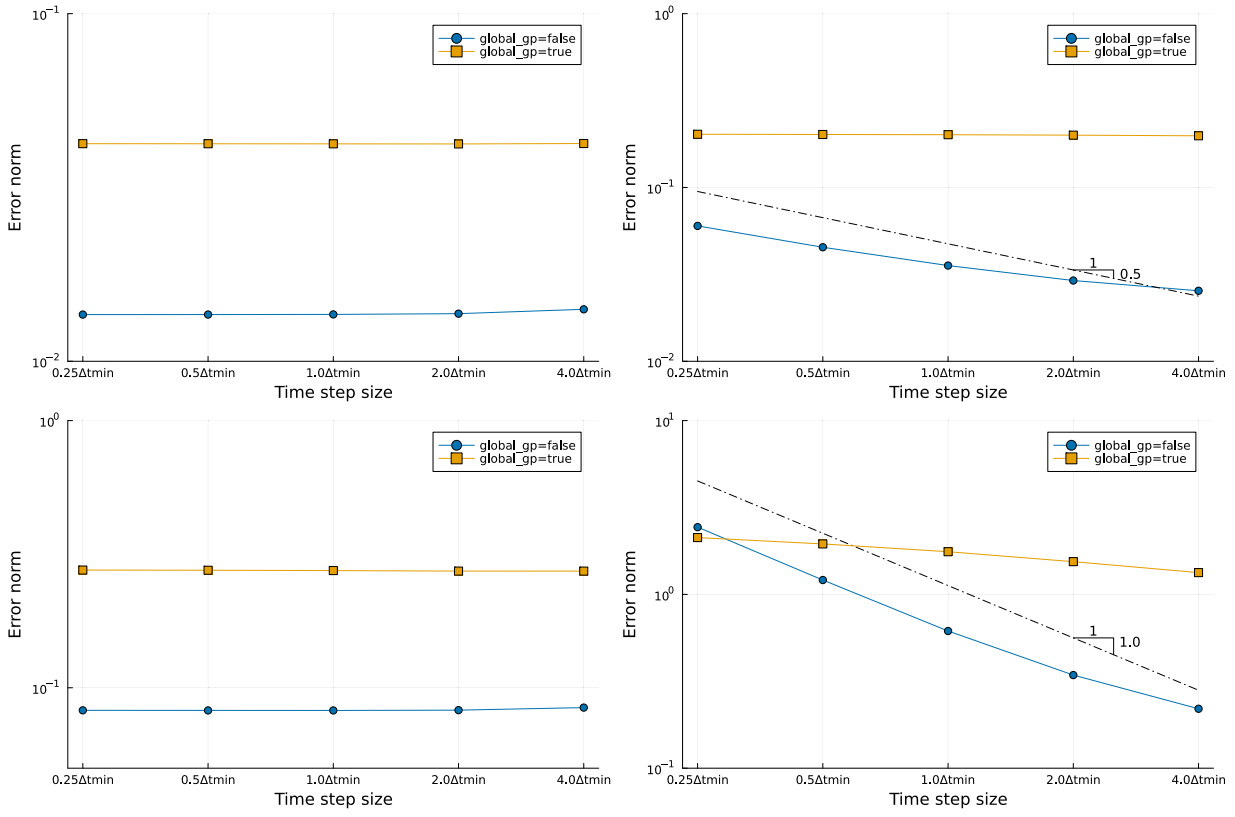


Fig. 22. Sensitivity with respect to the time step size Δt with and without global ghost penalty. L^2-norm (top) and L^∞-norm (bottom) in time of the velocity (left) and pressure (right) L^2 error norms.

The problem is solved in a rectangular background domain $\Omega = [-2, 1] \times [-1, 1]$, around an embedded circle of radius $R = 0.5$ and center x_c moving according to the law

$$\boldsymbol{x}_c(t) = \begin{bmatrix} t \\ 0.0 \end{bmatrix}. \tag{53}$$

Here the same boundary conditions used in the previous case are enforced, i.e. Dirichlet boundaries on the left, top and bottom sides, and Neumann on the right side of the square. We define the maximum initial velocity field as $U^* \overset{\text{def}}{=} \sqrt{(4\pi)^2 + (2\pi)^2}$, which will also be the maximum velocity for $t \leq 1.0$. With this maximum velocity we can compute the equivalent time step size such that the CFL condition is 1.0 for the smallest mesh size considered in this section, i.e. $h^* = 2/80$. This leads to a characteristic step size of $\Delta t^* \overset{\text{def}}{=} \frac{h^*}{U^*}$. The viscosity is set to $\nu = 0.01$ and the problem is solved for $t \in [0, 20\Delta t^*]$. Again, the default algorithmic constants for β_2^u and β_2^p are $\kappa = 1.0$ and $\kappa = 0.1$, respectively. The problem is solved in time using a Gerneralized-α time integration scheme with a spectral radius $\rho_\infty = 0.0$.

To assess the performance of the method we will consider the discrete L^2-norm and L^∞-norm in space and time, defined as:

$$\|e\|_{L^2(L^2(\Omega))} = \left(\Delta t \sum_{i=1}^{N} \|e_i\|^2_{L^2(\Omega)} \right)^{1/2}, \tag{54}$$

$$\|e\|_{L^\infty(L^2(\Omega))} = \max_{i=1,\dots,N} \|e_i\|_{L^2(\Omega)}, \tag{55}$$

with N the number of time steps and $\|e_i\|_{L^2(\Omega)}$ the L^2-norm in space of the solution at step i, given in Eq. (48).

In this section we first investigate what is the effect of the choice of the ghost penalty parameter, β_2, between two cases: a) the definition given in Eq. (17), where $\beta_2 = 0.0$ on the interior edges and denoted by *global_gp=false* in the figures below, or b) the definition proposed in [75], where $\beta_2 \neq 0.0$ on all edges of the domain, denoted by *global_gp=true*.

We first do a convergence study in space considering the two cases, see Fig. 21 where we depict the L^2-norm and L^∞-norm in time for the velocity and pressure fields. For this study we fix the time step size to $\Delta t = \Delta t^*$ for all meshes, a small-enough time step size that avoids the pollution of the spatial error by the time integration error. We start with a coarsest mesh of 15×10 elements and refine successively 3 times. In Fig. 21 we see that for all cases the method preserves optimal accuracy. However, the case with

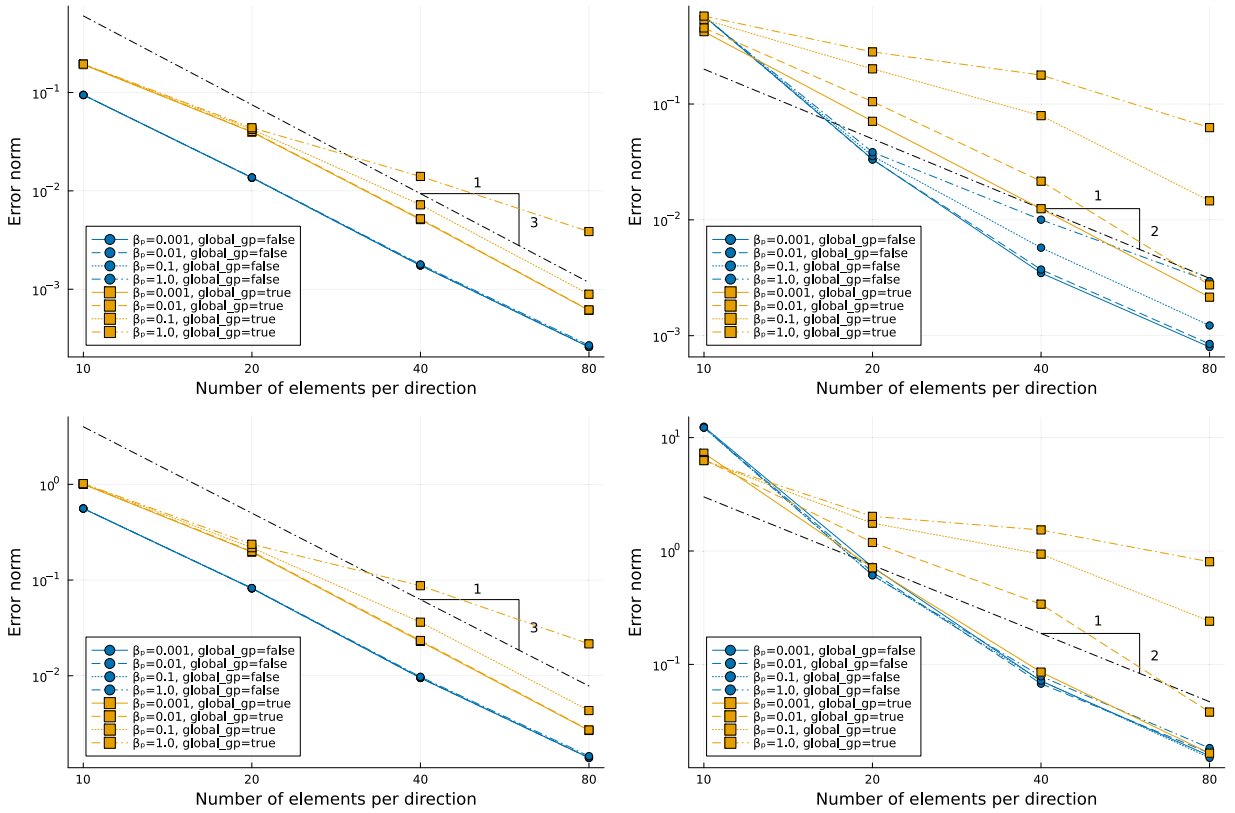


Fig. 23. Sensitivity with respect to the ghost-penalty constant β_2^p. L^2-norm (top) and L^∞-norm (bottom) in time of the velocity (left) and pressure (right) L^2 error norms.

ghost penalty active on all edges (including interior edges) introduces a consistency error, noticed by the constant shift upwards of the *global_gp = true* curve.

Following the convergence in space, we study the behaviour of the method for small time steps. Here we solve the same problem as before in a fixed mesh of 30×20 elements for $t \in [0, 20\Delta^*]$. It has been reported in the literature that for small time step sizes, sharp embedded methods on moving domains for incompressible flows result in spurious pressure oscillations, see for example [75,77,78]. More precisely, in [75] the authors state that adding a ghost penalty term also in interior edges limits the effect of such spurious oscillations. Here we assess this phenomenon by analysing the L^2 and L^∞-norms in time. The former norm averages the spurious pressure oscillation errors, while these error spikes in time remain in the later norm. This effect is clear in Fig. 22, where we see that the spatial error dominates over the time stepping error, leading to constant error rates for the L^2-norm of velocity and pressure, and L^∞-norm of the velocity. However, for the pressure L^∞-norm, the variant with ghost penalty only active on cut and extension edges shows an increase of the norm that is proportional to Δt^{-1}, while adding ghost penalty in all edges results in a constant error independent of Δt. This denotes the presence of pressure oscillations in time for the former case, which also goes in line with the results reported for the same example in [75]. Note that in general we see lower errors for the case of ghost penalty only on cut and extension faces. However, depending on the application, e.g. for applications where a very small time step is required, leading to a very low CFL based on the body motion, the global ghost penalty approach might lead to better results.

The results shown in Fig. 22 can also depend on the values of the algorithmic constant κ appearing in β_2, Eq. (17). To assess the relevance of the choice of this parameter we do a sensitivity study of the algorithmic constant of the pressure penalty β_2^p. Using the same parameters as in the mesh convergence test, in Fig. 23 we report the convergence in L^2 and L^∞-norms of velocity and pressure. It is observed that the case *ghost_penalty = true* has a strong dependency on the algorithmic constant for both, velocity and pressure errors. In contrast, the case of *ghost_penalty = false* results in very little changes in the error, only the L^2-norm of the pressure error is affected by this parameter. We see that the choice of $\kappa = 0.1$ in β_2^p is a good compromise between accuracy and stability of the solution in moving domains.

Another relevant term of the formulation is the *grad-div* stabilization term. This term is controlled by the β_{div} parameter. Therefore, here we study the sensitivity with respect to such parameter, see Fig. 24. Again, here we also consider the two cases as in the previous figure, namely a non-global ghost penalty choice (*global_gp = false*) and the global ghost penalty definition (*global_gp = true*). In this case we solve the problem in a fixed mesh of 30×20 elements, using a time step size of $\Delta t = \Delta t^*$ from $t = 0$ to $t = 20\Delta t^*$. We assess the influence of β_{div} on the L^2-norm (top) and L^∞-norm (bottom) in time of the velocity (left) and pressure (right) L^2 error norms.

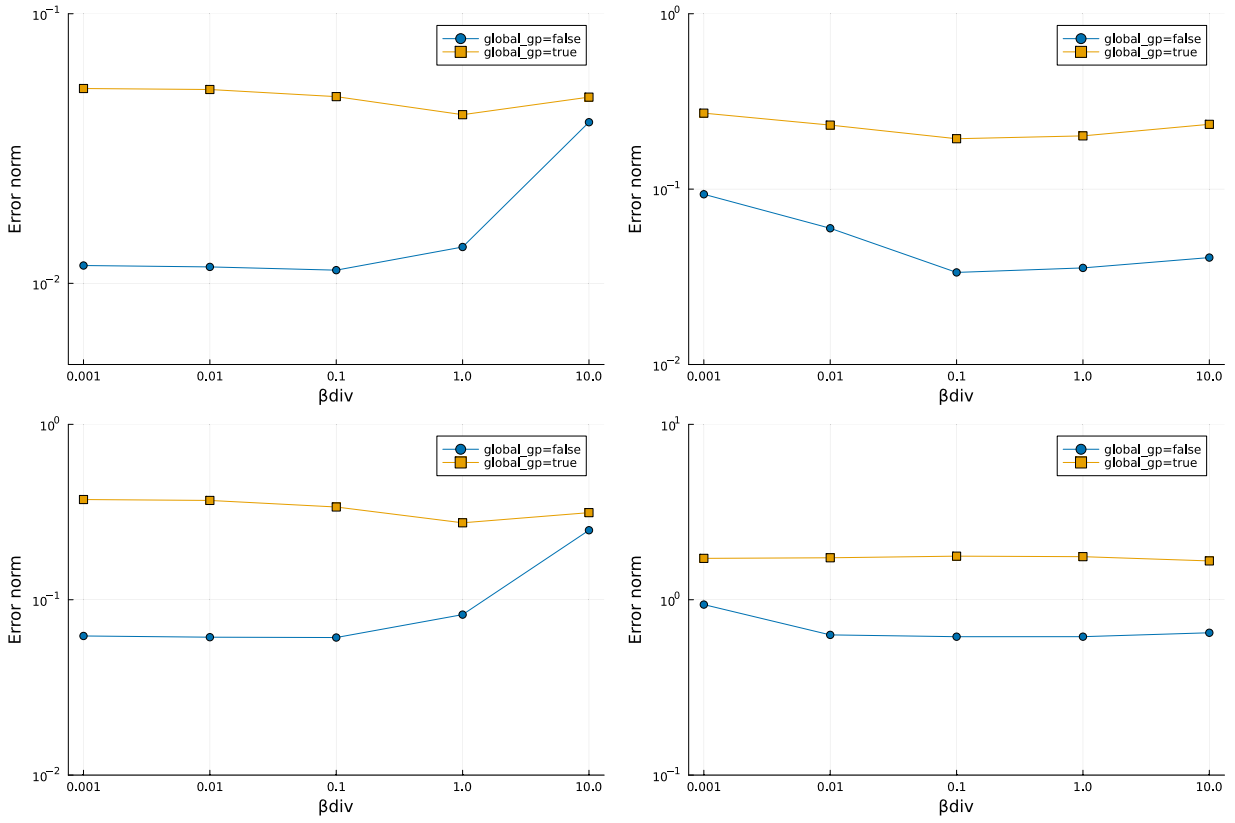


Fig. 24. Sensitivity with respect to the grad-div constant β_{div} with and without global ghost penalty. L^2-norm (top) and L^∞-norm (bottom) in time of the velocity (left) and pressure (right) L^2 error norm.

In Fig. 24 we see that a larger value of β_{div} results in lower error for the pressure field, but for $\beta_{\text{div}} > 1.0$ the error increases. For the velocity field, the error remains constant until $\beta_{\text{div}} = 1.0$, for larger values an error increase is also observed. Based on these results, the value chosen for the previous tests, i.e. $\beta_{\text{div}} = 1.0$, is deemed appropriate.

4.3.3. Oscillating cylinder in crossflow

Here we consider the simulation of an oscillating cylinder in cross flow. This is a well-studied benchmark for unfitted finite element formulations in problems with moving domains, see [75,77]. The goal of this test is to assess how different algorithmic parameters introduced in the proposed formulation affect the presence of spurious pressure oscillations in time. Particularly, we focus on the effect of considering a global definition of the ghost penalty or only being active in the cut and extension region. Again, in this test we only consider the WSBM case, since the other variants (SBM and O-SBM) are not suitable for problems with moving domains. We refer the reader to [77] for a more detailed analysis of the equivalent WSBM formulation, including other obstacle geometries and flow conditions. In this test case we consider a cylinder of radius R = 1.0, embedded in a rectangular domain of size $[-10, 6] \times [-6, 6]$, see Fig. 25.

As in the previous example, we use a viscosity of $\nu = 0.01$. The cylinder oscillates with a frequency of $f_0 = 0.2237$Hz, with a motion amplitude of $y_0 = 0.5$. The position and velocity of the oscillating cylinder are defined by

$$\boldsymbol{x}_c(t) = \begin{bmatrix} -5.0 \\ y_0 \cos(2\pi f_0 t) \end{bmatrix}, \tag{56}$$

$$\boldsymbol{u}_c(t) = \begin{bmatrix} 0.0 \\ -U_0 \sin(2\pi f_0 t) \end{bmatrix}, \tag{57}$$

with a velocity amplitude of $U_0 = 2\pi f_0$. At the left side of the rectangular domain we enforce an inlet velocity of $u_{\text{left}} = [1.0, 0.0]^T$, at the top and bottom boundaries we enforce free slip flow, i.e. $\boldsymbol{u} \cdot \boldsymbol{n} = 0.0$, and at the right boundary a free outlet boundary condition is assumed, i.e $\sigma \cdot \boldsymbol{n} = [0.0, 0.0]^T$. We run the simulation for two oscillation periods, $t = [0, 2T]$ with $T = 1/f_0$. The problem is solved in a cartesian background grid of 80×60 elements, leading to a characteristic element size of $h = 0.2$. The default algorithmic constants for β_2^u and β_2^p are $\kappa = 1.0$ and $\kappa = 0.1$, respectively and here we select $\beta_{\text{div}} = 1.0$. The problem is solved in time using a Genralized-α time integration scheme with a spectral radius $\rho_\infty = 0.0$.

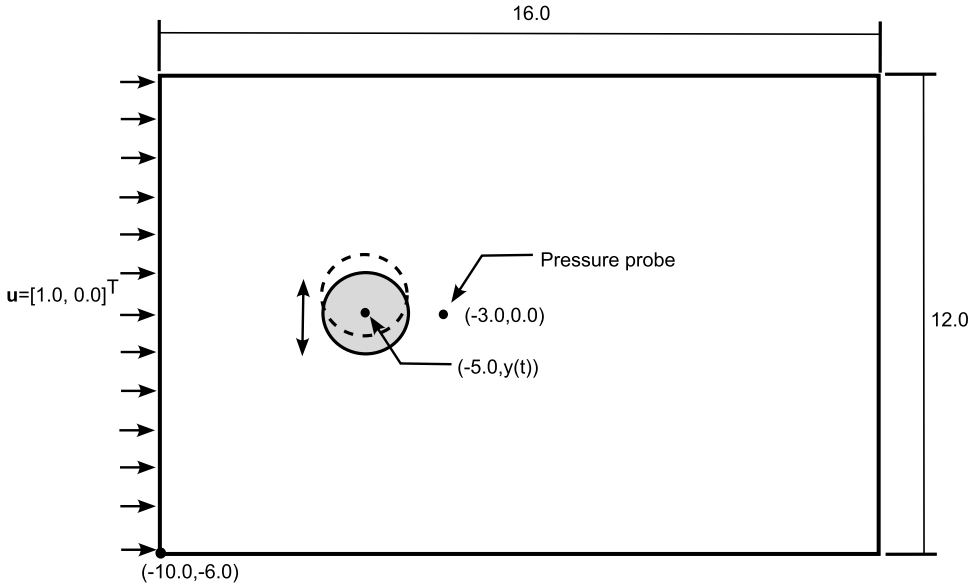


Fig. 25. Sketch of the geometry of the oscillating cylinder in cross flow problem.

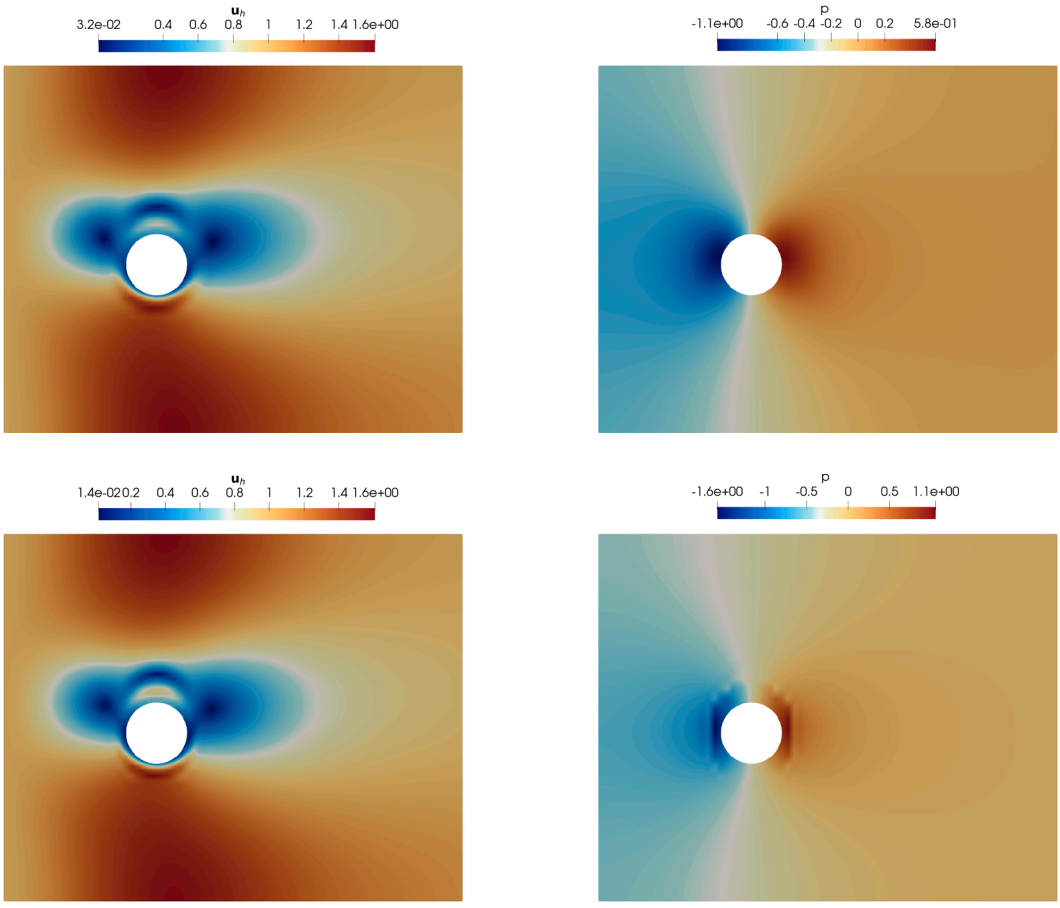


Fig. 26. Velocity (left) and pressure (right) fields at $t = 0.5T$ for the case with global ghost penalty (top) and ghost penalty active only at the cut and extension region (bottom).

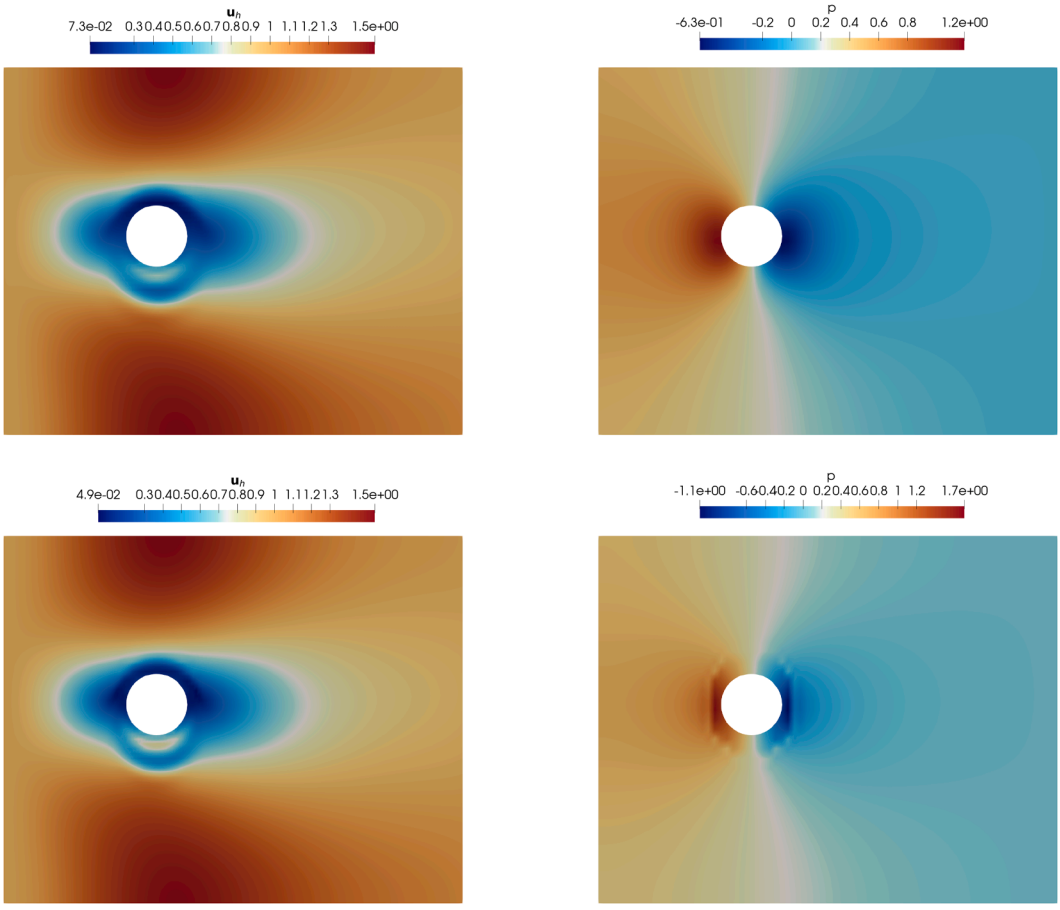


Fig. 27. Velocity (left) and pressure (right) fields at $t = T$ for the case with global ghost penalty (top) and ghost penalty active only at the cut and extension region (bottom).

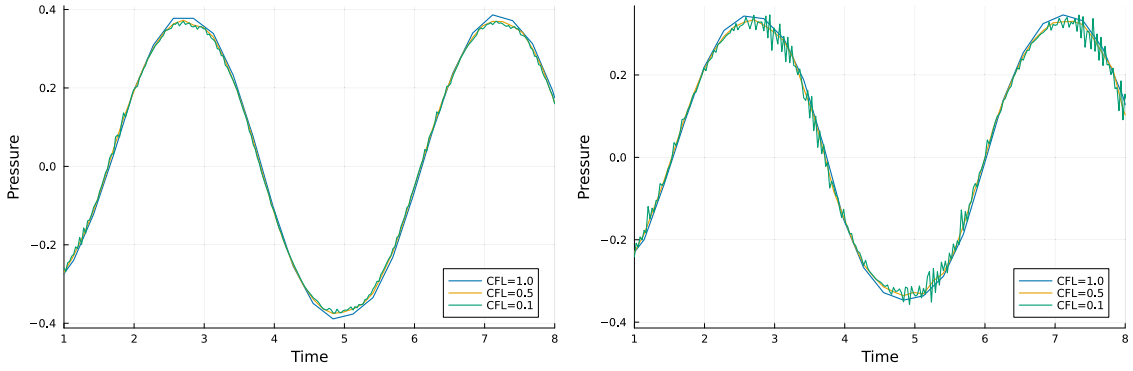


Fig. 28. Pressure value at point $[-3.0, 0.0]$ for the case *global_gp = true* (left) and *global_gp = false* (right) using different time step sizes equivalent to CFL $= \{1.0, 0.5, 0.1\}$.

Here we consider the solution of the problem using the non-global ghost penalty choice (*global_gp = false*) and the global ghost penalty definition (*global_gp = true*). We solve the problem with four different time step sizes corresponding to global CFL values of CFL $= \{1.0, 0.5, 0.1\}$, with CFL $= \frac{U_0 \Delta t}{h}$ defined based on characteristic cylinder velocity and element size. In Figs. 26 and 27 we depict the velocity and pressure fields of the two cases, *global_gp = true* and *global_gp = false*, for CFL $= 0.1$ at $t = 0.5T$ and $t = T$, respectively. We observe that for the velocity field there are very little differences. Looking at the pressure field, the case with *global_gp = false* results in a localized pressure difference close to the embedded boundary, while for the case with *global_gp = false* the pressure field

is smooth throughout the domain. This phenomena is expected due to the global smoothening effect introduced by adding a ghost penalty in all the edges of the domain, and goes in line with results reported in the literature, see for example [75].

In addition to the velocity and pressure fields at different time instances, we can also assess what is the behaviour of the pressure in time. For this, we plot the pressure value at point $[-3.0, 0.0]$ within the time interval $t = [1.0, 8.0]$, to exclude flow initialization effects, for different time step sizes equivalent to CFL $= \{1.0, 0.5, 0.1\}$, see Fig. 28. In the figure we see that for moderately large time step size, i.e. CFL$= 0.5$ or CFL$= 1.0$, there are no spurious pressure oscillations in time, neither for the case with global ghost penalty nor for the case where the ghost penalty term is added only at the cut and extension region. When the time step is reduced to an equivalent CFL$= 0.1$ there appear spurious pressure oscillations. The oscillations are much lower for the global ghost penalty case (left) than for the non-global ghost penalty (right). This result goes in line with the results observed in the previous test case, as shown in Fig. 22, where we see that the L^∞-norm of the pressure increases as the time step is reduced for the case without global ghost penalty and remains almost constant when the global ghost penalty is used. We would also like to highlight that these oscillations go in line with the observations in many studies in the literature [75,77,78], and associated to the lack of exact mass conservation of the extended velocity field.

Nonetheless, we also note that in general lower errors are observed for the case of ghost penalty only on cut and extension faces, see Section 4.3.2. Thus, depending on the application, e.g. for applications where a very small time step is required, leading to a very low CFL based on the body motion, the global ghost penalty approach might lead to better results.

5. Conclusions

In this work we present a generalized formulation, the Generalized Shifted Boundary Method, for geometry-parametric PDEs. The proposed GSBM formulation does not require the re-definition of FE spaces, integration rules or geometrical entities for varying or moving geometries. We showed that the proposed formulation reduces to the original Shifted Boundary Method and its weighted version (WSBM) for a particular choice of the weighting factors.

We demonstrated the performance of the proposed approach for three different model problems: a) Poisson problem, b) linear elasticity problem, and c) transient Stokes problem. For all of them, we demonstrate that the optimal rates of convergence are preserved for a variety of geometries and problems with topological changes. For the transient Stokes problem, we also confirmed the suitability of the methodology for problems with moving domains.

In future works, we plan to extend the proposed methodology to nonlinear problems, e.g. Navier-Stokes equations, and fluid-structure interaction problems.

Data availability

Data will be made available on request. The code used to generate this paper is publicly available in this repository [92].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Oriol Colomés: Conceptualization, Methodology, Software, Formal analysis, Investigation, Resources, Writing - original draft, Visualization, Funding acquisition; **Jan Modderman:** Writing - original draft, Visualization, Software, Methodology, Investigation, Conceptualization; **Guglielmo Scovazzi:** Writing - review & editing, Investigation, Conceptualization.

Acknowledgments

This publication is part of the project DigiOcean4Solar with file number 21225 of the research programme NWO Talent Programme Vidi AES 2023 which is financed by the Dutch Research Council (NWO) under the grant ID https://doi.org/10.61686/OPCTU16570. O. Coloméa gratefully acknowledges this NWO funding. G. Scovazzi gratefully acknowledges the support of the U.S. National Science Foundation under Grants DMS-2207164 and DMS-2409919 (Computational Mathematics Program, Division of Mathematical Sciences).

References

[1] C.A. Taylor, C.A. Figueroa, Patient-specific modeling of cardiovascular mechanics, Annu. Rev. Biomed. Eng. 11 (1) (2009) 109–134.
[2] A. Donizetti, T. Bellosta, A. Rausa, B. Re, A. Guardone, Level-set mass-conservative front-tracking technique for multistep simulations of in-flight ice accretion, J. Aircr. 60 (4) (2023) 1050–1060.
[3] 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.
[4] J. Wang, S. Santhosh, O. Colomés, M. Capaldo, L. Yang, Experimental study of dynamic response of passive flapping hydrofoil in regular wave, Phys. Fluids 35 (7) (2023).

[5] 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.
[6] 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.
[7] E. Burman, P. Hansbo, Fictitious domain methods using cut elements: III. a stabilized Nitsche method for Stokes' problem, ESAIM: Math. Model. Numer. Anal. 48 (3) (2014) 859–874.
[8] B. Schott, W.A. Wall, A new face-oriented stabilized XFEM approach for 2D and 3D incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 276 (2014) 233–265.
[9] M. Lesueur, H. Rattez, O. Colomés, μCT scans permeability computation with an unfitted boundary method to improve coarsening accuracy, Comput. Geosci. 166 (2022) 105118.
[10] T.E. Tezduyar, Finite element methods for flow problems with moving boundaries and interfaces, Arch. Comput. Methods Eng. 8 (2001) 83–130.
[11] D. Kim, H. Choi, Immersed boundary method for flow around an arbitrarily moving body, J. Comput. Phys. 212 (2) (2006) 662–680.
[12] 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.
[13] 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.
[14] R. Codina, J. Baiges, I. Castañar, I. Martínez-Suárez, L. Moreno, S. Parada, An embedded strategy for large scale incompressible flow simulations in moving domains, J. Comput. Phys. 488 (2023) 112181.
[15] J. Gabbard, W.M. van Rees, A high-order finite difference method for moving immersed domain boundaries and material interfaces, J. Comput. Phys. 507 (2024) 112979.
[16] L.T. Zhang, M. Gay, Immersed finite element method for fluid-structure interactions, J. Fluids Struct. 23 (6) (2007) 839–857.
[17] J. Yang, S. Preidikman, E. Balaras, A strongly coupled, embedded-boundary method for fluid–structure interactions of elastically mounted rigid bodies, J. Fluids Struct. 24 (2) (2008) 167–182.
[18] 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.
[19] F.-B. Tian, H. Dai, H. Luo, J.F. Doyle, B. Rousseau, Fluid–structure interaction involving large deformations: 3D simulations and applications to biological systems, J. Comput. Phys. 258 (2014) 451–469.
[20] V. Lakshminarayan, C. Farhat, A. Main, An embedded boundary framework for compressible turbulent flow and fluid–structure computations on structured and unstructured grids, Int. J. Numer. Methods Fluids 76 (6) (2014) 366–395.
[21] 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.
[22] C.E. Kees, J.H. Collins, A. Zhang, Simple, accurate, and efficient embedded finite element methods for fluid–solid interaction, Comput. Methods Appl. Mech. Eng. 389 (2022) 114404.
[23] G. Wang, Y. Hong, S.H. Huo, C. Jiang, An immersed edge-based smoothed finite element method with the stabilized pressure gradient projection for fluid–structure interaction, Comput. Struct. 270 (2022) 106833.
[24] J. Modderman, O. Colomés, Application of unfitted finite element methods for estimating added mass and added damping in floating structures, Adv. Comput. Sci. Eng. 4 (2025) 142–167.
[25] J. Modderman, O. Colomés, Estimation of hydrodynamic coefficients of floating offshore structures using the aggregated unfitted finite element method, in: ISOPE International Ocean and Polar Engineering Conference, ISOPE, 2025, pp. ISOPE–I.
[26] J. Parvizian, A. Düster, E. Rank, Finite cell method: h-and p-extension for embedded domain problems in solid mechanics, Comput. Mech. 41 (1) (2007) 121–133.
[27] C. Annavarapu, M. Hautefeuille, J.E. Dolbow, Stable imposition of stiff constraints in explicit dynamics for embedded finite element methods, Int. J. Numer. Methods Eng. 92 (2) (2012) 206–228.
[28] T. Rüberg, F. Cirak, J.M. García Aznar, An unstructured immersed finite element method for nonlinear solid mechanics, Advanced Modeling and Simulation in Engineering Sciences 3 (2016) 1–28.
[29] N.M. Atallah, C. Canuto, G. Scovazzi, The shifted boundary method for solid mechanics, Int. J. Numer. Methods Eng. 122 (20) (2021) 5935–5970.
[30] S. Cotin, M. Duprez, V. Lleras, A. Lozinski, K. Vuillemot, ϕ-FEM: An efficient simulation tool using simple meshes for problems in structure mechanics and heat transfer, Partition of Unity Methods (2023) 191–216.
[31] T. Belytschko, T. Black, Elastic crack growth in finite elements with minimal remeshing, Int. J. Numer. Methods Eng. 45 (5) (1999) 601–620.
[32] C.A. Duarte, O.N. Hamzeh, T.J. Liszka, W.W. Tworzydlo, A generalized finite element method for the simulation of three-dimensional dynamic crack propagation, Comput. Methods Appl. Mech. Eng. 190 (15–17) (2001) 2227–2262.
[33] S. Soghrati, A.M. Aragón, C. Armando Duarte, P.H. Geubelle, An interface-enriched generalized FEM for problems with discontinuous gradient fields, Int. J. Numer. Methods Eng. 89 (8) (2012) 991–1008.
[34] R. Rangarajan, M.M. Chiaramonte, M.J. Hunsweck, Y. Shen, A.J. Lew, Simulating curvilinear crack propagation in two dimensions with universal meshes, Int. J. Numer. Methods Eng. 102 (3–4) (2015) 632–670.
[35] K. Li, A. Rodríguez-Ferran, G. Scovazzi, The simple shifted fracture method, Int. J. Numer. Methods Eng. 124 (12) (2023) 2837–2875.
[36] J. Parvizian, A. Düster, E. Rank, Topology optimization using the finite cell method, Optimization and Engineering 13 (1) (2012) 57–78.
[37] D. Makhija, K. Maute, Numerical instabilities in level set topology optimization with the extended finite element method, Struct. Multidiscip. Optim. 49 (2014) 185–197.
[38] Z.J. Wegert, J. Manyer, C. Mallon, S. Badia, V.J. Challis, Level-set topology optimisation with unfitted finite elements and automatic shape differentiation,(2025). arXiv:2504.09748
[39] E. Neiva, S. Badia, A.F. Martín, M. Chiumenti, A scalable parallel finite element framework for growing geometries. Application to metal additive manufacturing, Int. J. Numer. Methods Eng. 119 (11) (2019) 1098–1125.
[40] M. Carraturo, S. Kollmannsberger, A. Reali, F. Auricchio, E. Rank, An immersed boundary approach for residual stress evaluation in selective laser melting processes, Addit. Manuf. 46 (2021) 102077.
[41] 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.
[42] 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.
[43] F. de Prenter, C.V. Verhoosel, E.H. van Brummelen, M.G. Larson, S. Badia, Stability and conditioning of immersed finite element methods: analysis and remedies, Arch. Comput. Methods Eng. 30 (6) (2023) 3617–3656.
[44] E. Burman, Ghost penalty, Comptes Rendus. Mathématique 348 (21–22) (2010) 1217–1220.
[45] S. Badia, E. Neiva, F. Verdugo, Linking ghost penalty and aggregated unfitted methods, Comput. Methods Appl. Mech. Eng. 388 (2022) 114232.
[46] 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.
[47] S. Badia, A.F. Martín, E. Neiva, F. Verdugo, The aggregated unfitted finite element method on parallel tree-based adaptive meshes, SIAM J. Sci. Comput. 43 (3) (2021) C203–C234.
[48] E. Neiva, S. Badia, Robust and scalable h-adaptive aggregated unfitted finite elements for interface elliptic problems, Comput. Methods Appl. Mech. Eng. 380 (2021) 113769.
[49] S. Gross, A. Reusken, Analysis of optimal preconditioners for cutFEM, Numerical Linear Algebra with Applications 30 (5) (2023) e2486.

[50] S. Zahedi, A space-time cut finite element method with quadrature in time, in: Geometrically Unfitted Finite Element Methods and Applications: Proceedings of the UCL Workshop 2016, Springer, 2018, pp. 281–306.
[51] 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.
[52] M.G. Larson, S. Zahedi, Stabilization of high order cut finite element methods on surfaces, IMA Journal of Numerical Analysis 40 (3) (2020) 1702–1745.
[53] P.A. Martorell, S. Badia, High order unfitted finite element discretizations for explicit boundary representations, J. Comput. Phys. 511 (2024) 113127.
[54] S. Badia, E. Neiva, F. Verdugo, Robust high-order unfitted finite elements by interpolation-based discrete extension, Computers & Mathematics with Applications 127 (2022) 105–126.
[55] S. Badia, P.A. Martorell, F. Verdugo, Geometrical discretisations for unfitted finite elements on explicit boundary representations, J. Comput. Phys. 460 (2022) 111162.
[56] B. Müller, F. Kummer, M. Oberlack, Highly accurate surface and volume integration on implicit domains by means of moment-fitting, Int. J. Numer. Methods Eng. 96 (8) (2013) 512–528.
[57] X. Zeng, T. Song, G. Scovazzi, A shifted boundary method for the compressible Euler equations, J. Comput. Phys. 520 (2025) 113512.
[58] 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.
[59] N.M. Atallah, K. Mittal, G. Scovazzi, V.Z. Tomov, A high-order shifted interface method for Lagrangian shock hydrodynamics, J. Comput. Phys. 523 (2025) 113637.
[60] N.M. Atallah, G. Scovazzi, Nonlinear elasticity with the shifted boundary method, Comput. Methods Appl. Mech. Eng. 426 (2024) 116988.
[61] K. Li, J.G. Michopoulos, A. Iliopoulos, J.C. Steuben, G. Scovazzi, Complex-geometry simulations of transient thermoelasticity with the shifted boundary method, Comput. Methods Appl. Mech. Eng. 418 (2024) 116461.
[62] L. Nouveau, M. Ricchiuto, G. Scovazzi, High-order gradients with the shifted boundary method: an embedded enriched mixed formulation for elliptic PDEs, J. Comput. Phys. 398 (2019) 108898.
[63] N.M. Atallah, C. Canuto, G. Scovazzi, The high-order shifted boundary method and its analysis, Comput. Methods Appl. Mech. Eng. 394 (2022) 114885.
[64] R. Zorrilla, R. Rossi, G. Scovazzi, C. Canuto, A. Rodríguez-Ferran, A shifted boundary method based on extension operators, Comput. Methods Appl. Mech. Eng. 421 (2024) 116782.
[65] N. Antonelli, R. Aristio, A. Gorgi, R. Zorrilla, R. Rossi, G. Scovazzi, R. Wüchner, The shifted boundary method in isogeometric analysis, Comput. Methods Appl. Mech. Eng. 430 (2024) 117228.
[66] 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.
[67] E.N. Karatzas, G. Stabile, L. Nouveau, G. Scovazzi, G. Rozza, A reduced-order shifted boundary method for parametrized incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 370 (2020) 113273.
[68] 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.
[69] 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) (2019) 492–518. https://doi.org/10.1002/nme.6231
[70] 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.
[71] 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.
[72] K. Li, A. Gorgi, R. Rossi, G. Scovazzi, The shifted boundary method for contact problems, Comput. Methods Appl. Mech. Eng. 440 (2025) 117940.
[73] 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.
[74] J. Lee, J. Kim, H. Choi, K.-S. Yang, Sources of spurious force oscillations from an immersed boundary method for moving-body problems, J. Comput. Phys. 230 (7) (2011) 2677–2695.
[75] M. Olshanskii, H. von Wahl, Stability of instantaneous pressures in an Eulerian finite element method for moving boundary flow problems, J. Comput. Phys. 533 (2025) 113983.
[76] O. Colomés, A. Main, L. Nouveau, G. Scovazzi, A weighted shifted boundary method for free surface flow problems, J. Comput. Phys. 424 (2021) 109837.
[77] 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.
[78] G. Scovazzi, D. Xu, O. Colomés, A. Main, K. Li, N.M. Atallah, N. Abboud, A weighted shifted boundary method for the Navier-Stokes equations with immersed moving boundaries, Available at SSRN 5251653 (2025).
[79] M. Duprez, V. Lleras, A. Lozinski, V. Vigon, K. Vuillemot, Phi-FEM-FNO: a new approach to train a neural operator as a fast PDE solver for variable geometries,(2025), arXiv:2502.10033.
[80] S. Brivio, S. Fresca, A. Manzoni, Handling geometrical variability in nonlinear reduced order modeling through continuous geometry-Aware DL-ROMs, Comput. Methods Appl. Mech. Eng. 442 (2025) 117989.
[81] J. Nitsche, "Uber ein variationsprinzip zur l"osung von Dirichlet-Problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind, in: Abhandlungen Aus Dem Mathematischen Seminar Der Universit"At Hamburg, 36, Springer, 1971, pp. 9–15.
[82] 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. ibinfodoi10.1016/j.jcp.2017.10.026
[83] N. Atallah, C. Canuto, G. Scovazzi, Analysis of the weighted shifted boundary method for the Poisson and Stokes problems, Available at SSRN 5183772 (2025).
[84] 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.
[85] N.M. Atallah, C. Canuto, G. Scovazzi, Analysis of the shifted boundary method for the Poisson problem in general domains, arXiv:2006.00872 (2020).
[86] 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.
[87] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. 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.
[88] 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.
[89] 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.
[90] P.A. Martorell, S. Badia, STLCutters. jl: a scalable geometrical framework library for unfitted finite element discretisations, 309 (2025) 109479.
[91] H. von Wahl, T. Richter, C. Lehrenfeld, An unfitted Eulerian finite element method for the time-dependent Stokes problem on moving domains, IMA J. Numer. Anal. 42 (3) (2022) 2505–2544.
[92] C. Oriol M. Jan TheGeneralizedSBM.jl A julia-based implementationof the Generalized Shifted Boundary Method using Gridap.jl August 2025.