The Shifted Boundary Method in Isogeometric Analysis

Nicolò Antonelli[a,b], Ricky Aristio[c], Andrea Gorgi[a,b], Rubén Zorrilla[a,b,*], Riccardo Rossi[a,b], Guglielmo Scovazzi[d], Roland Wüchner[c]

[a]*Departament d'Enginyeria Civil i Ambiental, Universitat Polit*
ΩΩimmediateècnica de Catalunya, Barcelona, 08034, Spain
[b] International Center for Numerical Methods in Engineering (CIMNE), Barcelona, 08034, Spain
[c]Institute of Structural Analysis, Technische Universit
ΩΩimmediateät Braunschweig, Braunschweig, 38106, Germany
[d]Department of Civil and Environmental Engineering, Duke University, Durham, NC 27708, USA

Abstract

This work presents a novel application of the Shifted Boundary Method (SBM) within the Isogeometric Analysis (IGA) framework, applying it to two-dimensional and three-dimensional Poisson problems with Dirichlet and Neumann boundary conditions. The SBM boundary condition imposition is achieved by means of a fully penalty-free formulation, eliminating the need for penalty calibration. The numerical experiments demonstrate how order elevation, coupled with SBM through higher-order Taylor expansions, consistently achieves optimal convergence rates. Additionally, analyzing the condition number of the problem matrix reveals that SBM, when integrated with IGA, effectively circumvents the small cut-cell problem, a common issue in numerical methods with unfitted boundaries.

Keywords: Shifted Boundary Method (SBM), Isogeometric Analysis (IGA), Isogeometric Boundary Representation Analysis (IBRA), Unfitted Boundary Methods, Small cut-cell problem

1. Introduction

The advent of Isogeometric Analysis (IGA) has revolutionized computational mechanics by bridging the gap between Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE). IGA was originally introduced by Hughes *et al.* [1] and later explored by Cottrell *et al.* [2], Bazilevs *et al.* [3, 4] and Zhang *et al.* [5]. IGA enables the exact representation of a wide range of geometries, from simple to complex shapes. Furthermore, IGA facilitates high continuity across knot spans [6], which is pivotal for accurately capturing the physical behavior of materials and structures. The continuity and exact representation afforded by IGA facilitate the analysis of numerical simulation involving complex geometrical features, providing at the same time a robust tool for practitioners to explore innovative design solutions [7, 8, 9].

At the heart of IGA is the Cox de-Boor formula, which sets the basis for defining B-Splines curves. This formula is extendable to define B-Spline surfaces through tensor product operations on B-Spline curves and underscores the adaptability and precision of IGA in representing complex geometries. B-Spline basis functions, which are a subset of NURBS (Non-Uniform Rational B-Splines) basis functions, offer several advantages over traditional finite element basis functions. They can achieve continuity levels up to $\mathscr{C}^{p-1}$, where p is the order of the basis functions, ensuring smooth transitions across elements. In addition, while maintaining local support, they enable the implementation of local refinement techniques [10, 11].

*Corresponding author
Email addresses: nantonelli@cimne.upc.edu (Nicolò Antonelli), ricky.aristio@tu-braunschweig.de (Ricky Aristio), agorgi@cimne.upc.edu (Andrea Gorgi), rzorrilla@cimne.upc.edu (Rubén Zorrilla), rrossi@cimne.upc.edu (Riccardo Rossi), guglielmo.scovazzi@duke.edu (Guglielmo Scovazzi), r.wuechner@tu-braunschweig.de (Roland Wüchner)

However, the simulation of complex geometries with IGA introduces a notable level of complexity, requiring sophisticated computational techniques and a deep understanding of both geometric modeling and numerical analysis. Besides, the creation of complex geometries likely involves joining multiple NURBS patches. This introduces an extra difficulty when creating watertight geometries from the coupling of multiple NURBS patches [12, 13].

In [14, 15] the concept of T-Splines was introduced in order to deal with T-junctions. T-junctions eliminate the tensor-product restriction inherent to NURBS patches, enabling a more efficient location of the control points. Furthermore, they can also achieve watertight surfaces, even in the case of complex geometries, by using Analysis-Suitable T-Splines (ASTS) [16, 17]. Other spline alternatives that are worth mentioning are those found in [18, 19, 20], all of which aim to represent the exact boundaries of the domain.

In what follows, by the term body-fitted, or boundary-fitted, we mean the family of approaches that consider a discretization matching the boundaries of the domain of interest. All the IGA techniques cited so far can be considered within this family of methods, including [21, 22] for the treatment of trimmed models. These boundary-fitted approaches can handle many complex geometries, achieving optimal convergence and allowing for local refinement. However, when the complexity of the geometry is high, these methods may not have straightforward implementations in industrial applications, if the requisite of global high regularity of the basis functions needs to be maintained. Due to these limitations, [23, 24] introduced immersed boundary IGA methods, by combining IGA with the Finite Cell Method (FCM) [25, 26]. The family of immersogeometric analysis methods compromises on representing the geometry by immersing it as a non-boundary-fitted discretization in the background domain [27, 28, 29], which has many applications, particularly in solving fluid-structure interaction problems. We would like to specifically mention the Isogeometric Boundary Representation Analysis (IBRA) [30] which can also be considered within the family of immersed methods in IGA, by using the boundary representation (B-Rep) of the CAD model along with the basis functions as in IGA for the approximation of solution fields. It was initially applied for shell analysis of trimmed multipatch models but can easily be expanded to any other problems. The core of IBRA lies in its ability to handle complex geometries, including trimmed surfaces, within a robust design-through-analysis workflow [31, 32], thus reducing pre-processing times significantly. Furthermore, IBRA framework for three-dimensional analysis of solids was established in [33, 34].

Despite its numerous advantages, IBRA's application has drawbacks, mainly related to computational performance. Integrating the trimmed knot spans is complex and expensive, and the associated small cut-cell problem [35, 36] likely precludes the use of efficient iterative linear solvers. This small cut-cell problem results in large and potentially unbounded condition numbers of the associated linear system. As the trimmings approach infinitesimal sizes, the condition number increases significantly.

To avoid the small cut-cell problem and simplify the integration of the trimmed knot spans we propose the use of the Shifted Boundary Method (SBM). The SBM was initially introduced in [37, 38] and belongs to the family of approximate domain methods. The key idea in the SBM is to shift the location where boundary conditions are applied from the true boundary to a surrogate boundary composed of the edges of the grid. At the same time, also the boundary condition values are shifted (modified) by means of Taylor expansions in order to preserve optimal accuracy.

So far, the SBM has been applied only in the Finite Element Method (FEM) context, where it showed optimal convergence rates and proved to be free from the small cut-cell problem, contrary to the family of CutFEM methods [35, 39]. Due to its unfitted nature, the SBM avoids the need for mesh alignment with domain boundaries, thereby significantly simplifying both mesh generation and refinement. Furthermore, it can be adapted to support dynamic geometries, shape optimization, and large deformations without the necessity for re-meshing. Numerous studies were conducted across various fields using the SBM. For instance, the authors in [40, 41] explored the application of the SBM in linear and finite elasticity, while in [42], the method was applied to the incompressible Navier-Stokes equations. Additionally [43, 44, 45, 46] present various techniques to enhance the SBM all of them within the FEM context. In particular, in [47] the SBM was generalized to incorporate extension operators alternative to the Taylor expansions.

Our contribution explores the application of the SBM within the context of Isogeometric Analysis, an area of research still uncharted. Our approach becomes similar to that of IBRA, but without any trimming techniques. This allows to superimpose the true boundary onto the parameter space, treating it as an unfitted boundary using the SBM.

The paper is organized as follows: Section 2 introduces the Nitsche and penalty-free weak formulations of the Poisson Problem. Section 3 introduces the SBM method employing arbitrary order Taylor expansions on a Cartesian grid. It further elaborates on the optimal surrogate boundary definition and the conceptual algorithm. A variety of

numerical examples are provided in Section 4. A summary of findings concludes the article, in Section 5.

2. Review of the SBM for high-order Finite Element discretizations

2.1. Poisson's problem: governing equations

Let us consider a domain $\Omega \in \mathbb{R}^{n_d}$ ($n_d = 2, 3$ indicates the number of spatial dimensions) with Lipschitz boundary Γ. As a prototypical scenario and a point of departure in the discussion of the main ideas, we consider the Poisson problem with non-homogeneous Dirichlet and Neumann boundary conditions

$$-\Delta u = f \quad \text{in } \Omega, \tag{1}$$

$$u = u_D \quad \text{on } \Gamma_D, \tag{2}$$

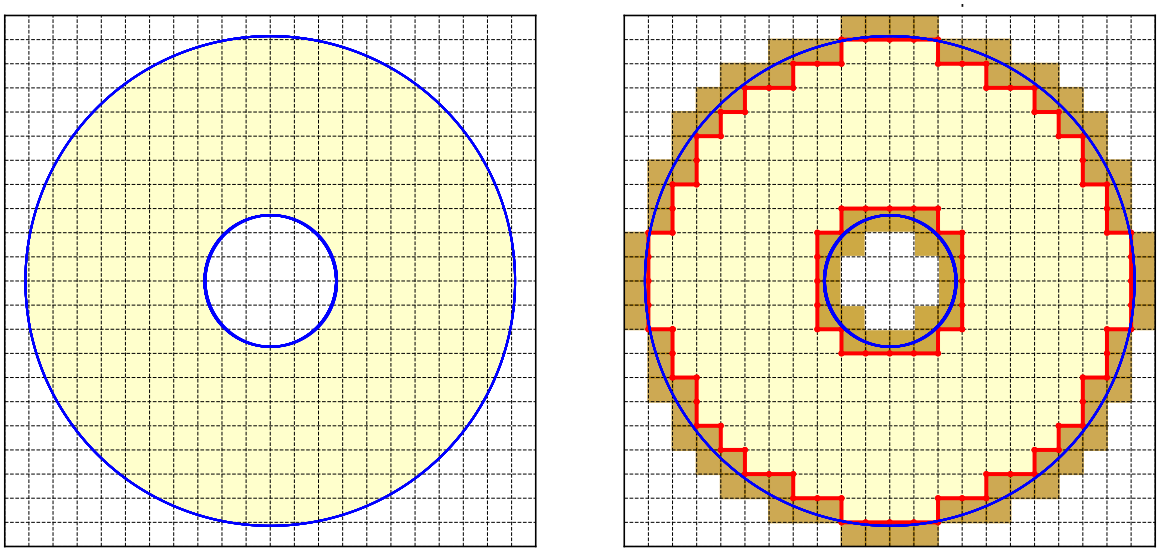
$$\nabla u \cdot \boldsymbol{n} = t_N \quad \text{on } \Gamma_N, \tag{3}$$

where u is the primary variable, u_D its value on the Dirichlet boundary Γ_D, $\boldsymbol{n}$ is the unit outer normal, t_N is the prescribed value of the normal derivative on the Neumann boundary Γ_N, and f a body force.

2.2. The shifted boundary framework: the surrogate domain, the distance field, and Taylor expansions

To start the discussion, we will review the SBM in the context of a standard high-order finite element method. In Section 3 we will then describe the differences in implementation in the IGA context. Consider a Cartesian grid featuring an unfitted discretization, where the boundaries of the computational grid do not match those of the problem domain (as illustrated in Fig. 1a), that is, the boundary of the domain of interest intersects the computational grid at arbitrary locations.

The first step in the implementation of the SBM is the definition of a surrogate domain $\tilde{\Omega}_h$, composed of all the elements that are strictly contained in the domain Ω. With this definition, it is clear that $\tilde{\Omega}_h \subset \Omega$, as shown in Fig. 1b. The boundary of the surrogate domain is denoted $\tilde{\Gamma}_h$, and it is composed of the edges of the background mesh that are close to the true boundary Γ. The governing equations are discretized in $\tilde{\Omega}_h$ rather than Ω, with the challenge of accurately imposing boundary conditions on $\tilde{\Gamma}_h$.



(a) CutFEM-like approach

(b) SBM approach

Figure 1: An example of a domain Ω with a ring-like shape treated with the SBM in a Cartesian grid. Blue and red solid lines denote the true and surrogate boundaries respectively. The computational domain is shaded in light yellow, while brown denotes the intersected cells that are not part of the SBM computation.

The one described so far is the original version of the SBM [37, 38]. However, recent studies in the context of Cartesian grids [46] have demonstrated that the numerical error can be reduced if the requirement that $\tilde{\Omega}_h \subset \Omega$ is lifted. To a certain extent, the construction of $\tilde{\Gamma}_h$ from Γ is a computational geometry problem, and several techniques can be borrowed from this field. For example, the authors of [46] proposed a strategy based on minimizing the area/volume of the gap between $\tilde{\Gamma}_h$ and Γ, in two/three dimensions, respectively. We now define the map

$$\begin{aligned} \boldsymbol{M}_h : \tilde{\Gamma}_h &\to \Gamma, \\ \tilde{\boldsymbol{x}} &\mapsto \boldsymbol{x}, \end{aligned} \tag{4}$$

that associates to a point $\tilde{\boldsymbol{x}} \in \tilde{\Gamma}_h$ its closest-point projection $\boldsymbol{x}$ on the true boundary Γ. $\boldsymbol{M}_h$ is instrumental in defining the distance vector function $\boldsymbol{d}(\tilde{\boldsymbol{x}})$ between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$:

$$\boldsymbol{d}(\tilde{\boldsymbol{x}}) = \boldsymbol{x} - \tilde{\boldsymbol{x}} = [\boldsymbol{M}_h - \boldsymbol{I}](\tilde{\boldsymbol{x}}), \tag{5}$$

where $\boldsymbol{I}$ is the identity operator.

The objective of the SBM is to impose the boundary conditions of the problem without integrating the equations over cut cells. The location where boundary conditions are imposed is shifted in location from Γ to $\tilde{\Gamma}_h$. Also the value of the boundary condition is shifted, that is modified, by means of a Taylor expansion. Note that these two steps ensure optimal convergence rates, while the overall order of convergence cannot be better than first-order, if the Taylor expansion is missing.

In the first version of the SBM [37, 38], only first-order Taylor expansions were considered since a piecewise-linear finite element basis was used. The high-order version of the SBM was presented and analyzed in [48]. In the higher-order SBM, boundary conditions are shifted from Γ to $\tilde{\Gamma}_h$ by performing an mth-order Taylor expansion of the variable of interest at the surrogate boundary. Assuming that u is sufficiently smooth in the strip between $\tilde{\Gamma}_h$ and Γ to allow for a pointwise mth-order Taylor expansion, let $\mathrm{D}^i_{\boldsymbol{d}}$ denote the ith-order directional derivative along $\boldsymbol{d}$, that is

$$\mathrm{D}^i_{\boldsymbol{d}} u = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^n, |\boldsymbol{\alpha}| = i} \frac{i!}{\boldsymbol{\alpha}!} \frac{\partial^i u}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} \boldsymbol{d}^{\boldsymbol{\alpha}}. \tag{6}$$

Then, the mth-order Taylor expansion of u reads

$$u(\boldsymbol{x}) = u(\tilde{\boldsymbol{x}} + \boldsymbol{d}(\tilde{\boldsymbol{x}})) = u(\tilde{\boldsymbol{x}}) + \sum_{i=1}^{m} \frac{\mathrm{D}^i_{\boldsymbol{d}} u(\tilde{\boldsymbol{x}})}{i!} + (\mathrm{R}^m(u, \boldsymbol{d}))(\tilde{\boldsymbol{x}}), \tag{7}$$

where the remainder $\mathrm{R}^m(u, \boldsymbol{d})$ satisfies $|\mathrm{R}^m(u, \boldsymbol{d})| = o(\|\boldsymbol{d}\|^m)$ as $\|\boldsymbol{d}\| \to 0$. Suppose that the Dirichlet condition $u(\boldsymbol{x})_\Gamma = u_D(\boldsymbol{x})$ needs to be applied on the true boundary Γ. Using the map $\boldsymbol{M}_h$, one can extend u_D from Γ to $\tilde{\Gamma}_h$ as $u_D(\tilde{\boldsymbol{x}}) = u_D(\boldsymbol{M}_h(\tilde{\boldsymbol{x}}))$. Introducing the boundary shift operator

$$\mathrm{S}^m_{\boldsymbol{d}} u(\tilde{\boldsymbol{x}}) := u(\tilde{\boldsymbol{x}}) + \sum_{i=1}^{m} \frac{\mathrm{D}^i_{\boldsymbol{d}} u(\tilde{\boldsymbol{x}})}{i!} \tag{8}$$

for every $\tilde{\boldsymbol{x}} \in \tilde{\Gamma}_h$, the Taylor expansion can be used to identify the conditions to be applied on the surrogate boundary $\tilde{\Gamma}_h$ as

$$\mathrm{S}^m_{\boldsymbol{d}} u - u_D + \mathrm{R}^m(u, \boldsymbol{d}) = 0 \quad \text{on } \tilde{\Gamma}_h. \tag{9}$$

Neglecting the remainder $\mathrm{R}^m(u, \boldsymbol{d})$, we arrive at the final expression for the shifted approximation of the boundary condition of order m

$$\mathrm{S}^m_{\boldsymbol{d}} u \approx u_D \quad \text{on } \tilde{\Gamma}_h. \tag{10}$$

In what follows, the SBM boundary condition will always be enforced weakly.

2.3. Classic and Penalty-Free formulations

Let Ω be a computational domain discretized into a Cartesian grid of rectangular elements and $\tilde{\Omega}_h$ be the surrogate domain enclosed by the surrogate boundary.

Throughout the paper, we will denote by $L^2(\Omega)$ the space of square integrable functions on Ω. We will use the Sobolev spaces $H^m(\Omega) = W^{m,2}(\Omega)$ of index of regularity $m \geq 0$ and index of summability 2, equipped with the (scaled) norm

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

where $\boldsymbol{D}^i$ is the ith-order spatial derivative operator and $l(A) = \text{meas}_{n_d}(A)^{1/n_d}$ is a characteristic length of the domain A. Note that $H^0(\Omega) = L^2(\Omega)$. As usual, we use a simplified notation for norms and semi-norms, i.e., we set $\| v \|_{m,\Omega} = \| v \|_{H^m(\Omega)}$ and $| v |_{i,\Omega} = \| \boldsymbol{D}^i v \|_{0,\Omega} = \| \boldsymbol{D}^i v \|_{L^2(\Omega)}$.

We define the finite element space $V_h^m(\tilde{\Omega}_h)$ as:

$$V_h^m(\tilde{\Omega}_h) = \left\{ v_h \in \mathscr{C}^0(\tilde{\Omega}_h) : v_h|_K \in \mathscr{P}^m(K),\ \forall K \in \mathscr{T}_h \right\}, \tag{12}$$

where $\mathscr{T}_h$ denotes the triangulation of Ω into rectangular elements K, $\mathscr{C}^0(\tilde{\Omega}_h)$ represents the space of continuous functions over $\tilde{\Omega}_h$ and $\mathscr{P}^m(K)$ is a space of functions over the element K, obtained as the product space of polynomials of degree m along each spatial coordinate. Therefore, $v_h \in V_h^m$ is a scalar-valued function that is continuous across element boundaries and polynomial of degree m within each element K. The SBM variational formulation relies on Nitsche's method, and reads:

Find $u_h \in V_h^m(\tilde{\Omega}_h)$ such that, for all $v_h \in V_h^m(\tilde{\Omega}_h)$

$$a_h^m(u_h, v_h) = l_h(v_h), \tag{13a}$$

where

$$a_h^m(u_h, v_h) = (\nabla u_h, \nabla v_h)_{\tilde{\Omega}_h} - \langle \nabla u_h \cdot \tilde{\boldsymbol{n}}, v_h \rangle_{\tilde{\Gamma}_h} - \langle S_{\boldsymbol{d}}^m u_h, \nabla v_h \cdot \tilde{\boldsymbol{n}} \rangle_{\tilde{\Gamma}_h} + \frac{\alpha}{h} \langle v_h, S_{\boldsymbol{d}}^m u_h \rangle_{\tilde{\Gamma}_h}, \tag{13b}$$

$$l_h(v_h) = (f, v_h)_{\tilde{\Omega}_h} - \langle u_D, \nabla v_h \cdot \tilde{\boldsymbol{n}} \rangle_{\tilde{\Gamma}_h} + \frac{\alpha}{h} \langle v_h, u_D \rangle_{\tilde{\Gamma}_h}. \tag{13c}$$

Here $\tilde{\boldsymbol{n}}$ denotes the unit outward-pointing normal to the surrogate boundary $\tilde{\Gamma}_h$, distinct from the outward-pointing normal $\boldsymbol{n}$ of Γ. Furthermore, α represents a penalty parameter, and h is the characteristic length of the elements. As mentioned before, $S_{\boldsymbol{d}}^m u_h$ is the Taylor expansion of order m of u_h between any point $\tilde{\boldsymbol{x}} \in \tilde{\Gamma}_h$ and its closest point $\boldsymbol{x} \in \Gamma$.

Nitsche's weak formulation is numerically stable for a sufficiently large value of the parameter α. The selection of a stable value of α involves a detailed estimation of the constants in the trace inequalities associated with the specific grid utilized. This problem in turn can be related to an eigenvalue problem and it is in general tedious to implement. In [45] the authors proposed a penalty-free SBM for the Poisson and linear elasticity equations. In the case of the Poisson problem, the penalty-free weak formulation reads as:

Find $u_h \in V_h^m(\tilde{\Omega}_h)$ such that, for all $v_h \in V_h^m(\tilde{\Omega}_h)$

$$a_h^m(u_h, v_h) = l_h(v_h), \tag{14a}$$

where

$$a_h^m(u_h, v_h) = (\nabla u_h, \nabla v_h)_{\tilde{\Omega}_h} - \langle \nabla u_h \cdot \tilde{\boldsymbol{n}}, v_h \rangle_{\tilde{\Gamma}_h} + \langle S_{\boldsymbol{d}}^m u_h, \nabla v_h \cdot \tilde{\boldsymbol{n}} \rangle_{\tilde{\Gamma}_h}, \tag{14b}$$

$$l_h(v_h) = (f, v_h)_{\tilde{\Omega}_h} + \langle u_D, \nabla v_h \cdot \tilde{\boldsymbol{n}} \rangle_{\tilde{\Gamma}_h}. \tag{14c}$$

The above variational form differs from (13) because the penalty is set to zero ($\alpha = 0$) and the sign of the term $\langle S_{\boldsymbol{d}}^m u_h - u_D, \nabla v_h \cdot \tilde{\boldsymbol{n}} \rangle_{\tilde{\Gamma}_h}$ is changed. In Section 4.2, we will compare these two variational approaches. In any case, owing to its simplicity and accuracy, we focus primarily on the penalty-free weak formulation.

2.4. Neumann boundary conditions

The use of the SBM for imposing Neumann boundary conditions was already considered in [37], and is still a topic of intense investigation, for reasons that will appear clear momentarily. Following the general approach in [37], the objective is to use the term coming from the integration by parts, namely $\langle \nabla u_h \cdot \tilde{\boldsymbol{n}}, v_h \rangle_{\tilde{\Gamma}_h}$, in order to substitute the known flux $t_N = \nabla u_h \cdot \boldsymbol{n}$ at Γ_N. To do so, we can introduce the following equality

$$\tilde{\boldsymbol{n}} = (\tilde{\boldsymbol{n}} \cdot \boldsymbol{n})\boldsymbol{n} + (\tilde{\boldsymbol{n}} \cdot \boldsymbol{\tau})\boldsymbol{\tau}, \tag{15}$$

therefore, we can write

$$\langle \nabla u_h \cdot \tilde{\boldsymbol{n}}, v_h \rangle_{\tilde{\Gamma}_h} = \langle (\nabla u_h \cdot \boldsymbol{n})\tilde{\boldsymbol{n}} \cdot \boldsymbol{n}, v_h \rangle_{\tilde{\Gamma}_h} + \langle (\nabla u_h \cdot \boldsymbol{\tau})\tilde{\boldsymbol{n}} \cdot \boldsymbol{\tau}, v_h \rangle_{\tilde{\Gamma}_h}. \tag{16}$$

Given the definition of the Taylor expansion of each term of the gradient of u

$$\mathrm{S}_d^m \nabla u_h(\tilde{\boldsymbol{x}}) := \nabla u_h(\tilde{\boldsymbol{x}}) + \sum_{i=1}^{m} \frac{\mathrm{D}_d^i \nabla u_h(\tilde{\boldsymbol{x}})}{i!}, \tag{17}$$

and analogously to the shifted boundary approximation for Dirichlet boundary conditions, we can write the shifted boundary approximation of the flux t_N

$$\mathrm{S}_d^m \nabla u_h \cdot \boldsymbol{n} \approx t_N \quad \text{on } \tilde{\Gamma}_h. \tag{18}$$

Combining Eq. (17) and Eq. (18)

$$\nabla u_h(\tilde{\boldsymbol{x}}) \cdot \boldsymbol{n} \approx t_N - \Big[\sum_{i=1}^{m} \frac{\mathrm{D}_d^i \nabla u_h(\tilde{\boldsymbol{x}})}{i!} \Big] \cdot \boldsymbol{n} \tag{19}$$

and substituting $\nabla u_h(\tilde{\boldsymbol{x}}) \cdot \boldsymbol{n}$ in Eq. (16) one obtains

$$\langle \nabla u_h \cdot \tilde{\boldsymbol{n}}, v_h \rangle_{\tilde{\Gamma}_h} = \langle (t_N - \Big[\sum_{i=1}^{m} \frac{\mathrm{D}_d^i \nabla u_h(\tilde{\boldsymbol{x}})}{i!} \Big] \cdot \boldsymbol{n})\tilde{\boldsymbol{n}} \cdot \boldsymbol{n}, v_h \rangle_{\tilde{\Gamma}_h} + \langle (\nabla u_h \cdot \boldsymbol{\tau})\tilde{\boldsymbol{n}} \cdot \boldsymbol{\tau}, v_h \rangle_{\tilde{\Gamma}_h}. \tag{20}$$

Finally, adding and subtracting $\langle (\nabla u_h \cdot \boldsymbol{n})\tilde{\boldsymbol{n}} \cdot \boldsymbol{n}, v_h \rangle_{\tilde{\Gamma}_h}$ leads to a simpler implementation that eliminates the computation of the tangential unit vector $\boldsymbol{\tau}$. Hence,

$$\begin{aligned}
\langle \nabla u_h \cdot \tilde{\boldsymbol{n}}, v_h \rangle_{\tilde{\Gamma}_h} &= \langle (t_N - \Big[\sum_{i=1}^{m} \frac{\mathrm{D}_d^i \nabla u_h(\tilde{\boldsymbol{x}})}{i!} \Big] \cdot \boldsymbol{n})\tilde{\boldsymbol{n}} \cdot \boldsymbol{n}, v_h \rangle_{\tilde{\Gamma}_h} + \langle (\nabla u_h \cdot \boldsymbol{\tau})\tilde{\boldsymbol{n}} \cdot \boldsymbol{\tau}, v_h \rangle_{\tilde{\Gamma}_h} + \langle (\nabla u_h \cdot \boldsymbol{n})\tilde{\boldsymbol{n}} \cdot \boldsymbol{n}, v_h \rangle_{\tilde{\Gamma}_h} + \\
&\quad - \langle (\nabla u_h \cdot \boldsymbol{n})\tilde{\boldsymbol{n}} \cdot \boldsymbol{n}, v_h \rangle_{\tilde{\Gamma}_h} \\
&= \langle (t_N - \Big[\sum_{i=1}^{m} \frac{\mathrm{D}_d^i \nabla u_h(\tilde{\boldsymbol{x}})}{i!} \Big] \cdot \boldsymbol{n})\tilde{\boldsymbol{n}} \cdot \boldsymbol{n}, v_h \rangle_{\tilde{\Gamma}_h} + \langle \nabla u_h \cdot \tilde{\boldsymbol{n}}, v_h \rangle_{\tilde{\Gamma}_h} - \langle (\nabla u_h \cdot \boldsymbol{n})\tilde{\boldsymbol{n}} \cdot \boldsymbol{n}, v_h \rangle_{\tilde{\Gamma}_h} \\
&= \langle (t_N - \nabla u_h \cdot \boldsymbol{n} - \Big[\sum_{i=1}^{m} \frac{\mathrm{D}_d^i \nabla u_h(\tilde{\boldsymbol{x}})}{i!} \Big] \cdot \boldsymbol{n})\tilde{\boldsymbol{n}} \cdot \boldsymbol{n}, v_h \rangle_{\tilde{\Gamma}_h} + \langle \nabla u_h \cdot \tilde{\boldsymbol{n}}, v_h \rangle_{\tilde{\Gamma}_h} \\
&= \langle (t_N - \mathrm{S}_d^m \nabla u_h \cdot \boldsymbol{n})\tilde{\boldsymbol{n}} \cdot \boldsymbol{n}, v_h \rangle_{\tilde{\Gamma}_h} + \langle \nabla u_h \cdot \tilde{\boldsymbol{n}}, v_h \rangle_{\tilde{\Gamma}_h}.
\end{aligned} \tag{21}$$

The result is a weak imposition of the Neumann constraint that is naturally penalized by the factor $\tilde{\boldsymbol{n}} \cdot \boldsymbol{n}$, which relates to the ratio of areas between a surrogate boundary portion and its projection on the true boundary. The corresponding linear and bilinear forms read as

$$a_h^m(u_h, v_h) = (\nabla u_h, \nabla v_h)_{\tilde{\Omega}_h} - \langle \nabla u_h \cdot \tilde{\boldsymbol{n}}, v_h \rangle_{\tilde{\Gamma}_h} + \langle (\mathrm{S}_d^m \nabla u_h \cdot \boldsymbol{n})\tilde{\boldsymbol{n}} \cdot \boldsymbol{n}, v_h \rangle_{\tilde{\Gamma}_h}, \tag{22a}$$

$$l_h(v_h) = (f, v_h)_{\tilde{\Omega}_h} + \langle t_N \tilde{\boldsymbol{n}} \cdot \boldsymbol{n}, v_h \rangle_{\tilde{\Gamma}_h}. \tag{22b}$$

Remark 1. This application of the SBM approach to Neumann conditions leads to the loss of one order of convergence in the L^2-norm of the error, with respect to the optimal rates: the L^2-error is $\mathcal{O}(h^p)$ instead of $\mathcal{O}(h^{p+1})$. This occurs because the term of order p in the Taylor expansion Eq. (18) vanishes. It has been shown in [40, 47] how optimal convergence rates can be recovered by using a mixed approach in which the gradients of the solution are used as unknown variables. This approach was successfully tested in the one-dimensional case for B-Spline shape functions, but it is not pursued here, for the sake of simplicity: we instead exploit the trade-off of a simpler implementation for a loss of one order of accuracy.

3. Numerical solution of the Poisson problem with the SBM and IGA

3.1. Optimal surrogate boundary in parameter space

Recently, Yang *et al.* [46] introduced the concept of an optimal surrogate boundary in SBM simulations with Cartesian grids. The authors proved that the SBM remains stable even when the surrogate domain $\tilde{\Omega}_h$ is not fully contained within the true domain Ω. In Fig. 1b, we have illustrated the original approach of the SBM in scenarios involving the management of both external and internal embedded boundaries. However, it is intuitive that a better approximation for the surrogate boundary can be selected, for instance, by also using element edges that are not fully contained within the true domain Ω. In [46] the authors introduced the parameter λ, which quantifies how much of the total volume of a cut element lies outside the true domain Ω. In other words, λ is a bound on the ratio between the portion of elemental volume lying outside of Ω and the total elemental volume. A surrogate domain $\tilde{\Omega}_h$ with $0 < \lambda < 1$ includes cut elements for which the ratio between the elemental volume external to Ω and the total elemental volume is less than λ. Hence: if $\lambda = 0$, we have the traditional construction of the SBM surrogate domain, which excludes all cut elements; if $\lambda = 1$, all the cut elements are included in the surrogate domain; and if $\lambda = 0.5$, cut elements that lie in the true domain by at least 50% in volume are included in the surrogate domain. It was shown in [46] that the λ which minimizes the distance function for every point $\tilde{\boldsymbol{x}}$ on the surrogate boundary $\tilde{\Gamma}_h$ is $\lambda = 0.5$. This value of λ results in what the authors referred to as the optimal surrogate boundary, since it minimizes the numerical error in SBM computations.

In the context of IGA, the concept of optimal surrogate boundary remains unchanged. The knot spans are treated as quadrilateral elements of a classical FEM approach.

Remark 2. This work assumes an ordered clockwise discretization of the true boundary Γ, but the method can be extended to a true boundary described by a NURBS curve. The Taylor expansion is always computed to a point in the exact geometry of Γ. In fact, having at least one exact geometry point for each knot span is sufficient for our SBM method.

To determine whether a cut element, i.e. knot span, should be considered as part of $\tilde{\Omega}_h$, we initially select n^2 equally spaced points within the cut knot span. Experimentally, we found that $n = 5$ is sufficient to accurately determine the optimal surrogate boundary. For each of those points, utilizing the two closest points belonging to the discretized true boundary, we compute the signed area using a cross-product. Consider $\boldsymbol{P_1}$ in Fig. 2, which is one of the n^2 equally spaced points of a cut knot span. We would like to check if it lies inside the true domain Ω or not. First, a search algorithm is used to find the closest segment belonging to the true boundary Γ. Note that the true boundary is simply a polygon and it is composed of segments (red line in Fig. 2). Then, defining the ordered points as $\boldsymbol{P_2}$ and $\boldsymbol{P_3}$, we specify the following vectors

$$\begin{aligned} \mathbf{v_1} &= \boldsymbol{P_2} - \boldsymbol{P_1} \\ \mathbf{v_2} &= \boldsymbol{P_3} - \boldsymbol{P_1}. \end{aligned} \tag{23}$$

Assuming the true boundary points to be ordered clockwise, we have that $\boldsymbol{P_1}$ is contained in Ω only if the z-component of the vector product is positive, i.e

$$\Big(\mathbf{v_1} \times \mathbf{v_2}\Big)_z > 0. \tag{24}$$

Once we have counted the number of internal points in the cut knot span under consideration, we consider it as contained in $\tilde{\Omega}_h$ if the ratio between internal and total points is larger than λ. At this point, we can repeat the same

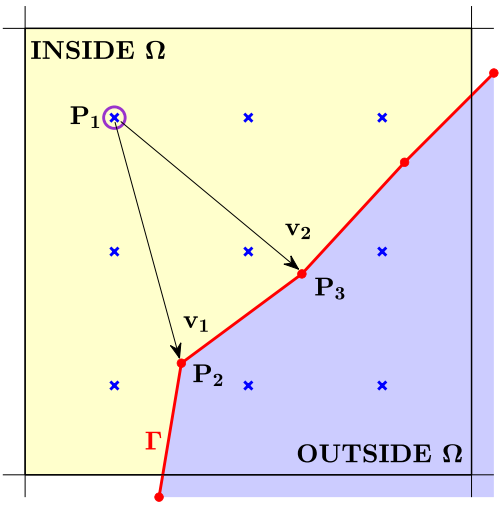


Figure 2: Surrogate Optimal Boundary Selection. Graphical representation of the vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ which determines, using the cross-product, whether $\boldsymbol{P_1}$ is contained in Ω.

procedure for all the cut knot spans. After determining which are included in $\tilde{\Omega}_h$, we can identify the surrogate boundary $\tilde{\Gamma}_h$, which only cuts the parameter space along the knot values.
Consider the two-dimensional Stanford Bunny used as an external boundary. Fig. 3 illustrates examples for $\lambda = 0,\ 1/2,$ and 1. Four colors are used to distinguish the knot spans: white indicates knot spans completely outside the true domain Ω; yellow denotes knot spans fully inside Ω; light blue represents the cut knot spans that are predominantly inside Ω, and brown marks the cut knot spans that are mainly outside Ω. Given those definitions, one can reconstruct the surrogate boundaries for the three values of λ. In particular, the surrogate boundary that encloses the yellow knot spans is identical to the first version of the SBM proposed in [37, 38]. The optimal surrogate boundary ($\lambda = 0.5$), which encloses the surrogate domain delineated by the sum of the yellow and light blue knot spans, locally minimizes the distance function $\boldsymbol{d}$. Indeed, as demonstrated in [46] and as will be further illustrated in Section 4.1, the optimal surrogate boundary enables the Taylor expansion of the Dirichlet boundary conditions to use the shortest distances.

3.2. Implementation of the SBM in IGA

The implementation of the SBM weak formulations of the Poisson problem in the IGA context comes naturally starting from classical FEM. To do so, let us first summarise the main features that differentiate the IBRA framework from classical FEM approaches:

1. no mesh has to be constructed;
2. the basis functions are different but they still have local support, i.e. $(p+1)(q+1)$ knot spans;
3. it is possible to increase the degrees p and q of the basis functions in the x and y direction;
4. even when degree elevation is applied, it is possible to have up to $p-1$ continuity between knot spans.
5. IBRA requires a specific approach to integrate the trimmed knot spans.

All the IGA approaches used in this work (i.e., the SBM and the classical IBRA methods) have been implemented in the open-source Kratos Multiphysics framework [49, 50], which can read any two-dimensional trimmed geometry, compute the required integration points, and assemble the stiffness matrix. IBRA takes $(p+1)(q+1)$ integration points for every not-trimmed knot span. Conversely, for trimmed knot spans, a triangulation approximates the area, and then Gaussian quadrature formulae are applied on each triangle to integrate over the cut geometry.

The IGA mathematical foundations heavily rely on the Cox-de Boor recursion formula, which is essential for constructing one-dimensional B-Spline basis functions of degree p. This recursion formula, shown for the first time in [51, 52], is formalized as follows:

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}, \tag{25}$$

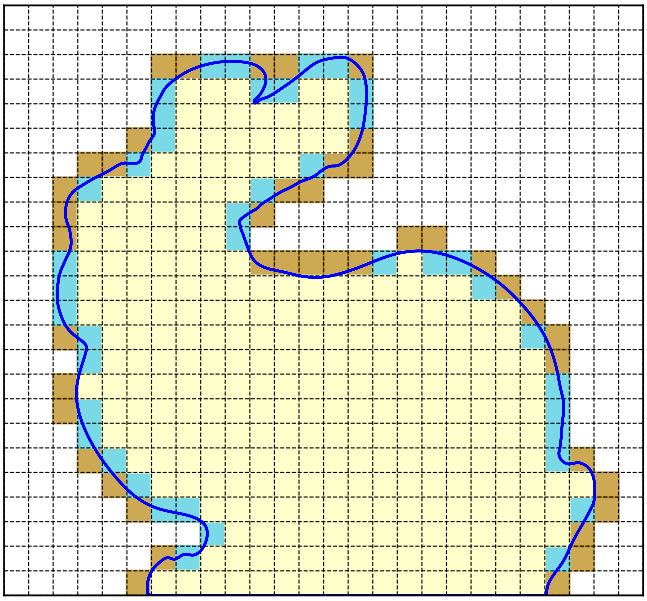


Figure 3: Two-dimensional Stanford Bunny used as an embedded external boundary applying the SBM and varying λ. The blue solid line denotes the true boundary; the cells colored in light yellow are the surrogate domain when $\lambda = 0$, the light blue are also considered as active when $\lambda = 0.5$, and the brown cells are active if $\lambda = 1$.

for the zeroth degree ($p = 0$). For higher degrees ($p > 0$), it is given by

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \tag{26}$$

The B-Spline basis functions $N_{i,p}$ are defined by their polynomial degree p and a sorted sequence of parametric coordinates, referred to as the knot vector

$$\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}, \tag{27}$$

where ξ_i denotes the ith knot value in the vector, and n represents the number of basis functions. The basis functions $N_{i,p}$ are $\mathscr{C}^{p-\bar{k}}$ continuous, where $\bar{k}$ is the knot multiplicity. Throughout this work, we employ $\bar{k} = 1$ for all inner knots and $\bar{k} = p + 1$ at the ends of the knot interval, resulting in an open knot vector.

A bivariate parameterization for surfaces is obtained from the tensor product of the shape functions in two directions. The mapping from a point $\xi = (\xi, \eta)$ in a parametric space defined by the knot vectors $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$ and $H = \{\eta_1, \eta_2, \ldots, \eta_{m+q+1}\}$ to its corresponding physical point is computed as

$$\bar{S}(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(\xi) N_{j,q}(\eta) P_{i,j}, \tag{28}$$

with control points $P_{i,j} \in \mathbb{R}^2$. n and m are the respective numbers of basis functions in the ξ and η directions. Before the embedding of any boundary Γ, the domain represented as a B-Spline surface is limited to a rectangular shape defined by uniform knot vectors. Any boundary that does not follow the external rectangular shape in our proposed SBM-IGA formulation is considered embedded and is managed by adopting the SBM (refer to Fig. 4). Utilizing this simplified approach, the creation of any B-Spline framework can be achieved through the implementation of algorithms focused on increasing the order and refining the knots, as detailed in [53]. Only a few basic inputs are required for this method: the size of the knot intervals, the polynomial degrees p and q, and the dimensions of the physical space. Although NURBS provide a more flexible parameterization than B-Splines, they offer no special benefits when used to create a rectangular surface. Rather, they add complexity to the modeling process with no apparent advantages.

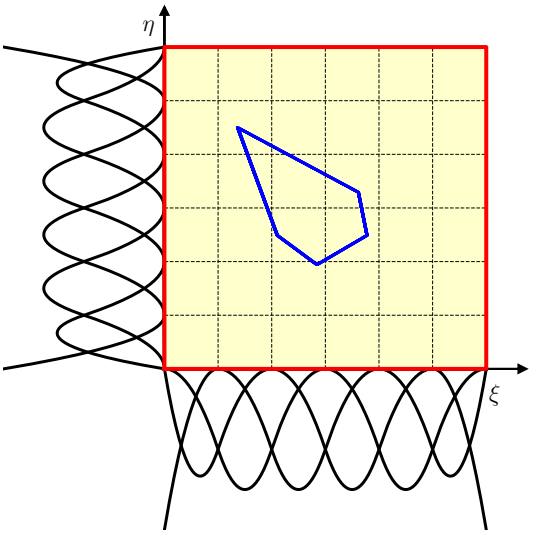


Figure 4: An illustration of a square background domain with open knot vectors in both directions that is represented as a B-Spline surface. An embedded boundary is highlighted in blue, and in red the body-fitted boundary of the B-Spline surface.

In comparison to the FEM discrete space defined in Eq. (12), the discrete space for B-Splines basis functions is given by

$$V_h^{p,q}(\tilde{\Omega}_h) = \left\{ v_h \in \mathscr{C}^{p-1,q-1}(\tilde{\Omega}_h) \mid v_h(\xi,\eta) = \sum_i v_i N_i^{p,q}(\xi,\eta),\ (\xi,\eta) \in [0,1]^2 \right\}, \tag{29}$$

where $N_i^{p,q}(\xi,\eta) = N_i^p(\xi)N_i^q(\eta)$ are the tensor product B-Spline basis functions of degrees p and q in the ξ and η directions, respectively, and $\mathscr{C}^{p-1,q-1}(\tilde{\Omega}_h)$ is the space of functions defined over $\tilde{\Omega}_h$ with order of continuity $p-1$ in x and $q-1$ in y. The coefficients v_i represent the degrees of freedom, each associated with a control point in a two-dimensional space.

The implementation of the SBM-IGA framework is based on the IBRA description [30, 31]. The core idea of IBRA is to utilize the B-Rep method for representing shapes using boundaries, which consist of geometry (shape) and topology (faces, edges, vertices). A face is bounded by edges on the surface. Similarly, an edge is defined by a curve determined by points named vertices. As a result, a surface is enclosed by trimming curves, enabling the imposition of boundary conditions at arbitrary positions within the domain. A trimmed surface is then defined by a surface $\bar{S}(\xi,\eta)$ and a set of N trimming curves $\bar{C}_k(\xi,\eta)$ with $k = 1, ..., N$ lying within the parameter space of the surface. The curves form the inner loops (clockwise) and outer loops (counter-clockwise) enclosing the surfaces. Element functions over trimmed surfaces are evaluated using integration points inside the trimmed domain. The key point of IBRA is to differentiate span-wise between the non-trimmed and trimmed spans.

The treatment of non-trimmed knot spans of the SBM is the same as the one of IBRA, the key difference is the absence of trimmed knot spans. Having said this, the main additional feature is the computation of the boundary term $\langle S_{\boldsymbol{d}}^m u_h, \nabla v_h \cdot \tilde{\boldsymbol{n}} \rangle_{\tilde{\Gamma}_h}$ in Eq. (14b), which allows the imposition of the shifted boundary conditions. Given the definition of the mth-order Taylor expansion in Eq. (8), the extended Taylor formula to compute the scalar function u_h in $(x,y) \in \Gamma$, given the value of u_h in $(\tilde{x},\tilde{y}) \in \tilde{\Gamma}_h$ and its derivatives, is

$$S_{\boldsymbol{d}}^{\infty} u_h(x,y) = u_h(\tilde{x},\tilde{y}) + \sum_{n=1}^{\infty} \sum_{k=0}^{n} \frac{1}{k!(n-k)!} \frac{\partial^n u_h}{\partial x^k \partial y^{n-k}}\bigg|_{(\tilde{x},\tilde{y})} (x-\tilde{x})^k (y-\tilde{y})^{n-k}. \tag{30}$$

The implementation of this term in the higher-order SBM-IGA applications directly depends on the degree of the B-Spline basis functions selected to describe the B-Spline surface under consideration. For instance, in the case of quadratic basis functions all the derivatives with $n > 2$ are zeros. Hence, the Taylor expansion $S_{\boldsymbol{d}}^m u_h$ can be truncated

at $m = 2$ leading to a Taylor expansion of the form

$$\begin{aligned} S^2_{\boldsymbol{d}} u_h(x,y) &= u_h(\tilde{x},\tilde{y}) + \sum_{n=1}^{2}\sum_{k=0}^{n} \frac{1}{k!(n-k)!} \frac{\partial^n u_h}{\partial x^k \partial y^{n-k}}\bigg|_{(\tilde{x},\tilde{y})} (x-\tilde{x})^k (y-\tilde{y})^{n-k} \\ &= u_h(\tilde{x},\tilde{y}) + \frac{\partial u_h}{\partial y}\bigg|_{(\tilde{x},\tilde{y})} (y-\tilde{y}) + \frac{\partial u_h}{\partial x}\bigg|_{(\tilde{x},\tilde{y})} (x-\tilde{x}) + \\ &\quad + \frac{1}{2}\frac{\partial^2 u_h}{\partial y^2}\bigg|_{(\tilde{x},\tilde{y})} (y-\tilde{y})^2 + \frac{\partial^2 u_h}{\partial y \partial x}\bigg|_{(\tilde{x},\tilde{y})} (x-\tilde{x})(y-\tilde{y}) + \frac{1}{2}\frac{\partial^2 u_h}{\partial x^2}\bigg|_{(\tilde{x},\tilde{y})} (x-\tilde{x})^2 \\ &= u_h(\tilde{x},\tilde{y}) + \nabla u_h(\tilde{x},\tilde{y}) \cdot \boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^{\mathrm{T}} \boldsymbol{H}(\tilde{x},\tilde{y}) \boldsymbol{d}, \end{aligned} \tag{31}$$

where by definition $\boldsymbol{d} = \boldsymbol{x} - \tilde{\boldsymbol{x}}$ and $\boldsymbol{H}$ is the Hessian matrix which contains the second-order partial derivatives of u_h. Therefore, depending on which B-Spline basis functions degree is chosen, the Taylor expansion can recover the basis function value at any point with an error proportional to $\|\boldsymbol{d}\|^{p+1}$, and so optimal converge is expected.

3.3. Evaluation of the solution and quantities of interest on the true boundary

In engineering applications, it is often necessary to compute the solution on the true boundary. For instance, in an aerospace application, calculating the pressure drag coefficient requires integrating the pressure solution p along the true boundary Γ, which corresponds to the immersed shape.

The SBM can be utilized to obtain the solution on the true boundary by reusing the Taylor expansion at the end of the simulation. Indeed, for every arbitrary point on the true boundary, we can look for the closest point on the surrogate boundary. This point will be at a distance $\boldsymbol{d}$, whose magnitude will be less than h for both two-dimensional and three-dimensional applications. This allows us to compute the solution at the arbitrary point using the m-th order Taylor expansion in Eq. (7) between the integration point $\boldsymbol{x}$ and the corresponding closest point on $\tilde{\Gamma}_h$

$$u(\boldsymbol{x}) = u(\tilde{\boldsymbol{x}}) + \sum_{i=1}^{m} \frac{\mathrm{D}^i_{\boldsymbol{d}} u(\tilde{\boldsymbol{x}})}{i!} + o\left(\|\boldsymbol{d}\|^m\right). \tag{32}$$

Section 4.5 will demonstrate that employing this SBM post-processing enables the recovery of the solution over the true boundary with optimal convergence. Similar considerations hold for quantities of interest derived from the solution.

4. Results

In this chapter, we test the proposed numerical method in various scenarios. We focus almost exclusively on the penalty-free formulation unless stated otherwise (Section 4.2). Error convergence is assessed by means of the following manufactured solution:

$$u(x,y) = \sin(x)\sinh(y). \tag{33}$$

The boundary conditions are of Dirichlet type unless explicitly stated otherwise as Neumann type (Section 4.4). Moreover, the boundary conditions are considered body-fitted whenever a portion of the boundary coincides with an external knot value. On the other hand, the SBM is used for imposing the conditions along the trimmed knot values. The Taylor expansions consider all the possible derivatives that the basis functions can provide. For instance, if $p = 2$, the SBM uses only first and second derivatives for computing the $\mathrm{S}^m_{\boldsymbol{d}} u_h$ term defined in Eq. (8).

Considering the notation stated in Section 2.3, the error is measured using the L^2- and H^1-norm of the difference between the analytical and obtained solutions

$$\begin{aligned} \|u - u_h\|_{0,\tilde{\Omega}_h} &= \sqrt{\int_{\tilde{\Omega}_h} (u-u_h)^2 d\tilde{\Omega}_h}, \\ \|u - u_h\|_{1,\tilde{\Omega}_h} &= \|u - u_h\|_{0,\tilde{\Omega}_h} + |u - u_h|_{1,\tilde{\Omega}_h}, \end{aligned} \tag{34}$$

where the integral is computed applying the Gauss integration rule that guarantees precision of order $p+1$. In particular in all the following convergence studies the normalized error norm will be considered. Note that, as it is customary in the SBM convergence tests, the error is computed in the surrogate domain $\tilde{\Omega}_h$ by evaluating the analytical solution at the integration points.

4.1. Selection of the surrogate boundary

Many computational problems of interest involve domains with internal holes. For instance, studying the fluid flow surrounding an immersed object may be of interest in computational fluid dynamics. This can be achieved with the SBM by creating a hole in the background computational domain and adding additional boundary conditions along an internal surrogate boundary. Consider Fig. 5 where two types (curvilinear- and diamond-shaped) embedded holes are considered. The true domain Ω is, therefore, the square $[0,0]\times[2,2]$ minus the hole, while the surrogate domain $\tilde{\Omega}_h$ is the square minus the inactive knot spans (i.e., the white knot spans in the figure). Two different surrogate boundaries are considered, as illustrated in Fig. 5c: the classical definition of the surrogate boundary ($\lambda=0$), shown in red, and the optimal definition of the surrogate boundary ($\lambda=0.5$), shown in green. Fig. 6 shows the convergence study of the curvilinear boundary shape and the diamond boundary shape for $\lambda=0$ and $\lambda=1/2$. We always consider uniform and open knot vectors such that the knot size of our discretization is

$$h=[0.25,0.125,0.0625,0.03125,0.015625].$$

The comparison is done utilizing linear, quadratic, and cubic basis functions. The convergence of the L^2-norm is always optimal, i.e. $p+1$. Furthermore, the optimal surrogate boundary ($\lambda=0.5$) yields more accurate results, especially for large p. Therefore, the best choice is $\lambda=1/2$.

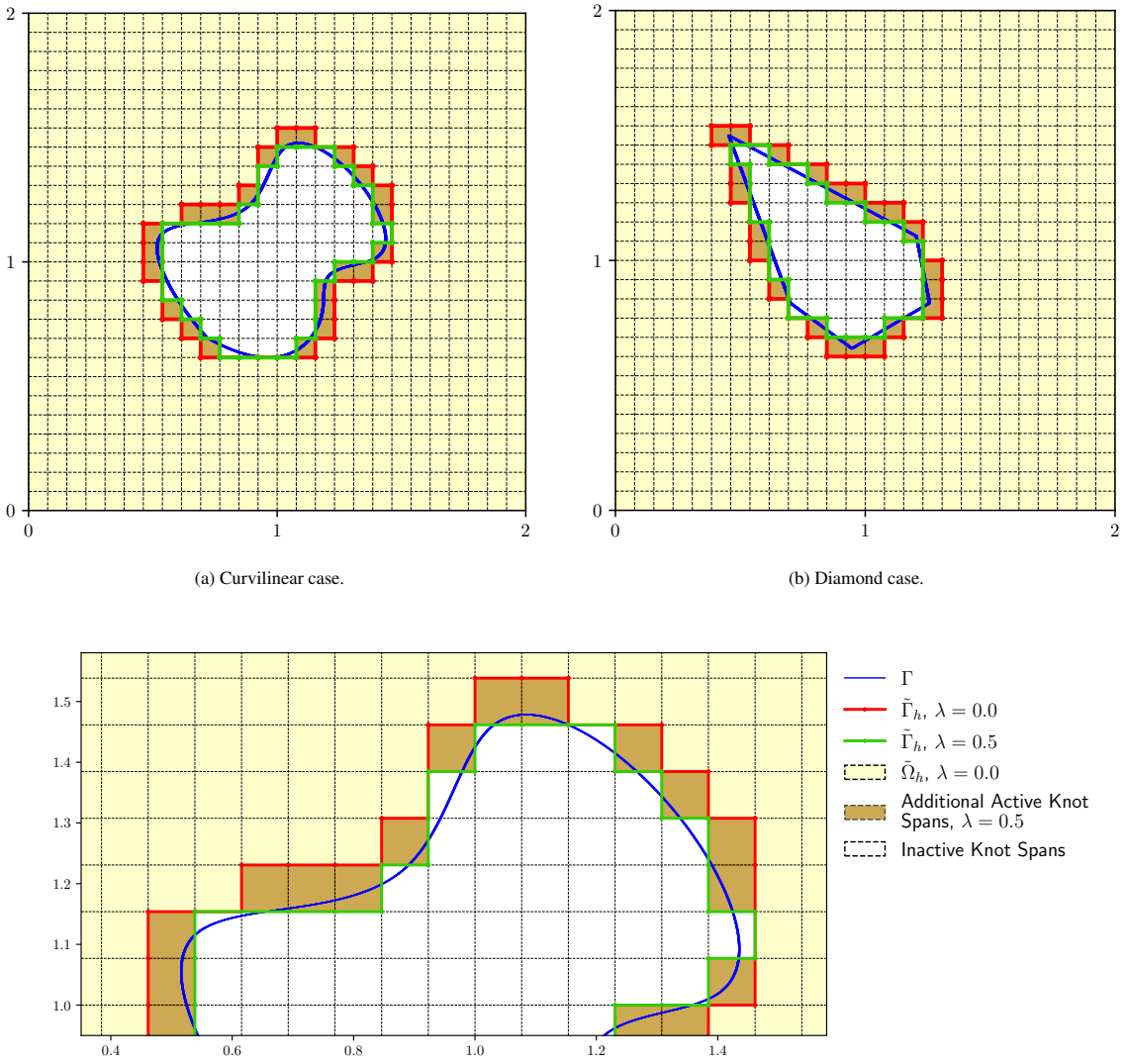
4.2. Classic vs Penalty-Free SBM formulations

An important exercise is to compare the standard and penalty-free versions of the Nitsche formulation for the SBM, in terms of accuracy and robustness. This section attempts to demonstrate the two formulations' capabilities and limitations when dealing with complex shapes. To illustrate this comparison, we focus on the diamond shape example utilizing the optimal surrogate boundary ($\lambda=0.5$). In Fig. 7a we report the contour of the obtained solution. Fig. 7b presents a convergence study comparison between the two approaches. The results show that the standard Nitsche formulation yields lower L^2-norm error values for linear and quadratic basis functions, while the penalty-free Nitsche approach has higher accuracy and less sensitivity to round-off when $p>2$. Note also that the calibration of the penalty term α of Eq. (13b) is a very delicate task, especially for high-order basis functions. In this particular case, we found that $\alpha=10p^2$ is the value that guarantees the lowest errors overall in terms of L^2-norm of the error.

4.3. SBM vs pseudo body-fitted approach

At this point, one may wonder about the disadvantages of using the SBM over a traditional body-fitted approach. Although the convergence rates are optimal, it is useful to compare the two approaches in terms of the actual L^2-norm of the error. To achieve this goal we define a pseudo body-fitted simulation by slightly changing the problem. We consider a body-fitted simulation in which the exact manufactured solution is imposed as a boundary condition on $\tilde{\Gamma}_h$ rather than Γ. In this way, we can compare the SBM approach (for which the true boundary remains Γ) against a body-fitted IGA computation (for which the true boundary is $\tilde{\Gamma}_h$).

To show the robustness of the proposed algorithm we consider a more complex immersed object whose shape is the two-dimensional Stanford Bunny. The convergence study comparison and the problem definition are shown in Fig. 8. Note that the pseudo body-fitted counterpart shares the same geometry as the SBM case, with the exception that the exact value is weakly imposed instead of applying any Taylor expansion. The convergence study shows barely any differences between the SBM and the pseudo body-fitted solutions in terms of convergence and magnitude of the L^2-norm of the error.



(a) Curvilinear case.

(b) Diamond case.

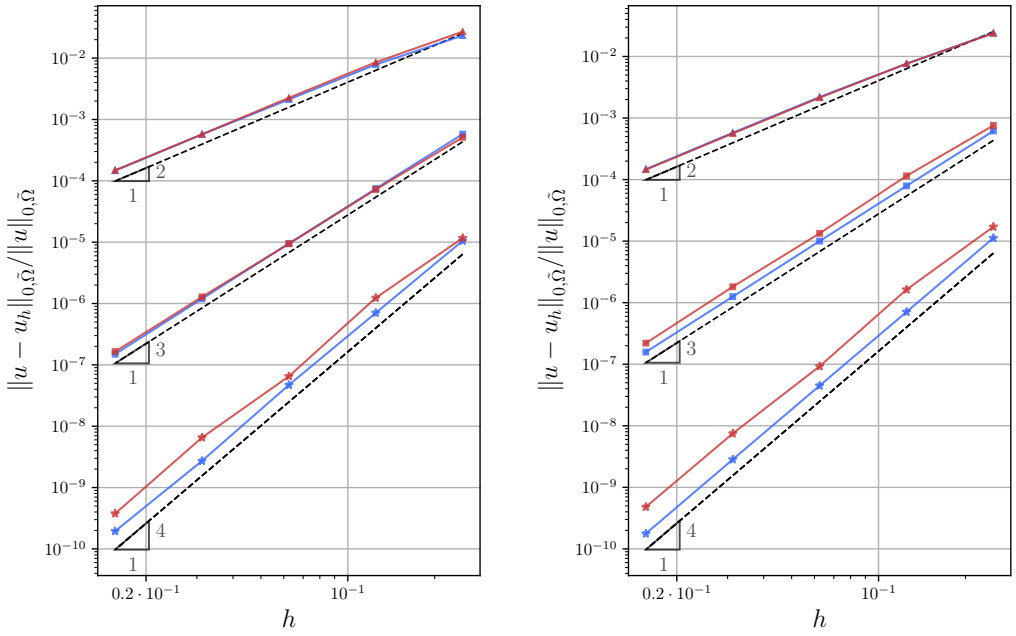
(c) Zoom of curvilinear case.

Figure 5: Selection of the surrogate boundary. Two-dimensional complex embedded holes treated using the SBM. The red solid line denotes the surrogate boundary corresponding to $\lambda = 0$ and the green one to $\lambda = 0.5$ (optimal surrogate boundary). Bottom: an enlargement of Fig. 5a details the differences between the two surrogate boundaries, $\lambda = 0.0$ and $\lambda = 0.5$.

4.4. Neumann boundary conditions

In Section 2.4, we discussed the incorporation of Neumann boundary conditions in the proposed SBM-IGA formulation. The suggested weak formulation approximates the flux t_N with accuracy of order p, due to the vanishing pth-order term in the Taylor expansion. Referring to the geometric setup in Fig. 5b and the manufactured solution in Eq. (33), it is simple to calculate the heat flux t_N on the true boundary by using the true normal vector $\boldsymbol{n}$. We present a comparison between problems utilizing Neumann and Dirichlet boundary conditions in Fig. 9b. As anticipated, the scenarios with Neumann boundary conditions demonstrate convergence of order p, with a loss of one order with respect to the case of Dirichlet boundary conditions. Fig. 9a shows the spatial distribution of error for the Neumann case. This figure reveals that the error is predominantly localized near the surrogate boundary, consistent with the observations made in Remark 1.

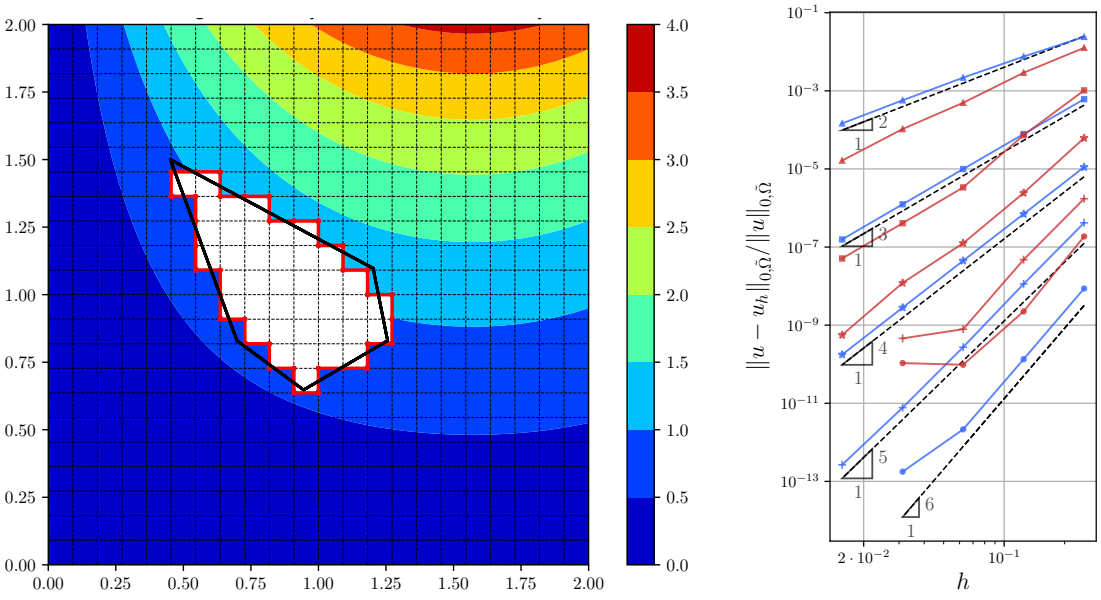
In Fig. 10, we present the normalized H^1- and L^∞-norms for the two previously described cases. Fig. 10a shows



(a) Curvilinear case.

(b) Diamond case.

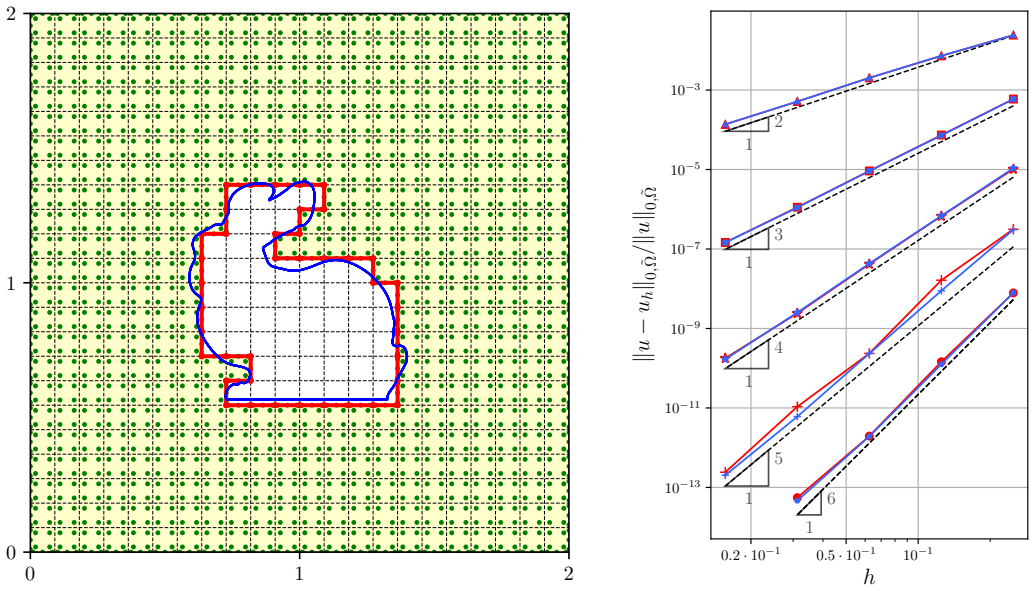
Figure 6: Selection of the surrogate boundary. Convergence study of two cases curvilinear and diamond of Fig. 5a and Fig. 5b respectively, considering $\lambda = 0$ (red solid line) and $\lambda = 0.5$ (blue solid line). The order of the basis functions is indicated by the following symbols: triangles $p = 1$, squares $p = 2$, and stars $p = 3$.



(a) Solution field.

(b) Convergence rates.

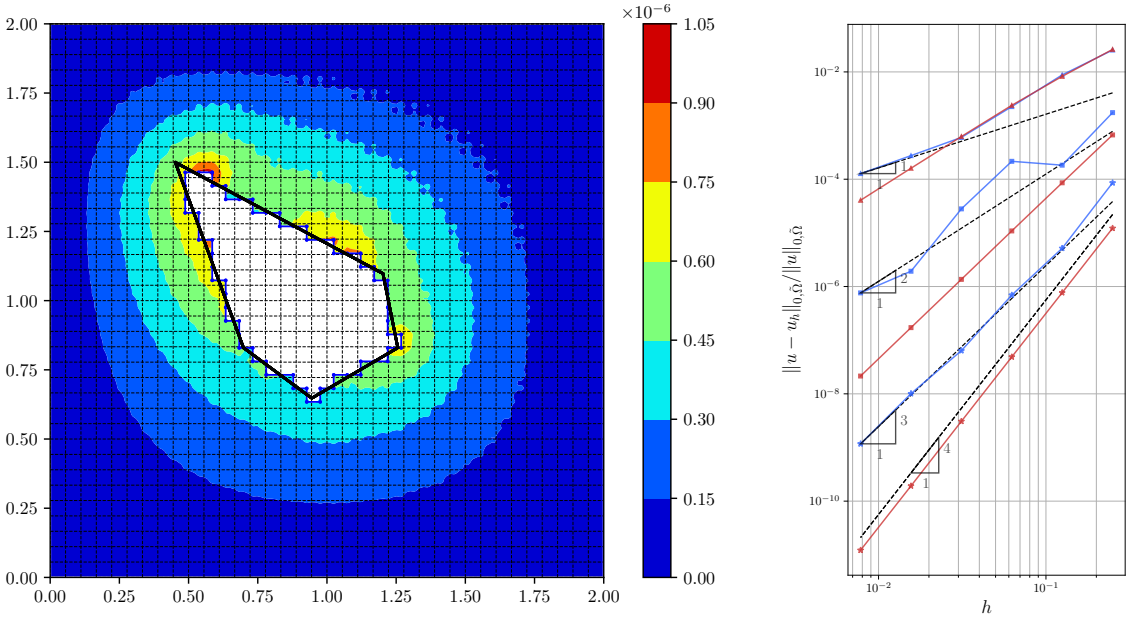
Figure 7: Classic vs Penalty-Free SBM formulations. Left: contour plot of the solution of the diamond case, using $p = 5$ and a uniform knot vector with $h = 0.091$ in both directions. Right: the convergence study of the same case, varying $p = 1, 2, \ldots, 5$ for the two formulations: penalty (red solid line) and penalty-free (blue solid line). The order of the basis functions is indicated by the following symbols: triangles $p = 1$, squares $p = 2$, stars $p = 3$, crosses $p = 4$, and circles $p = 5$.



(a) Computational set-up.

(b) Convergence rates.

Figure 8: SBM vs pseudo body-fitted approach. Left: the two-dimensional Stanford Bunny is used as an internal hole; the true and surrogate boundaries are highlighted in blue and red respectively. Right: the convergence study comparison between the SBM (red solid line) and pseudo body-fitted (blue solid line) approaches varying p. The order of the basis functions is indicated by the following symbols: triangles $p = 1$, squares $p = 2$, stars $p = 3$, crosses $p = 4$, and circles $p = 5$.

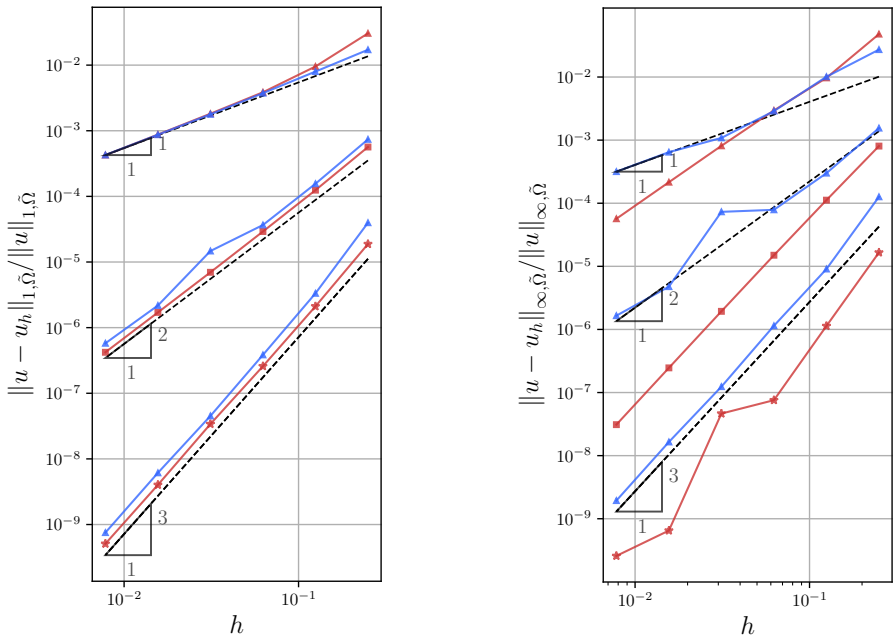


(a) Pointwise error.

(b) Convergence rates.

Figure 9: Neumann boundary conditions. Left: pointwise absolute error in the case Neumann boundary conditions are applied through the SBM. Right: the convergence analysis comparison, using $p = 1$ (triangles symbols), $p = 2$ (square symbols), and $p = 3$ (star symbols) between fully SBM Dirichlet boundary conditions (red solid line) and internal SBM Neumann boundary conditions (blue solid line).

a remarkable result, indicating that the use of SBM Neumann conditions does not impact the convergence order of the error in the H^1-norm. This implies that in both cases we approximate the gradient of the solution with a precision of order p. Conversely, Fig 10b demonstrates that the solution error loses one order of convergence when Neumann conditions are applied. These results are consistent with those in Fig. 9b.



(a) Convergence rates H^1-norm.

(b) Convergence rates L^∞-norm.

Figure 10: Neumann boundary conditions. Comparison of the rates of convergence for the SBM with embedded Dirichlet boundary conditions (red solid line) and embedded Neumann boundary conditions (blue solid line). The optimal surrogate boundary has been chosen in all computations. The symbols are as follows: $p = 1$ (triangles symbols), $p = 2$ (square symbols), and $p = 3$ (star symbols). Left: H^1-norm. Right: L^∞-norm.

4.5. Computing the solution and quantities of interest on the true boundary

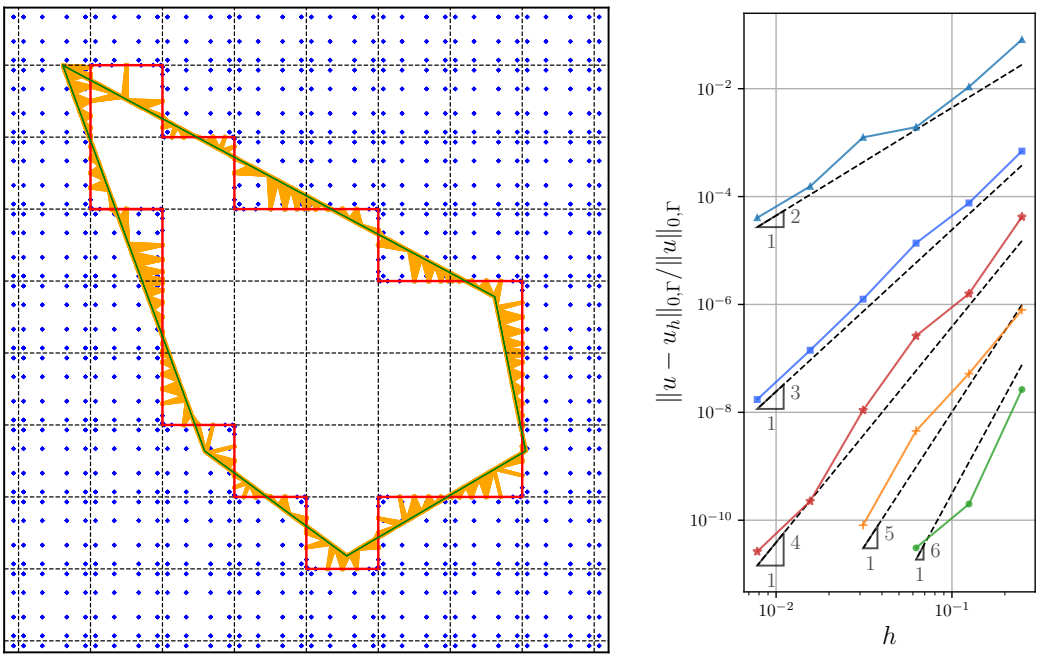
Section 3.3 explained the theory for recovering the solution on the true boundary as a post-process operation. To prove that the integral over the true boundary Γ of the solution using the post-process results of the SBM converges to the true value of the integral we compute

$$\|u - u_h\|_{0,\Gamma} = \sqrt{\int_\Gamma (u - u_h)^2 d\Gamma}, \tag{35}$$

and check that the normalized norm converges optimally. Note that u is the manufactured solution evaluated on the true boundary and u_h is given by reversing the Taylor expansion from the surrogate to the true boundary. In Fig. 11a, we illustrate the technique used for computing u_h. Given any integration rule on Γ, each integration point on Γ is associated with the nearest integration point on $\tilde{\Gamma}_h$, where the solution and its derivatives are available. This approach is sufficient even if it is not considered the exact closest point on $\tilde{\Gamma}_h$. As shown in Fig. 11b, we achieve optimal convergence because the distance $\boldsymbol{d}$ used in the post-processing with Taylor expansion is always smaller than the knot size h. This method can be further improved by finding the actual closest surrogate point, but then the solution must be recovered at that point.

4.6. SBM vs IBRA

In this section, we compare the SBM and the IBRA techniques. The distinction between them lies in their treatment of the cut knot spans. The SBM approach avoids integrating these spans, instead employing a Taylor expansion



(a) Computational set-up.

(b) Convergence rates.

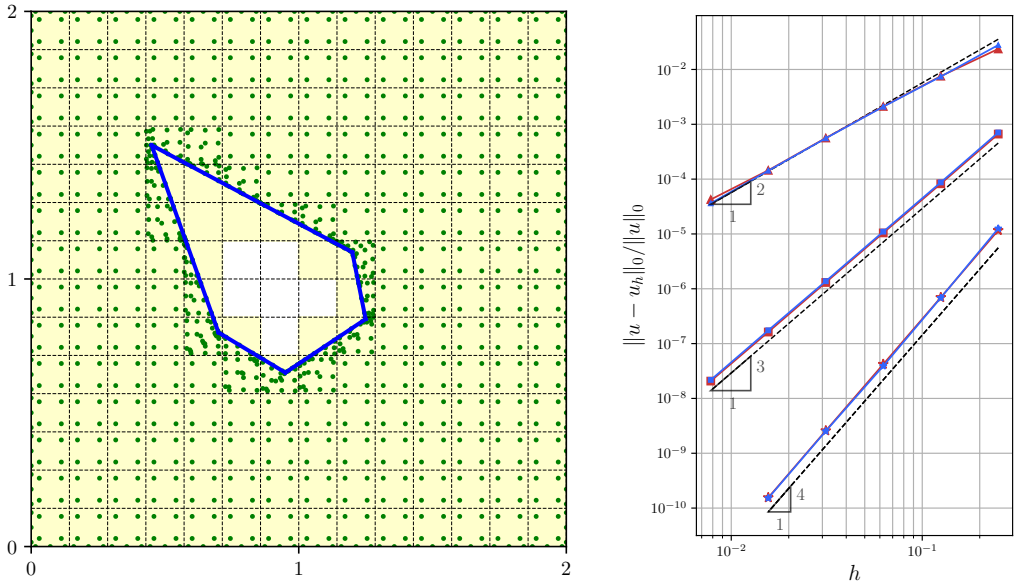
Figure 11: Computing the solution and quantities of interest on the true boundary. Left: the blue dots represent the integration points of the surrogate domain and surrogate boundary, while the orange lines indicate the Taylor expansions from each integration point on the true to the closest integration point on $\tilde{\Gamma}_h$. Right: the corresponding convergence study varying $p = 1, 2, \ldots, 5$.

to account for the boundary shift. Conversely, the IBRA method preserves optimal convergence by integrating the cut knot spans using a triangulation of the active cut portion (Fig. 12a).

In Fig. 12b we report the comparison between the two approaches in the case of the embedded diamond hole. The two approaches yield very similar results in terms of the L^2-norm of the error. The SBM approach produces marginally better outcomes than IBRA, possibly because the error is assessed in the surrogate domain, which might be larger or smaller than the actual one, depending on the resulting optimal surrogate boundary.

The condition number of the matrix associated with the bilinear form $a_h^k(\cdot,\cdot)$ in Eq. (14b) scales like h^{-2} on a quasi-uniform standard FEM mesh of size h [45]. Fig. 13 presents an analysis of the condition number for the diamond hole depicted in Fig. 5b, utilizing the SBM and IBRA respectively to apply Dirichlet boundary conditions to an internal hole. The condition numbers $\kappa(\mathbf{A})$ of the SBM formulation are in general order of magnitude smaller than the ones of the IBRA formulation, and grow asymptotically as h^{-2} (see Fig. 13b). This observation aligns with the findings of Collins *et al.* [45], where a similar scenario was analyzed within the FEM. A notable initial increase in the condition number correlates with the deactivation of basis functions entirely within the hole, a phenomenon that intensifies with higher values of p. This intensification is likely due to the activation of basis functions whose support lies on few active knot spans, resulting in degrees of freedom that poorly contribute. However, upon refinement, the growth of the condition number adheres to the h^{-2} behavior. In Fig. 13a the condition number of the same example is reported using the IBRA approach.

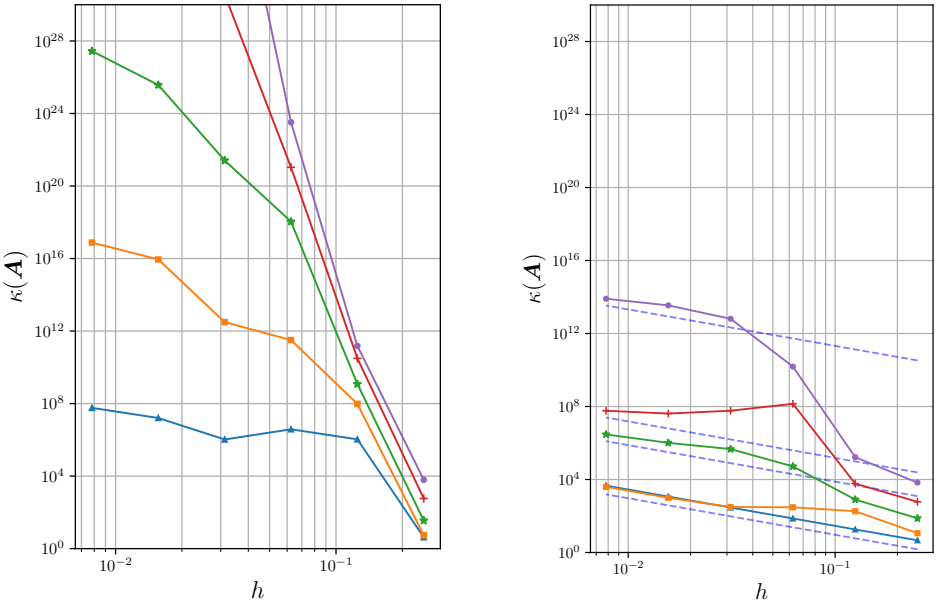
Finally, we highlight that these results prove that the SBM, in combination with either the IGA or FEM, does not suffer from the so-called small-cut cell instability, a common issue in CutFEM strategies and other methods that involve integrating over the cut elements. Moreover, the IBRA-type discretization is known to suffer from this instability, rendering iterative solvers ineffective for solving the associated linear systems. Conversely, the SBM approach leads to systems of equations that could be consistently solved with both direct and iterative solvers.



(a) IBRA Computational set-up.

(b) Convergence rates.

Figure 12: SBM vs IBRA. Left: the diamond shape applying the IBRA technique; the green dots denote the integration points, while the solid blue line highlights the true boundary. Right: the convergence analysis of the same case, using $p = 1$ (triangular symbols), $p = 2$ (square symbols), and $p = 3$ (star symbols) and comparing the IBRA approach (red solid line) and the SBM (blue solid line) with the optimal surrogate boundary.



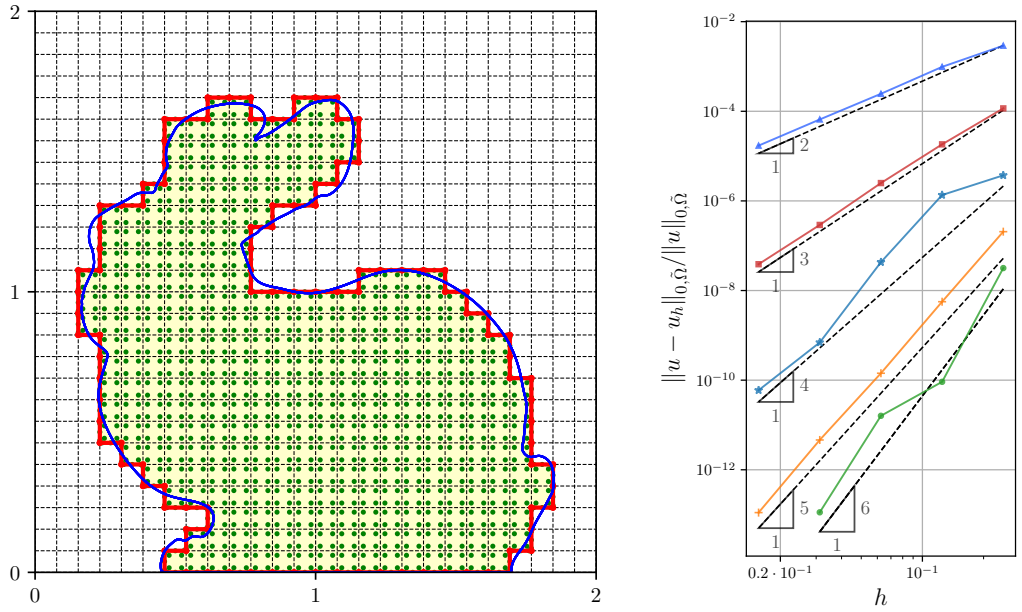
(a) IBRA

(b) SBM

Figure 13: SBM vs IBRA. Condition number analysis, considering IBRA (left panel) and the SBM (right panel), varying the basis function order for the diamond case in Fig. 5. The order of the basis functions is indicated by the following symbols: triangles $p = 1$, squares $p = 2$, stars $p = 3$, crosses $p = 4$, and circles $p = 5$. The dashed lines denote the reference slope h^{-2}.

4.7. Capabilities of the proposed approach

In this section, we consider the case in which the SBM is applied to a complex external boundary, instead of an internal hole. This scenario is common in structural mechanics and fluid-structure interaction applications. Consider the case depicted in Fig. 14a, in which the SBM is applied on the external boundary of the two-dimensional Stanford Bunny. Therefore the surrogate domain $\tilde{\Omega}_h$, represented by the yellow knot spans in the figure, consists of the internal knot spans that are enclosed by the surrogate boundary and the bottom body-fitted boundary. The results in Fig. 14b demonstrate optimal convergence overall; however, it is observed that the convergence exhibits a slightly oscillatory behavior for $p = 3$ and $p = 5$. This phenomenon was not encountered in scenarios involving internal boundaries but is not uncommon for the traditional SBM and for other types of immersed methods.



(a) Computational set-up.

(b) Convergence rates.

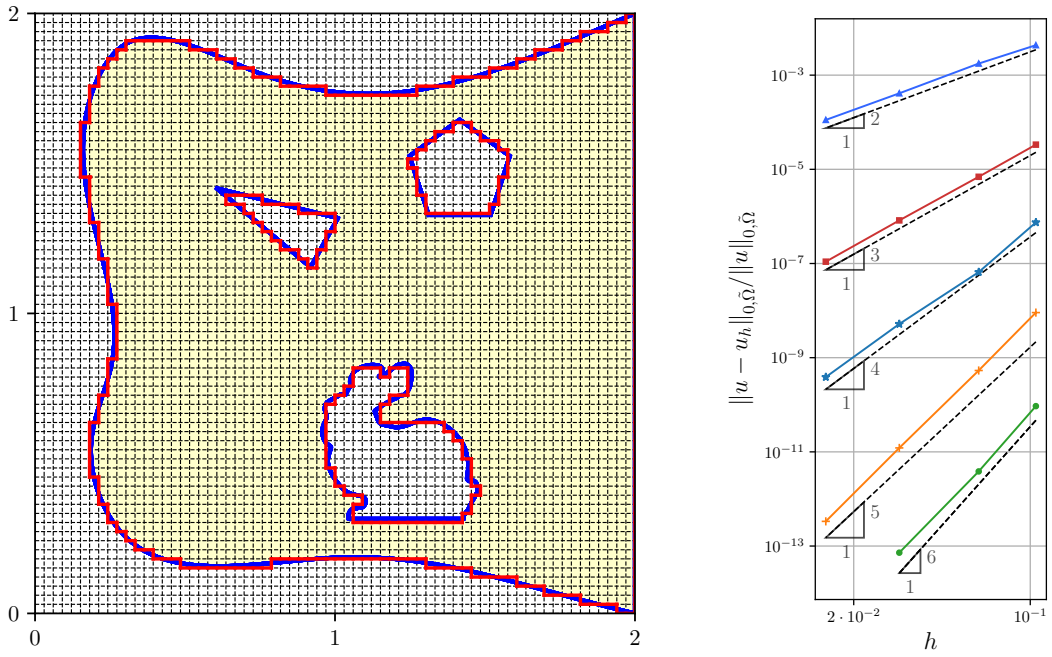
Figure 14: Capabilities of the proposed approach. Left: the two-dimensional Stanford Bunny is used as the external embedded boundary with the SBM considering the optimal surrogate boundary (red solid line), in blue is highlighted the true boundary, and the green dots are the integration points. Right: the convergence study varying $p = 1, 2, \ldots, 5$.

In Fig. 15a a more complex case is shown. In particular, the problem involves an external curvilinear boundary partially treated using the SBM, and three embedded inner loops which form complex holes in the domain. Fig. 15b contains a convergence study that shows optimal convergence for all the order of the B-Spline basis functions, which range from 1 to 5.

We include here a simple example demonstrating the applicability of these ideas to three-dimensional problems. The three-dimensional extension of the concepts discussed in Section 3.2 requires some additional but otherwise natural modifications. Consider a cubic domain with a central spherical hole and the Poisson problem. The three-dimensional manufactured solution is given by

$$u(x, y) = \sin(x\sqrt{2}) \sinh(y) \cosh(z), \tag{36}$$

which corresponds to a zero-forcing term. Upon identifying the surrogate surface, Dirichlet boundary conditions are weakly imposed using the Taylor expansion in three dimensions. In Fig. 16a we report a picture of the surrogate surface and in Fig. 16b the convergence study for linear, quadratic, and cubic B-Splines basis functions.



(a) Computational set-up.

(b) Convergence rates.

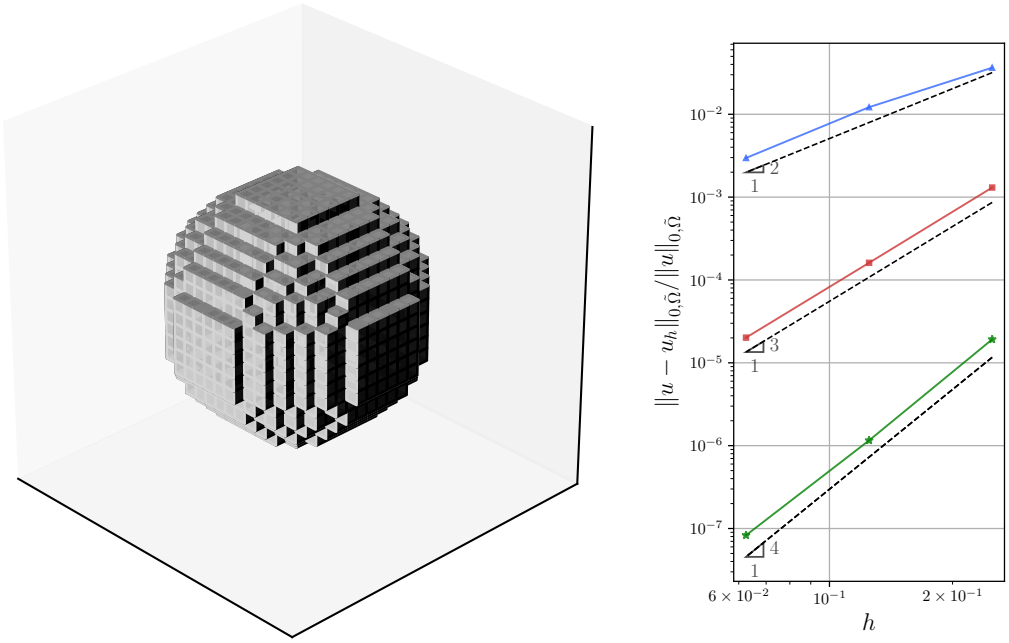
Figure 15: Capabilities of the proposed approach. Left: a complex example composed of an external and three internal embedded boundaries treated with the SBM. In blue it is highlighted the true boundary, and in red the optimal surrogate boundary. Right: the corresponding convergence study varying $p = 1, 2, \ldots, 5$.

5. Conclusions

Based on the comprehensive analysis and results presented throughout the paper, we conclude that the integration of the SBM within the IGA framework represents a significant advancement in computational mechanics. This novel approach elegantly combines the geometric flexibility and precision of IGA with the robustness and simplicity of the SBM, facilitating the direct use of CAD-based geometries in simulations without the need for complex meshing or re-meshing processes.

The implementation of the SBM within IGA demonstrates optimal convergence rates for a variety of problems, including complex geometries with both internal and external boundaries. This optimal convergence is maintained across different orders of basis functions, affirming the method's efficacy and accuracy. Our research underscores the stability and consistency of a penalty-free formulation in the SBM-IGA context. This formulation not only simplifies the application of boundary conditions but also negates the need for penalty parameter calibration, thereby enhancing the method's user-friendliness and robustness. Moreover, the condition number analysis demonstrates that the SBM-IGA approach effectively avoids the small cut-cell problem. This issue is a frequent challenge in numerical methods dealing with embedded boundaries, such as IBRA, which attempt to integrate over the cut elements. This ensures that the method remains stable and suitable for use with both direct and iterative solvers.

While this study has established a solid foundation for integrating the SBM within IGA, further exploration is necessary for fully exploiting its potential. Future efforts could be directed towards challenging the method with complex 3D geometries, extending the method to work with true boundaries and true surfaces described by NURBS. Additionally, there is a need to enhance the efficiency of the integration scheme and to investigate its suitability for a wider range of physical phenomena.



(a) Surrogate Surface.

(b) Convergence rates.

Figure 16: Capabilities of the proposed approach. Left: the surrogate surface of the inner spherical hole. Right: the convergence study, using $p = 1$ (triangles symbols), $p = 2$ (square symbols), and $p = 3$ (star symbols) applying SBM Dirichlet boundary conditions and using the optimal surrogate boundary.

CRediT authorship contribution statement

Nicolò Antonelli: Software, Validation, Investigation, Writing - Original Draft, Visualization. **Ricky Aristio:** Software, Investigation. **Andrea Gorgi:** Software, Investigation. **Rubén Zorrilla:** Software, Writing - Review & Editing, Supervision. **Riccardo Rossi:** Conceptualization, Writing - Review & Editing, Supervision, Funding acquisition. **Guglielmo Scovazzi:** Conceptualization, Formal analysis, Writing - Review & Editing. **Roland Wüchner:** Funding acquisition.

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.

Data availability

Data will be made available on request.

Acknowledgments

The authors gratefully acknowledge the Design for IGA-type discretization workflows (GECKO) project. The Design for IGA-type discretization workflows has received funding from the European Union's Horizon Europe research and Innovation programme under grant agreement No 101073106 Call: HORIZON-MSCA-2021-DN-01.
This research is partly supported by the European High-Performance Computing Joint Undertaking (JU) through the project eFlows4HPC (grant agreement No 955558). The JU receives support from the European Union Horizon

2020 research and innovation program and Spain, Germany, France, Italy, Poland, Switzerland, Norway. The R&D project PCI2021-121944, financed by MCIN/AEI/10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR" is also acknowledged.
Rubén Zorrilla and Riccardo Rossi acknowledge the research project TED2021-130471B-I00, financed by MICIU/ AEI /10.13039/501100011033 and by the "European Union Next GenerationEU/ PRTR".
Guglielmo Scovazzi is supported by the Army Research Office (ARO) under Grant W911NF-18-1-0308, the National Science Foundation under Grant 2207164 (Division of Mathematical Sciences – DMS), and by Lawrence Livermore National Laboratory under an LDRD Grant.

References

[1] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4135–4195, 2005.

[2] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195(41):5257–5296, 2006.

[3] Yuri Bazilevs, Lourenco Beirao da Veiga, John A Cottrell, Thomas JR Hughes, and Giancarlo Sangalli. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16(07):1031–1090, 2006.

[4] Y. Bazilevs, V. M. Calo, Y. Zhang, and T. J. R. Hughes. Isogeometric Fluid–structure Interaction Analysis with Applications to Arterial Blood Flow. *Computational Mechanics*, 38(4):310–322, 2006.

[5] Yongjie Zhang, Yuri Bazilevs, Samrat Goswami, Chandrajit L. Bajaj, and Thomas J.R. Hughes. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering*, 196(29):2943–2959, 2007.

[6] J.A. Cottrell, T.J.R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196(41):4160–4183, 2007.

[7] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with Kirchhoff–Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198(49):3902–3914, 2009.

[8] D.J. Benson, Y. Bazilevs, M.C. Hsu, and T.J.R. Hughes. Isogeometric shell analysis: The Reissner–Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 199(5):276–289, 2010. Computational Geometry and Analysis.

[9] Kenji Takizawa, Yuri Bazilevs, Tayfun E. Tezduyar, Ming-Chen Hsu, and Takuya Terahara. Computational Cardiovascular Medicine With Isogeometric Analysis. *Journal of Advanced Engineering and Computation*, 6(3):167–199, 2022.

[10] Massimo Carraturo, Carlotta Giannelli, Alessandro Reali, and Rafael Vázquez. Suitably graded THB-spline refinement and coarsening: Towards an adaptive isogeometric analysis of additive manufacturing processes. *Computer Methods in Applied Mechanics and Engineering*, 348:660–679, 2019.

[11] Carlotta Giannelli, Bert Jüttler, and Hendrik Speleers. THB-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29(7):485–498, 2012. Geometric Modeling and Processing 2012.

[12] Thomas W. Sederberg, G. Thomas Finnigan, Xin Li, Hongwei Lin, and Heather Ipson. Watertight trimmed NURBS. *ACM Transactions on Graphics*, 27(3):1–8, 2008.

[13] René R. Hiemstra, Kendrick M. Shepherd, Michael J. Johnson, Lulin Quan, and Thomas J.R. Hughes. Towards untrimmed NURBS: CAD embedded reparameterization of trimmed B-rep geometry using frame-field guided global parameterization. *Computer Methods in Applied Mechanics and Engineering*, 369:113227, 2020.

[14] Thomas W. Sederberg, Jianmin Zheng, Almaz Bakenov, and Ahmad Nasri. T-splines and T-NURCCs. *ACM Transactions on Graphics*, 22(3):477–484, 2003.

[15] Thomas W. Sederberg, David L. Cardon, G. Thomas Finnigan, Nicholas S. North, Jianmin Zheng, and Tom Lyche. T-spline simplification and local refinement. *ACM Transactions on Graphics*, 23(3):276–283, 2004.

[16] Hugo Casquero, Lei Liu, Yongjie Zhang, Alessandro Reali, Josef Kiendl, and Hector Gomez. Arbitrary-degree T-splines for isogeometric analysis of fully nonlinear Kirchhoff–Love shells. *Computer-Aided Design*, 82:140–153, 2017. Isogeometric Design and Analysis.

[17] Hugo Casquero, Xiaodong Wei, Deepesh Toshniwal, Angran Li, Thomas J.R. Hughes, Josef Kiendl, and Yongjie Jessica Zhang. Seamless integration of design and Kirchhoff–Love shell analysis using analysis-suitable unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering*, 360:112765, 2020.

[18] Xiaodong Wei, Yongjie Jessica Zhang, Deepesh Toshniwal, Hendrik Speleers, Xin Li, Carla Manni, John A. Evans, and Thomas J.R. Hughes. Blended B-spline construction on unstructured quadrilateral and hexahedral meshes with optimal convergence rates in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 341:609–639, 2018.

[19] Derek C. Thomas, Luke Engvall, Steven K. Schmidt, Kevin Tew, and Michael A. Scott. U-splines: Splines over unstructured meshes. *Computer Methods in Applied Mechanics and Engineering*, 401:115515, 2022.

[20] Zuowei Wen, Md. Sadman Faruque, Xin Li, Xiaodong Wei, and Hugo Casquero. Isogeometric analysis using G-spline surfaces with arbitrary unstructured quadrilateral layout. *Computer Methods in Applied Mechanics and Engineering*, 408:115965, 2023.

[21] Hyun-Jung Kim, Yu-Deok Seo, and Sung-Kie Youn. Isogeometric analysis for trimmed CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 198(37):2982–2995, 2009.

[22] Robert Schmidt, Roland Wüchner, and Kai-Uwe Bletzinger. Isogeometric analysis of trimmed NURBS geometries. *Computer Methods in Applied Mechanics and Engineering*, 241-244:93–111, 2012.

[23] Dominik Schillinger, Luca Dedè, Michael A. Scott, John A. Evans, Michael J. Borden, Ernst Rank, and Thomas J.R. Hughes. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 249-252:116–150, 2012. Higher Order Finite Element and Isogeometric Methods.

[24] M. Ruess, D. Schillinger, Y. Bazilevs, V. Varduhn, and E. Rank. Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell method. *International Journal for Numerical Methods in Engineering*, 95(10):811–846, 2013.

[25] Jamshid Parvizian, Alexander Düster, and Ernst Rank. Finite cell method: h-and p-extension for embedded domain problems in solid mechanics. *Computational Mechanics*, 41(1):121–133, 2007.

[26] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, and A. Düster. Geometric modeling, isogeometric analysis and the finite cell method. *Computer Methods in Applied Mechanics and Engineering*, 249-252:104–115, 2012.

[27] David Kamensky, Ming-Chen Hsu, Dominik Schillinger, John A. Evans, Ankush Aggarwal, Yuri Bazilevs, Michael S. Sacks, and Thomas J.R. Hughes. An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053, 2015.

[28] David Kamensky, Ming-Chen Hsu, Yue Yu, John A. Evans, Michael S. Sacks, and Thomas J.R. Hughes. Immersogeometric cardiovascu-

lar fluid–structure interaction analysis with divergence-conforming B-splines. *Computer Methods in Applied Mechanics and Engineering*, 314:408–472, 2017.
[29] Alessandro Nitti, Josef Kiendl, Alessandro Reali, and Marco D. de Tullio. An immersed-boundary/isogeometric method for fluid–structure interaction involving thin shells. *Computer Methods in Applied Mechanics and Engineering*, 364:112977, 2020.
[30] M. Breitenberger, A. Apostolatos, B. Philipp, R. Wüchner, and K.-U. Bletzinger. Analysis in computer aided design: Nonlinear isogeometric B-Rep analysis of shell structures. *Computer Methods in Applied Mechanics and Engineering*, 284:401–457, 2015. Isogeometric Analysis Special Issue.
[31] Tobias Teschemacher, AM Bauer, Thomas Oberbichler, Micheal Breitenberger, Riccardo Rossi, Roland Wüchner, and Kai-Uwe Bletzinger. Realization of CAD-integrated shell simulation based on isogeometric B-Rep analysis. *Advanced Modeling and Simulation in Engineering Sciences*, 5:1–54, 2018.
[32] Tobias Teschemacher, Anna M Bauer, Ricky Aristio, Manuel Meßmer, Roland Wüchner, and Kai-Uwe Bletzinger. Concepts of data collection for the cad-integrated isogeometric analysis. *Engineering with Computers*, 38(6):5675–5693, 2022.
[33] Manuel Meßmer, Tobias Teschemacher, Lukas F. Leidinger, Roland Wüchner, and Kai-Uwe Bletzinger. Efficient CAD-integrated isogeometric analysis of trimmed solids. *Computer Methods in Applied Mechanics and Engineering*, 400:115584, 2022.
[34] Manuel Meßmer, Stefan Kollmannsberger, Roland Wüchner, and Kai-Uwe Bletzinger. Robust numerical integration of embedded solids described in boundary representation. *Computer Methods in Applied Mechanics and Engineering*, 419:116670, 2024.
[35] Erik Burman. Ghost penalty. *Comptes Rendus. Mathématique*, 348(21-22):1217–1220, 2010.
[36] Santiago Badia, Eric Neiva, and Francesc Verdugo. Linking ghost penalty and aggregated unfitted methods. *Computer Methods in Applied Mechanics and Engineering*, 388:114232, 2022.
[37] A. Main and G. Scovazzi. The shifted boundary method for embedded domain computations. Part I: Poisson and Stokes problems. *Journal of Computational Physics*, 372:972–995, 2018.
[38] A. Main and G. Scovazzi. The shifted boundary method for embedded domain computations. Part II: Linear advection–diffusion and incompressible Navier–Stokes equations. *Journal of Computational Physics*, 372:996–1026, 2018.
[39] R. Zorrilla, R. Rossi, R. Wüchner, and E. Oñate. An embedded finite element framework for the resolution of strongly coupled fluid–structure interaction problems. application to volumetric and membrane-like structures. *Computer Methods in Applied Mechanics and Engineering*, 368:113179, 2020.
[40] Nabil M. Atallah, Claudio Canuto, and Guglielmo Scovazzi. The shifted boundary method for solid mechanics. *International Journal for Numerical Methods in Engineering*, 122(20):5935–5970, 2021.
[41] Nabil M. Atallah and Guglielmo Scovazzi. Nonlinear elasticity with the Shifted Boundary Method. *Computer Methods in Applied Mechanics and Engineering*, 426:116988, 2024.
[42] Efthymios N. Karatzas, Giovanni Stabile, Leo Nouveau, Guglielmo Scovazzi, and Gianluigi Rozza. A reduced-order shifted boundary method for parametrized incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 370:113273, 2020.
[43] Nabil M. Atallah, Claudio Canuto, and Guglielmo Scovazzi. The second-generation Shifted Boundary Method and its numerical analysis. *Computer Methods in Applied Mechanics and Engineering*, 372:113341, 2020.
[44] Nabil M Atallah, Claudio Canuto, and Guglielmo Scovazzi. Analysis of the shifted boundary method for the poisson problem in domains with corners. *Mathematics of Computation*, 90(331):2041–2069, 2021.
[45] J. Haydel Collins, Alexei Lozinski, and Guglielmo Scovazzi. A penalty-free Shifted Boundary Method of arbitrary order. *Computer Methods in Applied Mechanics and Engineering*, 417:116301, 2023. A Special Issue in Honor of the Lifetime Achievements of T. J. R. Hughes.
[46] Cheng-Hau Yang, Kumar Saurabh, Guglielmo Scovazzi, Claudio Canuto, Adarsh Krishnamurthy, and Baskar Ganapathysubramanian. Optimal surrogate boundary selection and scalability studies for the shifted boundary method on octree meshes. *Computer Methods in Applied Mechanics and Engineering*, 419:116686, 2024.
[47] Rubén Zorrilla, Riccardo Rossi, Guglielmo Scovazzi, Claudio Canuto, and Antonio Rodríguez-Ferran. A shifted boundary method based on extension operators. *Computer Methods in Applied Mechanics and Engineering*, 421:116782, 2024.
[48] Nabil M. Atallah, Claudio Canuto, and Guglielmo Scovazzi. The high-order Shifted Boundary Method and its analysis. *Computer Methods in Applied Mechanics and Engineering*, 394:114885, 2022.
[49] Pooyan Dadvand, Riccardo Rossi, and Eugenio Oñate. An object-oriented environment for developing finite element codes for multi-disciplinary applications. *Archives of computational methods in engineering*, 17:253–297, 2010.
[50] P. Dadvand, R. Rossi, M. Gil, X. Martorell, J. Cotela, E. Juanpere, S.R. Idelsohn, and E. Oñate. Migration of a generic multi-physics framework to HPC environments. *Computers & Fluids*, 80:301–309, 2013. Selected contributions of the 23rd International Conference on Parallel Fluid Dynamics ParCFD2011.
[51] Maurice G. Cox. The numerical evaluation of B-splines. *IMA Journal of Applied Mathematics*, 10(2):134–149, 1972.
[52] Carl De Boor. On calculating with B-splines. *Journal of Approximation theory*, 6(1):50–62, 1972.
[53] Les Piegl and Wayne Tiller. *The NURBS book*. Springer Science & Business Media, 2012.