Journal of Computational Physics 528 (2025) 113807

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A high-order immersed boundary method to approximate flow problems in domains with curved boundaries

S. Colombo [a,*], G. Rubio [a,b], J. Kou [c], E. Valero [a,b], R. Codina [d,e], E. Ferrer [a,b]

[a] ETSIAE-UPM-School of Aeronautics, Universidad Politécnica de Madrid, Madrid, Spain
[b] Center for Computational Simulation, Universidad Politécnica de Madrid, Madrid, Spain
[c] School of Aeronautics, Northwestern Polytechnical University, Xi'an, China
[d] Universitat Politècnica de Catalunya, Barcelona, Spain
[e] Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), Barcelona, Spain

ARTICLE INFO

Keywords:
Immersed boundary method
Curved boundary conditions
High-order h/p solvers
Discontinuous Galerkin
Horses3D

ABSTRACT

High-order h/p solvers in computational fluid dynamics offer scalability, efficiency, and superior error reduction compared to traditional low-order methods. Immersed boundary methods eliminate the need for body-fitted meshes but often degrade the order of the solution near boundaries, which can damage the overall accuracy of the high-order solver. This paper presents a new approach to impose boundary conditions in high-order finite element or finite volume flow solvers that retain high-order $P+1$ convergence, where P is the polynomial order. Furthermore, the methodology takes into account curved boundary conditions without loss in accuracy. It introduces a surrogate boundary that eliminates instabilities due to badly cut elements.

We test the methodology using a high-order discontinuous Galerkin framework to solve purely elliptic problems and the compressible Navier-Stokes equations (2D and 3D), to show that we retain the formal order of convergence $P+1$. Finally, we compare the results with a volume penalization approach and show that spurious pressure oscillations on the immersed boundary are eliminated when the proposed methodology is used.

Contents

1. Introduction . . . 2
2. High-order discontinuous Galerkin solvers . . . 3
3. A high-order immersed boundary method . . . 4
   3.1. Identification of the shifted faces . . . 4
   3.2. Computation of the shifted faces state . . . 5
   3.3. Dirichlet boundary condition for the Poisson equation . . . 7
   3.4. Neumann boundary condition for the Poisson equations . . . 8
   3.5. Surface representation . . . 8
4. Test cases . . . 9
   4.1. 1D linear convection-diffusion . . . 9
   4.2. 2D Poisson equation with Dirichlet boundary conditions . . . 10

* Corresponding author.
E-mail address: stefano.colombo@upm.es (S. Colombo).

https://doi.org/10.1016/j.jcp.2025.113807
Received 17 July 2024; Received in revised form 28 January 2025; Accepted 30 January 2025

Available online 3 February 2025

4.3. 2D Poisson equation with Neumann boundary conditions . . . 10
4.4. Steady heat equation and curved boundaries . . . 12
4.5. Navier-Stokes equations . . . 12
4.5.1. Two dimensional circular cylinder . . . 13
4.5.2. Three dimensional sphere . . . 14
5. Conclusions . . . 16
CRediT authorship contribution statement . . . 16
Declaration of competing interest . . . 16
Acknowledgements . . . 17
Appendix A. Volume penalization immersed boundary . . . 17
Appendix B. Bassi-Rebay 1 . . . 17
Appendix C. Some consideration about the identification of the shifted faces . . . 18
Appendix D. Effect of the variation of the parameter α . . . 18
Data availability . . . 19
References . . . 19

1. Introduction

In the field of computational fluid dynamics, high-order h/p solvers (e.g., based on discontinuous Galerkin) have emerged as a promising approach due to their scalability in modern architectures [1] and superior efficiency for a given level of accuracy [2,3]. These methods require an accurate geometric description of the boundaries to maintain high-order accuracy near them. This is accomplished by generating a body-fitted mesh around curved walls, where the order of curvature used to represent the wall is the same as the order of the solution inside the element (i.e., an isoparametric representation is assumed). Although the generation of linear meshes has reached a high level of maturity, even for complex geometries, the same cannot be said for curvilinear meshes in complex geometries. In fact, the generation of body-fitted curved meshes is a bottleneck in the simulation process and remains a vibrant area of research; see, for example, [4,5].

Immersed boundary methods (IBMs) (or embedded methods or unfitted boundary methods) do not require body-fitted meshes. Originally introduced by Peskin [6], IBM simulates the presence of the geometries through artificial mechanisms, thus avoiding the need for body-fitted curved meshes and allowing complex flows using simple Cartesian grids. IBMs have seen a surge in popularity in recent years [7–9], and various techniques have been developed. In particular, current IBM approaches include cut-cell [10–12], direct forcing [13–16], ghost-cell [17,18], volume penalization [19–22] and its extension to high-order solvers and error analysis [22–26]. While IBMs circumvent the need for body-fitted meshes, they often do so at the expense of accuracy near surfaces; see, for example, the volume penalization approach in [22,20,27]. Other methods can maintain accuracy at the cost of the increased complexity and degraded conditioning introduced by cut cells. In fact, cut cells are prone to instabilities when small cut elements (at the intersection between the background mesh and the immersed boundary) emerge and attempts have been made to bypass the problem by introducing stabilization [28–30].

To address these difficulties, the finite-volume community has developed interpolation schemes to recover second-order accuracy near boundaries [31–33], while the finite-element community has proposed, among others, the shifted boundary method (SBM) [34,35] to retrieve higher accuracy near immersed boundaries. Here, the idea is to shift the boundary, where the boundary conditions are applied, from the actual surface to a surrogate surface. The conditions imposed on the surrogate surface are computed through a Taylor expansion.

In this paper, we propose a new technique, based on the recent work by Funada et al. [36], in which ideas from shifted boundaries and interpolation from finite volumes are combined. The key ideas of the method we propose are the following. Suppose that the flow domain is Ω, with boundary Γ and we have a mesh for a background domain Ω_s that covers Ω. The steps to follow are:

- Create a surrogate boundary Γ_s made of element boundaries of the background mesh,'close' to the physical boundary Γ. It can be both interior or exterior to the computational domain Ω. Working with this boundary *will eliminate possible numerical instabilities due to badly cut elements*.
- Reconstruct the unknowns of the problem along a straight line normal to the true boundary Γ from both interior degrees of freedom of a chosen donor cell (element) and the boundary values to be prescribed, either of the unknowns or of their derivatives. Thus, *both Dirichlet and Neumann boundary conditions can be considered*.
- Evaluate the reconstructed unknowns on the surrogate boundary Γ_s and prescribe these values weakly, for example by modifying appropriately the flux vector of the numerical scheme.

Using high-order Lagrange interpolators, we enable consistent high-order solutions. The proposed reconstruction can be seen as an adaptation of the method developed for finite volume in [37,38] to high-order solvers or as a high-order modification of the shifted boundary method (as developed by the finite element community). Furthermore, the method is applicable to curved boundaries without loss of accuracy. To the best of our knowledge, this is the first time such an approach has been developed for high-order h/p methods (here a discontinuous Galerkin method). In this work we develop the method in the context of the compressible Navier-Stokes equations, describing also its application to the simple Poisson's problem; nevertheless, the idea is general and can be applied

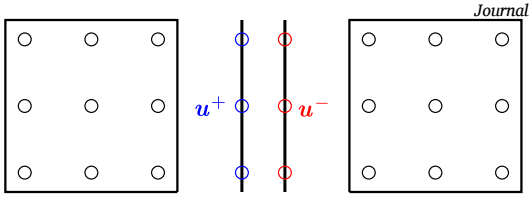


Fig. 1. Discontinuous Galerkin boundaries include two states $\boldsymbol{u}^+$ and $\boldsymbol{u}^-$, which are to evaluate the numerical fluxes at the elements interfaces.

to other problems as well. This method is robust and allows for the recovery of high-order accuracy in straight and curved geometric boundaries. Finally, in this work, we focus on discontinuous Galerkin discretizations. However, the proposed method can easily accommodate other numerical strategies (e.g., finite volumes, or high order continuous finite elements) when using weak enforcement of boundary conditions. Let us also remark that the method leads to non-symmetric algebraic problems even for symmetric boundary value problems, as the SBM or the purely algebraic approach proposed in [39]. Nevertheless, we shall consider explicit time integration schemes even to approximate stationary problems, and this non-symmetry will not be an issue.

The remainder of the paper is organized as follows. Section 2 provides a brief overview of the numerical framework used throughout the paper and how the boundary conditions are imposed. In Section 3, the innovative methodology is presented and explained. Section 3.5 gives some consideration about the discretization of the body surface, while Section 4 is dedicated to various test cases of increasing complexity. Finally, Section 5 presents the conclusions.

2. High-order discontinuous Galerkin solvers

All work described throughout the article has been implemented in the open source Horses3D solver [40], which is a high-order spectral discontinuous Galerkin solver capable of solving a variety of flow applications, including compressible flows (with or without shocks), incompressible flows, various RANS and LES turbulence models, particle dynamics, multiphase flows, and aeroacoustics. Horses3D handles body-fitted curved boundaries and also includes a volume penalization IBM. Both of these functionalities will be compared with the proposed methodology. Here, we provide only a brief description of the discontinuous Galerkin method implemented in Horses3D in the context of flow problems.

We consider the general case of a second-order time-dependent partial differential equation:

$$\tilde{\boldsymbol{u}}_t + \nabla_{\boldsymbol{x}} \cdot \tilde{\boldsymbol{F}}_e = \nabla_{\boldsymbol{x}} \cdot \tilde{\boldsymbol{F}}_v + \tilde{\boldsymbol{S}}, \quad \boldsymbol{x} \in \Omega, \quad t > 0 \tag{1}$$

where $\tilde{\boldsymbol{u}}$ is the state vector of conservative variables, $\tilde{\boldsymbol{F}}_e$ and $\tilde{\boldsymbol{F}}_v$ are the inviscid and viscous fluxes respectively, and $\tilde{\boldsymbol{S}}$ is a source term. The physical domain is subdivided into nonoverlapping curvilinear hexahedral elements, e, which are geometrically transformed to a reference element, el. Transformation is performed using a polynomial transfinite mapping that relates the physical coordinates $\boldsymbol{x}$ and the local reference coordinates $\boldsymbol{\xi} \in [-1,1]^d$, where d is the spatial dimension of the problem. Additionally, the state vector, fluxes, and source term are approximated by polynomials using a tensor product of the one-dimensional Lagrange basis, resulting in:

$$J\boldsymbol{u}_t + \nabla_{\boldsymbol{\xi}} \cdot \boldsymbol{F}_e = \nabla_{\boldsymbol{\xi}} \cdot \boldsymbol{F}_v + J\boldsymbol{S}, \tag{2}$$

where J is the Jacobian of the transfinite mapping, $\nabla_{\boldsymbol{\xi}}$ is the differential operator in the reference space, $\boldsymbol{S}$ is a source term and $\boldsymbol{F}_e$, $\boldsymbol{F}_v$ are the contravariant fluxes [41].

The weak formulation of the discontinuous Galerkin is obtained by multiplying Eq. (2) by a set of test functions ϕ_i for $0 \le i \le P$ and integrating over el:

$$\int_{el} \phi_i J\boldsymbol{u}_t = -\oint_{\partial el} \phi_i \left(\boldsymbol{F}_e - \boldsymbol{F}_v\right) \cdot \hat{\boldsymbol{n}} + \int_{el} \nabla_{\boldsymbol{\xi}} \phi_i \cdot \left(\boldsymbol{F}_e - \boldsymbol{F}_v\right) + \int_{el} \phi_i J\boldsymbol{S} \tag{3}$$

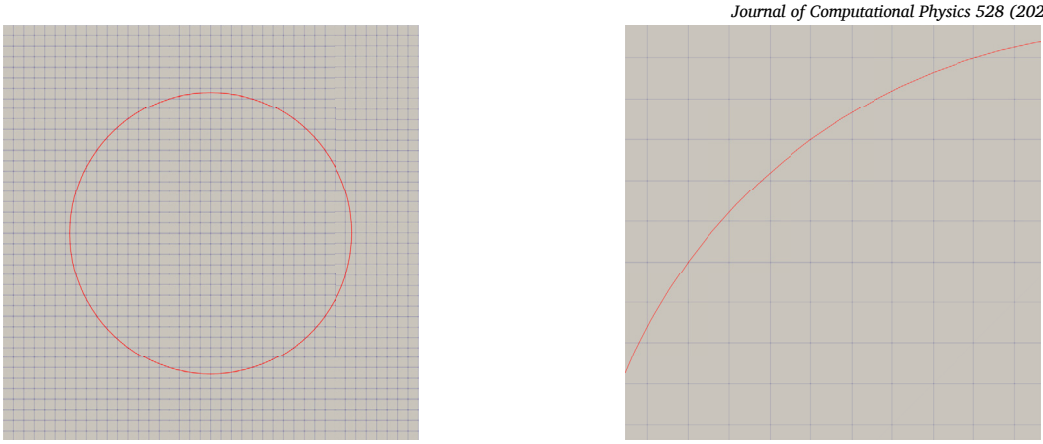
where $\hat{\boldsymbol{n}}$ is the unit normal vector to the interface ∂el; as customary for the discontinuous Galerkin methods, the test functions are taken equal to the basis functions. Finally, a summation is performed over all the elements in the mesh, and Eq. (3) becomes:

$$\sum_{el} \int_{el} \left[\phi_i J\boldsymbol{u}_t - \nabla_{\boldsymbol{\xi}} \phi_i \cdot (\boldsymbol{F}_e - \boldsymbol{F}_v) - \phi_i J\boldsymbol{S}\right] + \sum_{\partial el} \oint_{\partial el} \phi_i (\llbracket \boldsymbol{F}_e \rrbracket - \llbracket \boldsymbol{F}_v \rrbracket) = \boldsymbol{0}, \tag{4}$$

where $\llbracket * \rrbracket$ represents the jump of the fluxes (including normals) across the interface ∂el, introducing a coupling between el and the neighboring elements. Specifically, the jump across the interface is computed through the numerical fluxes.

The inviscid numerical flux $\llbracket \boldsymbol{F}_e \rrbracket = \boldsymbol{F}_e^\star$ can be computed using a upwind approach $\boldsymbol{F}_e^\star \to \boldsymbol{F}_e^\star(\boldsymbol{u}^+, \boldsymbol{u}^-, \hat{\boldsymbol{n}})$ and is a function of the state of the internal interface $\boldsymbol{u}^+$ and the state of the interface of the element sharing the face with el, $\boldsymbol{u}^-$, see Fig. 1. Similarly, the viscous numerical flux $\llbracket \boldsymbol{F}_v \rrbracket = \boldsymbol{F}_v^\star$ is a function of both $\boldsymbol{u}^+$, $\boldsymbol{u}^-$ and the gradient of conservative variables $(\nabla_{\boldsymbol{\xi}}\boldsymbol{u})^+$, $(\nabla_{\boldsymbol{\xi}}\boldsymbol{u})^-$: $\boldsymbol{F}_v^\star \to \boldsymbol{F}_v^\star(\boldsymbol{u}^+, \boldsymbol{u}^-, (\nabla_{\boldsymbol{\xi}}\boldsymbol{u})^+, (\nabla_{\boldsymbol{\xi}}\boldsymbol{u})^-, \hat{\boldsymbol{n}})$. In this study, we employ the Lax-Friedrichs flux for inviscid terms and use the Bassi-Rebay 1 (BR1) [42], the Interior-Penalty (IP) [43], and local discontinuous Galerkin (LDG) [44] fluxes to discretize viscous terms. The particular selection of fluxes is detailed for each test case.

In discontinuous Galerkin methods, boundary conditions on walls (for body-fitted meshes) are imposed weakly through the numerical fluxes. Therefore, no explicit imposition of boundary conditions appears in the formulation. In the case of continuous finite



(a) Embedded surface.

(b) Close up of a portion of the surface.

Fig. 2. Circle embedded in a simple Cartesian mesh; Fig. 2b shows a close up of the surface. The computational domain is the exterior of the circle.

element methods, a technique to weakly impose boundary condition needs to be chosen, such as Nitsche's method. In our case, we assume a ghost state for $\boldsymbol{u}^-$, see Fig. 1. When the element *el* lies on the physical surface, $\partial\Omega$, the interface state $\boldsymbol{u}^-$ and the interface gradient $(\nabla_{\boldsymbol{\xi}}\boldsymbol{u})^-$ are obtained from the boundary conditions of the problem considered: $\boldsymbol{u}^- = \boldsymbol{u}_b$ and $(\nabla_{\boldsymbol{\xi}}\boldsymbol{u})^- = (\nabla_{\boldsymbol{\xi}}\boldsymbol{u})_b$. To impose boundary conditions, an accurate representation of the body surface is essential to obtain the high-order accuracy of the DG. In fact, the geometry is described through the normal $\hat{\boldsymbol{n}}$ that appears in the interface integrals of Eq. (3) and failing to account for the boundary curvature can compromise the accuracy. As a consequence, a curvilinear mesh is required to properly describe the curved surface of the body. Concerning the mesh generation process, while creating linear meshes has become proficient even for complex shapes, the same cannot be said for curvilinear meshes. Therefore, there is considerable interest in employing high-order methods that do not require a high-order curved mesh. In the following section, we present a technique capable of achieving high-order precision with linear Cartesian meshes, bypassing the explicit curvature of the mesh. This is achieved by using an IBM as the one proposed next.

3. A high-order immersed boundary method

We aim to develop an IBM (see Fig. 2) capable of preserving the high-order accuracy of the underlying scheme, namely the discontinuous Galerkin method. The standard approach to developing IBMs involves modifying the equations; for instance, in the case of volume penalization, a source term is added to each degree of freedom located inside the body, which is treated as a porous medium with infinite permeability (see Appendix A and provided references). However, this method typically loses accuracy near the surface [22,23]. As an alternative, we propose to identify a set of faces (*shifted* faces) forming a closed piecewise line (2D) or a surface (3D). This will be the surrogate boundary, Γ_s, on which the state is properly modified. This boundary can be inside the flow domain, as for the SBM, or outside it. We are interested in flow problems and in boundaries defined by solid bodies, and we will consider Γ_s outside the flow domain and therefore inside the body that conforms the physical boundary; however the alternative of interior surrogate boundaries is also possible with the presented formulation. As we shall see, our approach implies an extrapolation of the unknowns, whereas the choice of interior Γ_s would amount to an interpolation.

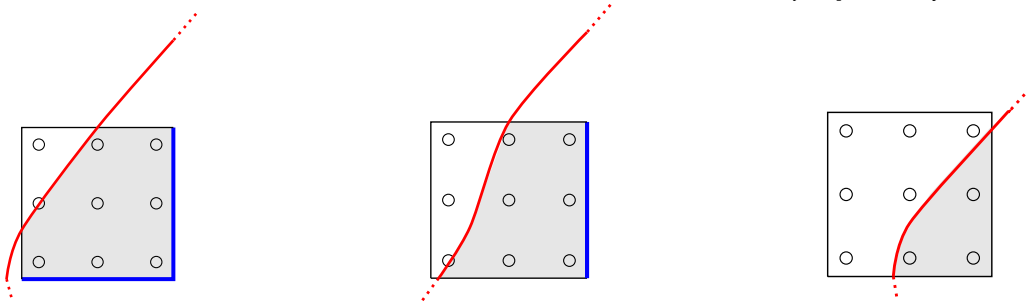
Specifically, the changed state is such that the boundary conditions are satisfied on the body surface and the numerical fluxes $\boldsymbol{F}_e^\star$ and $\boldsymbol{F}_v^\star$ (Eq. (4)) on the shifted faces are computed. In this way, the required state on these faces is not strongly enforced but is indirectly imposed through an interface flux, and thus weakly satisfied.

The identification of the shifted faces and the method of adjusting the state are explained in detail in the following section.

3.1. Identification of the shifted faces

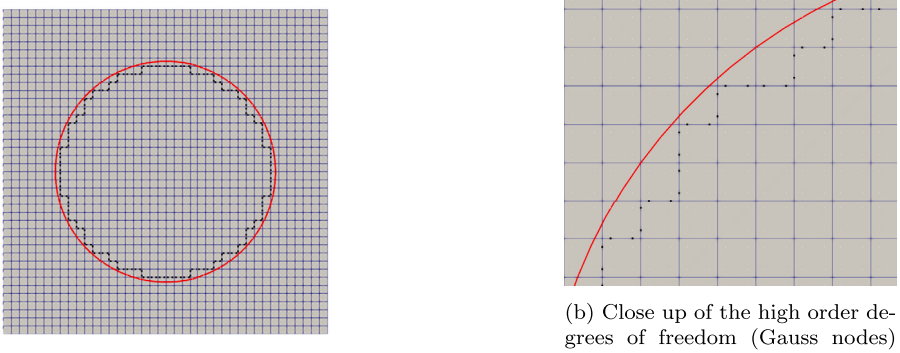
The set of shifted faces must satisfy two constraints. For the option of exterior surrogate boundary that we shall follow, the first is that the face has to be located entirely inside the body that conforms the boundary; the second is that the opposite face has to be partially outside the body. Fig. 3 shows examples of possible scenarios for 2D problems.

To identify the shifted faces, we use a ray-tracing technique [45], which is also used to compute the normal vectors to the points of the physical boundary; in particular, each degree of freedom belonging to a generic face is tagged when inside the body. Once the degrees of freedom are selected, each face that has all the degrees of freedom inside the body (outside the flow domain) is saved as a shifted face (if the second constraint mentioned above is also fulfilled). Two additional checks are required: if, following the procedure previously explained a face is tagged twice as a shifted face, it is discarded; the second check is done for all the elements having more than one shifted face and one (or at least one in 3D) face having one or more degrees of freedom in the fluid region. For what concerns the latter point, all the shifted faces belonging to the element are discarded, whereas the only one that is stored is the one in front of the face whose degrees of freedom are all or in part inside the fluid domain (see Appendix C).



(a) Two shifted faces. (b) One shifted face. (c) None shifted face.

Fig. 3. Different cases for shifted faces (in blue). The solid body (shaded gray area) is assumed to be on the right of the body boundary (red). The shifted faces are inside the solid, and therefore in the exterior of the flow domain where the problem is solved. The state of the shifted face is corrected. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)



(a) The black line is the close set of shifted faces.

(b) Close up of the high order degrees of freedom (Gauss nodes) belonging to the set of shifted faces.

Fig. 4. Shifted faces and degrees of freedom lying on these faces. The flow domain is the exterior of the circle.

A Cartesian mesh with an immersed boundary representing a circular cylinder is depicted in Fig. 4. The figure details the degrees of freedom belonging to the shifted faces shown in black. The state on each of these degrees of freedom is modified to take into account the presence of the embedded body.

3.2. Computation of the shifted faces state

Once the set of shifted faces has been selected, the next critical step is to compute the new state to be imposed to weakly satisfy the boundary conditions on the body surface. Each of the shifted faces can be seen as an interface separating an element fully inside the body (and therefore outside the flow domain) and an element partially inside the body, which is now called a *shifted* element. Moreover, they are characterized by the usual interface states $\boldsymbol{u}^+$ and $\boldsymbol{u}^-$ (see Fig. 1): $\boldsymbol{u}^+$ is the state coming from the shifted element (Fig. 5) while $\boldsymbol{u}^-$ is the state of the neighbor element, fully inside the body. Specifically, the latter state $\boldsymbol{u}^-$ is the one addressed by the proposed methodology and is properly modified so that the solution on the surface is the required one, *i.e.,* $\boldsymbol{u}_b$. The newly computed state $\boldsymbol{u}^- = \boldsymbol{u}_{sb}$ (the subscript $(.)_{sb}$ stands for shifted boundary) is then replaced in the numerical flux definition to obtain $\boldsymbol{F}_e^\star(\boldsymbol{u}^+, \boldsymbol{u}^-, \hat{\boldsymbol{n}}) \rightarrow \boldsymbol{F}_e^\star(\boldsymbol{u}^+, \boldsymbol{u}_{sb}, \hat{\boldsymbol{n}})$. For continuous finite element interpolations, the value of $\boldsymbol{u}_{sb}$ is what would be weakly prescribed on the surrogate boundary. In general, it is important to underline the fact that $\boldsymbol{u}_{sb} \neq \boldsymbol{u}_b$, see Fig. 5.

We now focus on the computation of the state $\boldsymbol{u}_{sb}$ to be imposed on each of the selected faces, which is the critical point of the formulation. While in the SBM $\boldsymbol{u}_{sb}$ would be obtained from $\boldsymbol{u}_b$ using a Taylor expansion, *to compute the value* $\boldsymbol{u}_{sb}$*, we reconstruct the solution on the shifted face by using a one-dimensional Lagrangian interpolation of order* P *along the body normal* $\hat{\boldsymbol{n}}_s$, see Fig. 7. Note that, in general, $\hat{\boldsymbol{n}}_s$ is different from the normal of the reference element $\hat{\boldsymbol{n}}$. We compute $\hat{\boldsymbol{n}}_s$ as exterior to the body, i.e., pointing inwards the computational domain.

The P^{th}-order Lagrangian polynomial needs the identification of $P+1$ points (*donor stencil*) on which the solution is known. Fig. 6 shows how these nodes are selected along a ray from a point on the physical boundary with a known normal. The geometrical coordinates of the generic j^{th}-node are:

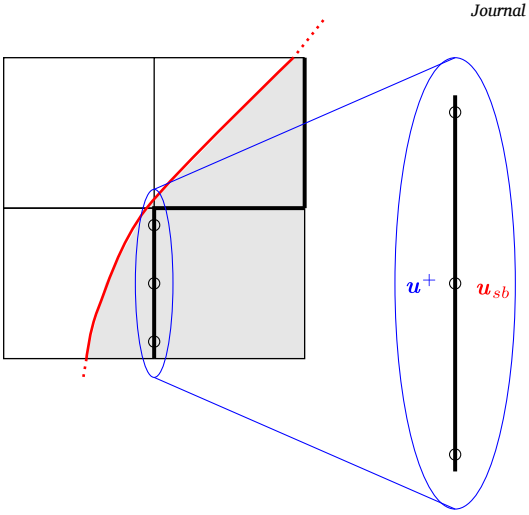


Fig. 5. Mesh region surrounding the immersed boundary. The red line represents a generic body (immersed boundary). The gray part is the interior of the body and shifted faces are highlighted in black. The zoomed image shows the states $\boldsymbol{u}^+$ and $\boldsymbol{u}_{sb}$ on one of the shifted faces. For completeness, Gauss nodes on the face (for $P = 2$) are also shown.

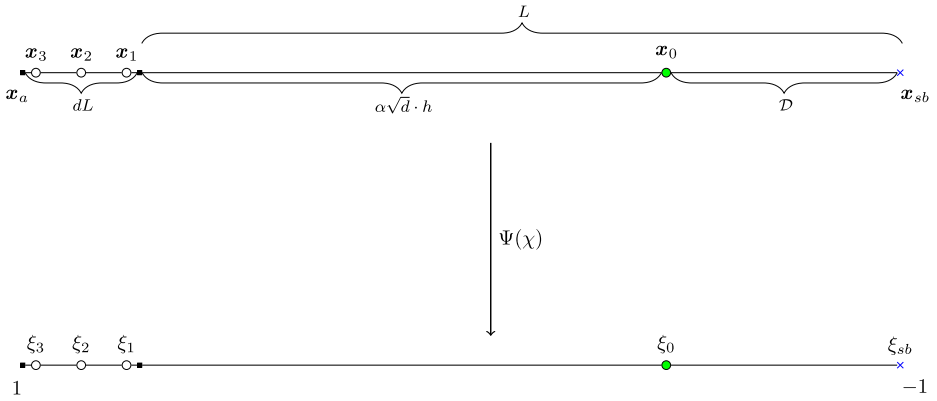


Fig. 6. The stencil points are shown along with the coordinates in the local reference frame. The local reference frame is such that for $\xi = -1$, $x = x_b$ and for $\xi = 1$, $x = x_a = D + \alpha\sqrt{d}\cdot h + dL$. It is important to note that nodes ξ_1, ξ_2 and ξ_3 are chosen so that, on the reference length dL, they correspond to the Gauss nodes ξ_1^G, ξ_2^G and ξ_3^G. The point ξ_0 arising from the intersection between the body boundary and the normal $\hat{n}$ is pictured in green.

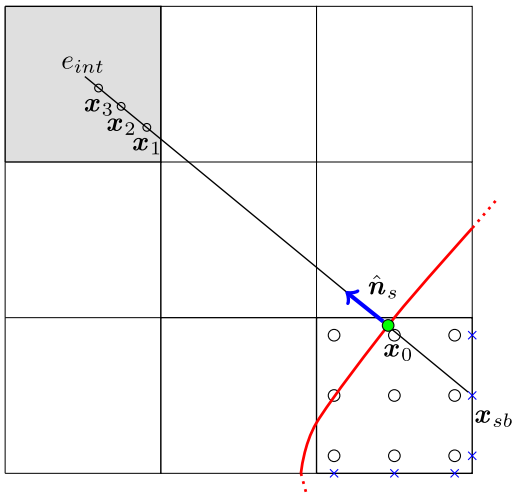


Fig. 7. The interpolation element e_{int} is shaded. The stencil is the same as the one of Fig. 6.

$$\begin{cases} \boldsymbol{x}_j = \boldsymbol{x}_{sb} + \chi_j \hat{\boldsymbol{n}}_s \\ \chi_j = L + \frac{(1+\xi_j^G)}{2} dL, \quad \text{with} \quad j = 1, \dots, P \\ \chi_0 = \mathcal{D}, \end{cases} \tag{5}$$

where ξ_j^G are the Gauss nodes distributed on dL. We defined the segment dL as $|\xi_0^G - \xi_1^G| \cdot h$, where h is the characteristic size of the mesh and L is the reference length. The boundary coordinates are obtained for $\chi_0 = \mathcal{D}$, with $\mathcal{D}$ the distance between the body surface and the point on the shifted boundary $\boldsymbol{x}_{sb}$.

The donor stencil must be far enough from the surface to avoid interdependencies. After some tests, we figured out experimentally that a good trade off for the definition of the reference length L seems to be:

$$L = \mathcal{D} + \alpha \sqrt{d} \cdot h, \tag{6}$$

where we recall that d is the dimension of the problem and α is a user-defined parameter. Finally, when selecting the value of the parameter α, different choices are possible; however, we found that a solid choice is to set $\alpha = 1 + \frac{\mathcal{D}}{h}$ (for more details, see Appendix D).

Having fixed one point on the body surface, we are left with P additional points to perform the interpolation at the shifted boundary given by Eq. (5). The $(P+1)^{th}$-point corresponds to the boundary point $\boldsymbol{x}_b = \boldsymbol{x}_{sb} + \mathcal{D}\hat{\boldsymbol{n}}_s$ where the solution is known.

Once the stencil has been computed, we perform a mapping from the physical space along the normal $\hat{\boldsymbol{n}}_s$ (defined by the scalar χ in Eq. (5)) to a local reference frame: $\Psi(\chi) \to \xi$, with $\xi \in [-1, 1]$. A simple linear map satisfying $\Psi(0) = -1$, $\Psi(L + dL) = 1$ is used:

$$\Psi(\chi) = 2\frac{\chi}{L + dL} - 1. \tag{7}$$

From the map, one can get the local reference frame coordinates for each point:

$$\xi_j = 2\frac{\chi_j}{(L + dL)} - 1 \quad \text{with} \quad j = 0, \dots, P. \tag{8}$$

Fig. 6 shows the mapping including the original stencil in the physical space and the one obtained after the application of the map. The Gauss nodes ξ_j^G for an approximation of $P = 3$ are shown together with all the quantities defined in Eq. (5). Once all geometric quantities are computed, the state is evaluated at each interpolation point by simply evaluating the polynomial representation of the solution at those locations. Recall that the state of the first point ξ_0, coming from the intersection between the body normal $\hat{\boldsymbol{n}}_s$ and the body surface, is known and comes from the boundary conditions. A single element, e_{int}, can be used to compute the state at each interpolation point even if the stencil does not belong to that specific element; Fig. 7 shows the stencil and the element e_{int} used to reconstruct the data.

In the local coordinate ξ, the interpolation constructed can be written as

$$\boldsymbol{u}(\xi) = \sum_{i=0}^{P} l_i(\xi)\boldsymbol{u}_i, \tag{9}$$

$l_i(\xi)$ being the interpolation functions. The known values of this interpolation are:

$$\boldsymbol{u}(\xi_0) = \boldsymbol{u}_b, \quad \boldsymbol{u}(\xi_i) = \boldsymbol{u}_i, \; i = 1, \dots, P,$$

whereas the boundary conditions to be prescribed on the surrogate boundary are:

$$\boldsymbol{u}_{sb} = \boldsymbol{u}(-1) = \sum_{i=0}^{P} l_i(-1)\boldsymbol{u}_i.$$

Expression (9) is the key of the proposed method. It allows one to obtain $\boldsymbol{u}_{sb}$ in terms of $\boldsymbol{u}_b$ (or of the derivatives of $\boldsymbol{u}$, see below) and of the values of the donor stencil $\boldsymbol{u}_i$, $i = 1, \dots, P$. Since these are unknown, they will contribute to the stiffness matrix of the problem, thus making it not symmetric; indeed for each DoF belonging to the shifted faces, Eq. (9) will contribute to the Jacobian, and once the system of equations is solved, the values on the shifted boundaries are updated. However, this is irrelevant if the equations are solved using an explicit time stepping, as mentioned earlier. The effect is similar to that of the SBM, in which the Taylor expansion used to obtain $\boldsymbol{u}_{sb}$ involves the derivatives of $\boldsymbol{u}$, which need to be expressed in terms of the nodal values of $\boldsymbol{u}$ itself and contribute to the stiffness matrix, when it is assembled.

An important remark is that instead of knowing $\boldsymbol{u}_b$ we may know its normal derivative, i.e., the derivative in the direction of ξ. This also allows us to compute $\boldsymbol{u}_b$ using Eq. (9) and, from this, obtain again $\boldsymbol{u}_{sb}$. This also opens the door to prescribe Neumann boundary conditions. In the following subsection, we explain how to apply this procedure to Poisson's problem.

We next detail the implementation of boundary conditions for the Poisson equation and for the Navier-Stokes equations. Our approach enables us to assign a predetermined value to the state vector, $\boldsymbol{u}$, on the body surface, $\partial\Omega$, expressed as $\boldsymbol{u}|_{\partial\Omega} = \boldsymbol{u}_b$.

3.3. Dirichlet boundary condition for the Poisson equation

Let us consider a generic Poisson equation:

$$-\nabla_{\boldsymbol{x}}^2 u = S, \quad \boldsymbol{x} \in \Omega, \tag{10}$$

with Dirichlet boundary conditions:

$$u = u_b \quad \text{on} \quad \partial\Omega, \tag{11}$$

where u is now a scalar function and S is a source term.

We can reconstruct u along the normal direction using the one-dimensional Lagrangian polynomial outlined in the preceding section, given in Eq. (9).

Note that the interpolation points are the points inside the fluid, *i.e.,* $i = 1, \dots, P$, and the point on the body corresponds to $i = 0$. Since the first interpolation point ($i = 0$) lies on the body, its state is known:

$$u_0 = u_b, \tag{12}$$

while u_i, with $i > 0$, are the nodal values at the points of the donor cell described in the previous section. The value on the shifted boundary can now be found as

$$u_{sb} = u|_{\xi=-1} = \underbrace{l_0(-1)u_b}_{\text{Dirichlet B.C.}} + \underbrace{\sum_{i=1}^{P} l_i(-1)u_i}_{\text{Fluid evaluation}}. \tag{13}$$

3.4. Neumann boundary condition for the Poisson equations

The Neumann boundary condition can be written as:

$$\frac{\partial u}{\partial \hat{\boldsymbol{n}}_s} = G, \tag{14}$$

with G given. Using the chain rules for the derivative, the left-hand side of Eq. (14) we get:

$$\frac{\partial u}{\partial \hat{\boldsymbol{n}}_s} = \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial \hat{\boldsymbol{n}}_s}; \tag{15}$$

in particular, keeping in mind that the reconstruction takes place along $\hat{\boldsymbol{n}}_s$ through a one-dimensional stencil and considering Eq. (8) one obtains:

$$\frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial \hat{\boldsymbol{n}}_s} = \frac{\partial u}{\partial \xi}\frac{\mathrm{d}\xi}{\mathrm{d}\chi} = \frac{\partial u}{\partial \xi}\frac{2}{(L + dL)}. \tag{16}$$

Finally, combining Eq. (14) and Eq. (16) and discretizing the solution using the one-dimensional Lagrangian polynomial as it was done for the Dirichlet boundary conditions, we can write:

$$\sum_{i=0}^{P} l_i'(\xi)u_i = \frac{(L + dL)}{2}G, \tag{17}$$

where $l_i'(\xi)$ is the derivative of the Lagrangian polynomial. It is now possible to compute the value u_0 so that Eq. (17) is satisfied, obtaining:

$$u_0 = \frac{\frac{(L+dL)}{2}G - \sum_{i=1}^{P} l_i'(\xi_0)u_i}{l_0'(\xi_0)}. \tag{18}$$

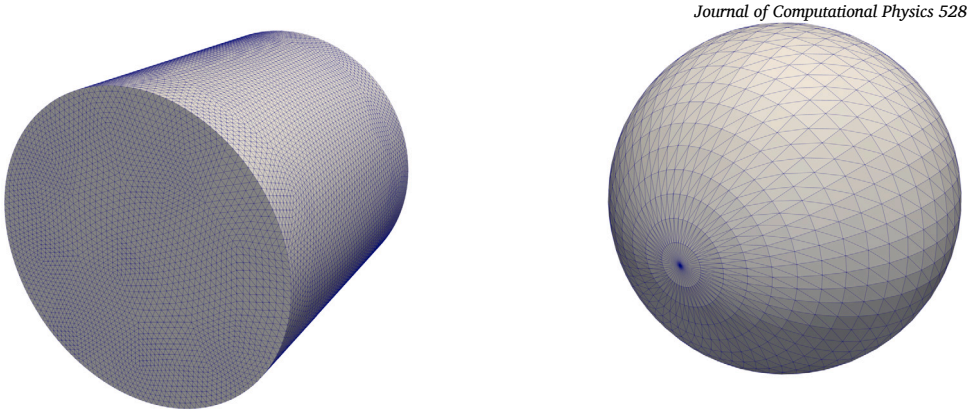
Once the state at u_0 is known, we proceed as computing the value on the shifted boundary:

$$u_{sb} = u|_{\xi=-1} = \underbrace{l_0(-1)u_0}_{\text{Neumann B.C.}} + \underbrace{\sum_{i=1}^{P} l_i(-1)u_i}_{\text{Fluid evaluation}}. \tag{19}$$

Note that the reconstruction of the unknown can be based on either knowing it or its normal derivative on the physical boundary, but in both cases u_{sb} is weakly prescribed on the surrogate boundary.

3.5. Surface representation

The discrete representation of the body surface has a crucial role in the framework of high-order methods. This is also true for our methodology, where the position of the body surface and its normal $\hat{\boldsymbol{n}}_s$, are required. When an analytical description of the shape is available, the exact normal body is $\hat{\boldsymbol{n}}_s$. More generally, in real application the analytical formula is not accessible and thus the geometry and normals are provided by a CAD software. In this work, we use the STL format for the geometry, and we calculate normals based on this geometry inside our solver. An STL is a file made of unstructured linear triangular elements (Fig. 8) and can be provided by most of the CAD software available today. This choice is due to the fact that STL are widely supported by most of the



(a) Cylinder. (b) Sphere.

Fig. 8. Examples of STL files.

software, are light in terms of size, and are open source. The drawback of STLs is that the triangulation is made of linear elements, and hence the curvature might not be accurate enough, but in these cases one can increase the number of triangles of the STL.

4. Test cases

The aim of this section is to verify the accuracy of the method. We will check the accuracy of the method for a range of problems: a 1D advection-diffusion equation, a Poisson problem including cut elements and curved elements, and finally a circular cylinder driven by the compressible Navier-Stokes equations.

The first test is a 1D linear convection-diffusion equation, which shows the capability of retrieving high-order accuracy even when taking into account the boundaries. Additionally, it is compared with the volume penalization IBM to show the superiority of the new method. The second case is a fully 2D Poisson equation solved on a squared boundary where no curvature is present. In particular, we show what happens when the shifted faces coincide with the physical boundary and when they do not leading to cut-cells. To check boundary conditions on curved boundaries, a third test case is proposed in which the steady-state heat equation is solved on two concentric circles. The last case compares our method, the body-fitted and volume penalization for the compressible Navier-Stokes. In all test cases, the Runge-Kutta (RK3) marching scheme is used to advance in time, even when dealing with steady problems. In the first test cases where the convergence is studied, the geometry is analytical and the body normals are analytically computed; on the other hand, in the Navier-Stokes case, a STL file is used.

4.1. 1D linear convection-diffusion

We first test our methodology using a 1D linear advection-diffusion equation, inspired from [22]:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0 \quad x \in [0,1], \tag{20}$$

where $c = 0.1$ is the advection speed and $\nu = 0.01$ is the diffusivity. Homogeneous boundary conditions are applied: $u(0,t) = 0$ and $u(1,t) = 0$. The time step used for the simulations is set to $\Delta t = 1.0 \cdot 10^{-7}$ to ensure time accuracy. The analytical solution of Eq. (20) is

$$u(x,t) = \mathrm{e}^{5x - t(\nu\pi^2 + 25\nu)} \sin(\pi x). \tag{21}$$

From Eq. (21) one can easily obtain the initial condition by setting $t = 0$.

To make this test suitable for the IBM, two additional elements are added at the extreme of the computational domain. In the classical volume penalization, a source term is added to Eq. (20) to simulate the presence of the boundaries. The source term is applied to all the degrees of freedom lying outside the computational domain, *i.e.,* all the points such that $x < 0$ and $x > 1$, see Fig. 9. Eq. (20) becomes:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = -\frac{1}{\eta}\left(u - u_s\right), \tag{22}$$

where η is the penalization term applied outside (0,1) that, in this simulation, is set to $\eta = \Delta t$ (the time step size) and u_s is the solution we want to impose on the boundaries, in this case $u_s = 0$. The final simulation time is $t_{max} = 0.01$. Fig. 10 shows the results obtained for traditional volume penalization (Fig. 10a) and those coming from the high-order approach (Fig. 10b), when discretizing the domain with N_{el} elements. As expected [22], the volume penalization method loses the high order accuracy near boundaries, while for our new proposed method the high order rate of convergence $\mathcal{O}(N_{el}^{-(P+1)})$ is maintained for all polynomial orders.

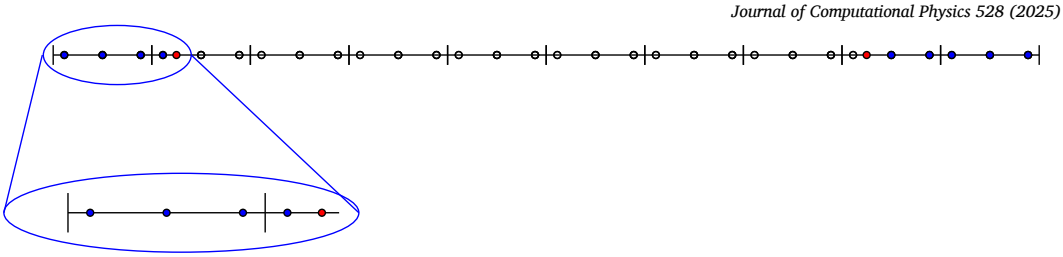
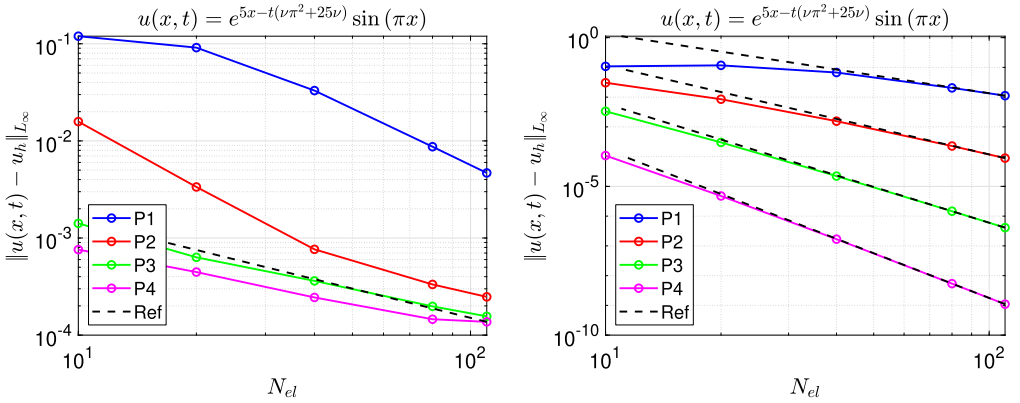


Fig. 9. One dimensional advection-diffusion: The mesh includes high order Gauss nodes (in black). The red nodes are the extreme of the domain, *i.e.,* 0 and 1. A close up on the element on the left show the blue nodes where the penalization is applied.



(a) Convergence with volume penalization.

(b) Convergence with our method.

Fig. 10. Convergence plots, N_{el} is the number of elements used for discretizing the one dimensional domain.

4.2. 2D Poisson equation with Dirichlet boundary conditions

The second test case is the two dimensional Poisson equation solved on a squared domain Ω:

$$\nabla^2 u(x,y) = S(x,y) \quad (x,y) \in [0,1]^2. \tag{23}$$

To find an analytical solution, a source term is added to the equation, in this case $S(x,y) = -2\pi^2[\sin(\pi x)\sin(\pi y)]$. The solution of Eq. (23) is

$$u(x,y) = \sin(\pi x)\sin(\pi y), \tag{24}$$

with Dirichlet boundary conditions on the squared geometry:

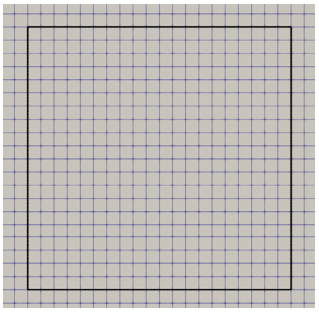
$$u(0,y) = u(1,y) = u(x,0) = u(x,1) = 0,$$

where $x, y \in [0,1]$. The solution is obtained using the RK3 time-marching scheme until a residual of $\mathcal{O}(10^{-9})$ is reached.

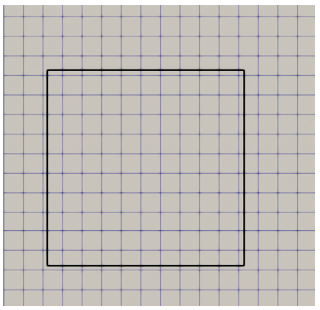
We perform two simulations since we want to check what happens when the boundaries coincide with the shifted faces and when there are cut elements. Fig. 11 shows the grids used for the two simulations. The black lines form the physical boundary that generates the squared domain $[0,1]^2$ where the solution is computed. In Fig. 12 the accuracy of the method is shown for different polynomial order on different h-meshes. Regarding the first test case (Fig. 12a), the proposed high-order method provides high-order accuracy with slopes $P+1$, which is the expected high-order accuracy. The results for the second case, which includes cut-cells, are reported in Fig. 12b. It can be seen that high-order accuracy is also obtained for all cases ($P+1$ slopes), although for a fixed mesh size h the errors are higher, as it usually happens in IBMs. This is a crucial validation, since it is the most common situation that can occur in a simulation where the STL arbitrarily lies on a background mesh. The developed method does not show instabilities and can recover the high-order accuracy of the underlying scheme.

4.3. 2D Poisson equation with Neumann boundary conditions

This test case is an extension of the previous one, where the main difference is the imposition of Neumann boundary condition on the right side of the domain, *i.e.* for $x = 1$. Taking this into account, the boundary conditions for the 2D Poisson problem can be recast as follows:

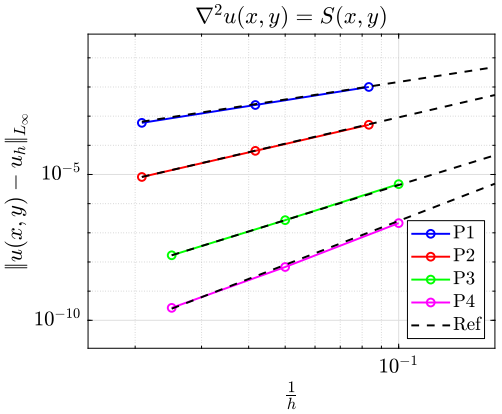


(a) Faces and boundaries coincident.

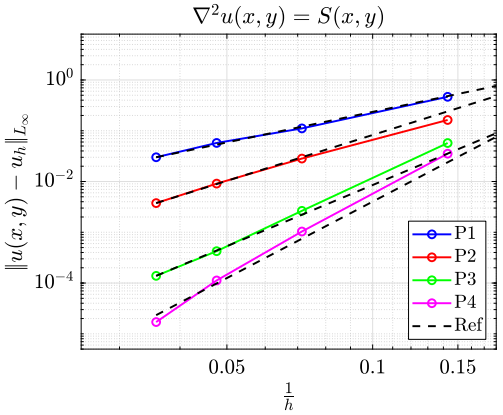


(b) Faces and boundaries not coincident.

Fig. 11. Poisson equation: The immersed boundary conforms with the background Cartesian mesh, Fig. 11a, and non-conforming meshes (with cut-cells), Fig. 11b.



(a) Convergence with faces and boundaries coincident.



(b) Convergence with faces and boundaries not coincident.

Fig. 12. Poisson equation: Convergence of the method for conforming faces and cut-cells with $P = 1$ to 4.

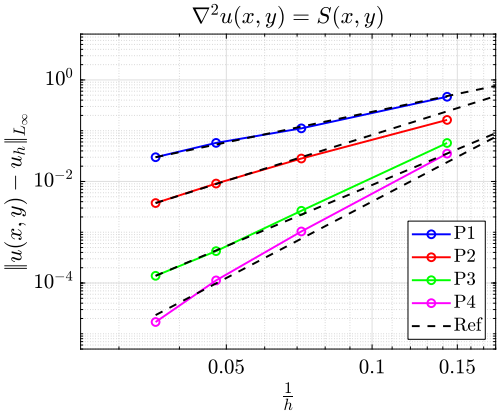
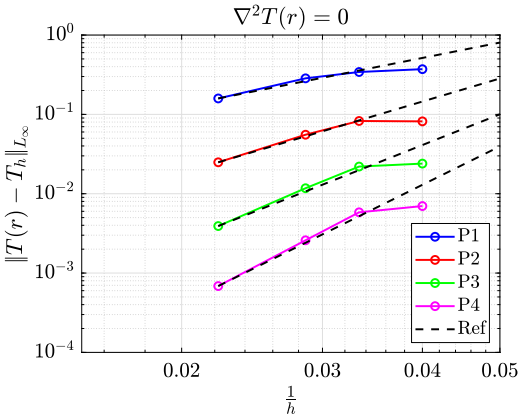


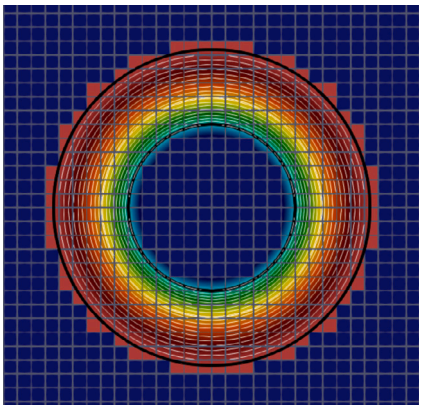
Fig. 13. Convergence with faces and boundaries not coincident when Neumann boundary conditions are imposed.

$$\begin{cases} u(0,y) = u(x,0) = u(x,1) = 0 \\ \frac{\partial u}{\partial \hat{\boldsymbol{n}}_s}|_{x=1} = \pi \sin(\pi y). \end{cases} \tag{25}$$

The results obtained for this simulation are reported in Fig. 13 and they very similar to those where only Dirichlet boundary conditions are imposed (see Fig. 12b); in both cases the high-order convergence of the underlying scheme is retrieved.



(a) Convergence of the heat equation.



(b) Temperature contours.

Fig. 14. Poisson equation with curved boundaries: Convergence of the method for the two cases with $P = 1$ to 4 along with the temperature contours for the steady heat equation.

4.4. Steady heat equation and curved boundaries

In this section, a steady heat equation is solved in the region occupied between two concentric circles whose radii are R_{max} and R_{min}. No source term is considered, and the boundary conditions are taken independently of the azimuthal angle. Since it is a symmetrical problem it can be expressed in polar coordinates as:

$$\nabla^2 T(r) = 0. \tag{26}$$

The solution of Eq. (26) is:

$$T(r) = A \ln\left(\frac{r}{R_{min}}\right) + B. \tag{27}$$

The constants A and B are computed using the Dirichlet boundary conditions that specify the temperature imposed on each circle, *i.e.,* $T(R_{max}) = T_{max}$ and $T(R_{min}) = T_{min}$; substituting these values in Eq. (27), we obtain the following values for A and B:

$$\begin{aligned} A &= \frac{(T_{max} - T_{min})}{\ln\left(\frac{R_{max}}{R_{min}}\right)}, \\ B &= T_{min}. \end{aligned} \tag{28}$$

For our test case, we set $T_{max} = 10$ and $T_{min} = 5$, while $R_{max} = 0.38$ and $R_{min} = 0.2$. As in the previous test case, a RK3 time marching scheme is used until a residual of $\mathcal{O}(10^{-9})$ is reached. Fig. 14a shows the convergence for polynomial orders $P = 1$ to 4 with different mesh sizes. We retrieve the formal-order accuracy $(P + 1)$ even when curving the physical boundaries. This highlights the capability of the method to obtain high-order convergence with curved boundaries with an underlying linear Cartesian mesh, avoiding the need for complex mesh generation. In addition, note that no particular treatment is needed for badly cut elements. Fig. 14b shows the temperature contours. Obviously, the solution outside the domain of interest can be neglected: if an element lies completely outside the region where the solution is computed, it can be discarded from the simulation, reducing the number of elements involved in the simulation process and, eventually, reducing the overall computational cost.

4.5. Navier-Stokes equations

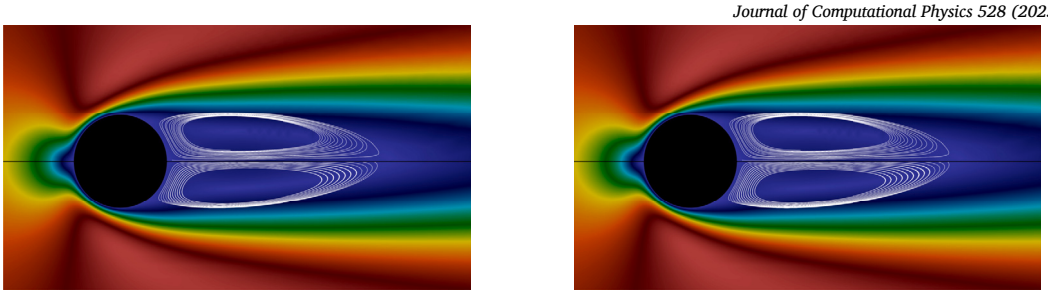
The compressible Navier-Stokes equations can be written as follows:

$$\tilde{\boldsymbol{u}}_t + \nabla_{\boldsymbol{x}} \cdot \left(\tilde{\boldsymbol{F}}_e - \tilde{\boldsymbol{F}}_v\right) = \boldsymbol{0}, \quad \boldsymbol{x} \in \Omega, \quad t > 0, \tag{29}$$

where $\boldsymbol{u}$ denotes the conservative variables, *i.e.,* $\boldsymbol{u} = (\rho, \rho u, \rho v, \rho w, E^t)$ and $\tilde{\boldsymbol{F}}_e$, $\tilde{\boldsymbol{F}}_v$ denote the non linear Navier Stokes inviscid and viscous fluxes, respectively:

$$\tilde{\boldsymbol{F}}_e = \begin{pmatrix} \rho \boldsymbol{v} \\ \rho \boldsymbol{v} \otimes \boldsymbol{v} + p\mathbb{I} \\ E^t \boldsymbol{v} \end{pmatrix}, \tag{30}$$

where $\boldsymbol{v} = (u, v, w)^T$ is the velocity field, p is the pressure (not to be confused with the polynomial order) and $E^t = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(\boldsymbol{v} \cdot \boldsymbol{v})$, being γ the heat capacity ratio and



(a) BF (top) vs VP-IBM (bottom). (b) BF (top) vs HO-IBM (bottom).

Fig. 15. Circular Cylinder at Re = 40: Comparison between body-fitted (BF), volume penalization (VP-IBM) and high-order IBM (HO-IBM) for the velocity for a $P = 3$ simulation. In both figures, the upper one is the body-fitted solution, whereas the immersed boundary is shown below.

$$\tilde{\boldsymbol{F}}_v = \begin{pmatrix} 0 \\ \boldsymbol{\tau} \\ \boldsymbol{v} \cdot \boldsymbol{\tau} - \lambda \nabla_{\boldsymbol{x}} T \end{pmatrix}. \tag{31}$$

The components of the shear stress tensor are $\tau_{ij} = \mu(\frac{\partial \boldsymbol{v}_i}{\partial x_j} + \frac{\partial \boldsymbol{v}_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial \boldsymbol{v}_k}{\partial x_k})$, λ is the thermal conductivity, and T is the temperature.

When dealing with the Navier-Stokes equations, the imposition of boundary conditions is not straightforward. Indeed, in this case, the required state is:

$$\boldsymbol{u}_{sb} = \begin{pmatrix} \rho_{sb} \\ (\rho u)_{sb} \\ (\rho v)_{sb} \\ (\rho w)_{sb} \\ E^t_{sb} \end{pmatrix}. \tag{32}$$

On the surface of the body, to impose no-slip boundaries, we require $u_b = v_b = w_b = 0$. Additionally, if we assume an adiabatic wall, we need to impose $(\nabla_{\boldsymbol{x}} T)_b \cdot \hat{\boldsymbol{n}}_s = 0$. The latter condition translates into the imposition of a specific temperature on the shifted boundary. However, to fully describe the state, we set the shifted boundary density equal to that coming from the internal element interface *i.e.,* $\rho_{sb} = \rho^+$ (i.e., extrapolation from the interior of the domain). Once all thermodynamic variables have been computed, it is possible to obtain the total energy as $E^t_{sb} = \rho^+ \left(\frac{T_{sb}}{\gamma(\gamma-1)M^2} + \frac{1}{2}\left(u^2_{sb} + v^2_{sb} + w^2_{sb}\right)\right)$, where M is the Mach number, allowing to compute the full state $\boldsymbol{u}_{sb}$. In particular, the final shifted boundary state is:

$$\boldsymbol{u}_{sb} = \begin{pmatrix} \rho^+ \\ \rho^+ u_{sb} \\ \rho^+ v_{sb} \\ \rho^+ w_{sb} \\ \rho^+ \left(\frac{T_{sb}}{\gamma(\gamma-1)M^2} + \frac{1}{2}\left(u^2_{sb} + v^2_{sb} + w^2_{sb}\right)\right) \end{pmatrix}. \tag{33}$$

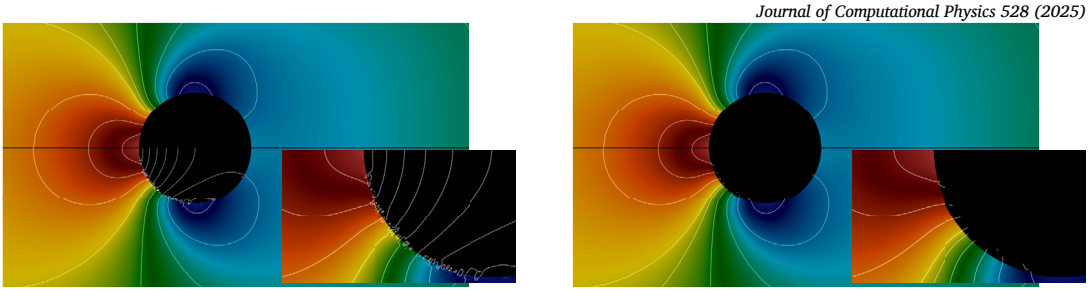
In what follows, we consider two and three-dimensional cases to validate our methodology when solving the Navier-Stokes equations.

4.5.1. *Two dimensional circular cylinder*

We first consider the flow around a circular cylinder at Reynolds 40 and Mach 0.3. Since no analytical solution is available for this case, the reference solution used is a body-fitted simulation with polynomial order $P = 3$. Moreover, we simulate the cylinder using a volume penalization approach (using the same high-order solver Horses3D) to compare the results. Fig. 15 shows the contours of the velocity field. When comparing the two IBMs (lower portion of Fig. 15) we see that both methods are able to capture the laminar recirculation bubble forming behind the cylinder. To quantify this observation, the geometric parameters of the bubble are reported in Table 1. It can be seen that both the volume penalization and the high-order method provide accurate results.

The main difference between volume penalization and the proposed high-order method arises when considering the pressure distribution on the body. To point out this discrepancy, the pressure contours are shown in Fig. 16. Fig. 16a shows one of the main known problems of volume penalization, which is the presence of spurious oscillations close to the surface. In addition, a close-up of the pressure contours is reported in Fig. 17 so that the two IBMs can be compared. Note that the proposed high-order method does not show spurious oscillations.

When interpolating/projecting the pressure on the STL surface, the resulting pressure (here the pressure coefficient C_p) is affected by these oscillations. Fig. 18 shows the pressure coefficient for the body fitted, the volume penalization, and the high-order method. The close-up of Fig. 18b shows the oscillatory behavior of the volume penalization, while the novel method shows no oscillations and values that are closer to the body-fitted reference. Finally, the pressure distribution over the body surface is required for the calculation of aerodynamics forces (lift, drag), which will be affected by these local errors. In the case of volume penalization, the presence of spurious oscillations prevents the correct reconstruction of the surface state, which is performed through an inverse distant



(a) BF (top) vs VP-IBM (bottom). (b) BF (top) vs HO-IBM (bottom).

Fig. 16. Circular Cylinder at Re = 40: Comparison between body-fitted (BF), volume penalization (VP-IBM) and high-order immersed boundary (HO-IBM) for a $P = 3$ simulation. In both figures, the upper one is the body-fit solution, whereas the immersed boundary is shown below, and a close up of the pressure contour lines is provided for both VP-IBM and HO-IBM.

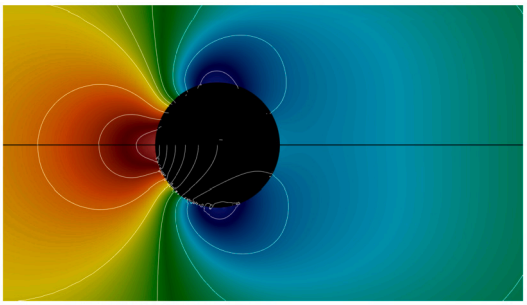
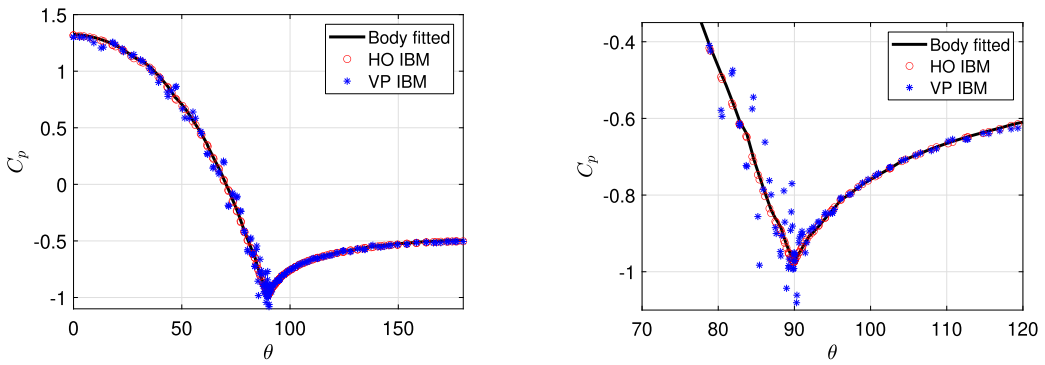


Fig. 17. Circular Cylinder at Re = 40: Close-up of the pressure contours showing the difference between high order and volume penalization for the pressure. The high-order immersed boundary is in the top half and the volume penalization in the bottom half of the figure.



(a) Coefficient of pressure. (b) Coefficient of pressure close up.

Fig. 18. Circular Cylinder at Re = 40: Pressure coefficient comparison between high-order immersed boundary, body-fitted and volume penalization.

weight interpolation (see [22]). This reconstruction technique has two main disadvantages, first, it is a first-order reconstruction, and second the final result depends on the number of interpolation points used. In the new proposed approach, the absence of oscillation allows one to reconstruct the solution on the STL using a high-order Lagrangian polynomial interpolation (as was done for the shifted boundary interpolation), giving a unique solution on the surface that does not depend on user-defined parameters. Table 1 compares the drag coefficients (c_D) for the two techniques analyzed, the reference body fitted together with the published data. The proposed high-order approach obtains results that are closer to the body-fitted reference (the relative error in drag decreases from 3.3% to 0.7%) and also close to published data.

4.5.2. *Three dimensional sphere*

The final test case is a three dimensional sphere at Reynolds 100 and Mach number 0.4. We compare again the high-order immersed boundary with the standard volume penalization with the aim of pointing out the main differences between the two. Specifically, in Fig. 19 the velocity streamlines obtained with both methodologies are compared. The same considerations as the two dimensional case can be done, *i.e.* the results are very similar since the volume penalization directly affects the momentum equations and both volume penalization and the high-order method provide accurate results.

Table 1
Circular Cylinder at Re = 40: Comparison between the reattachment length $\frac{L}{D}$, separation angle θ_r and the drag coefficient c_D.

	$\frac{L}{D}$	θ_r	c_D
Horses3D Body-fitted	2.29	53.6	1.53
Horses3D IBM VP	2.28	53.5	1.48
Horses3D High-order IBM	2.28	53.7	1.52
Dennis & Chang [46]	2.35	53.8	1.52
Fornberg [47]	2.24	55.6	1.50
Choi et al. [48]	2.21	53.6	1.49

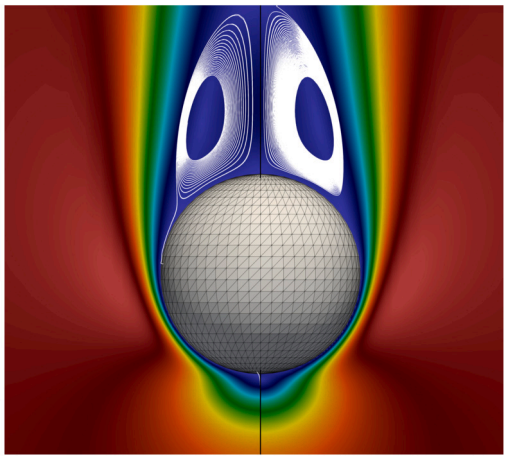
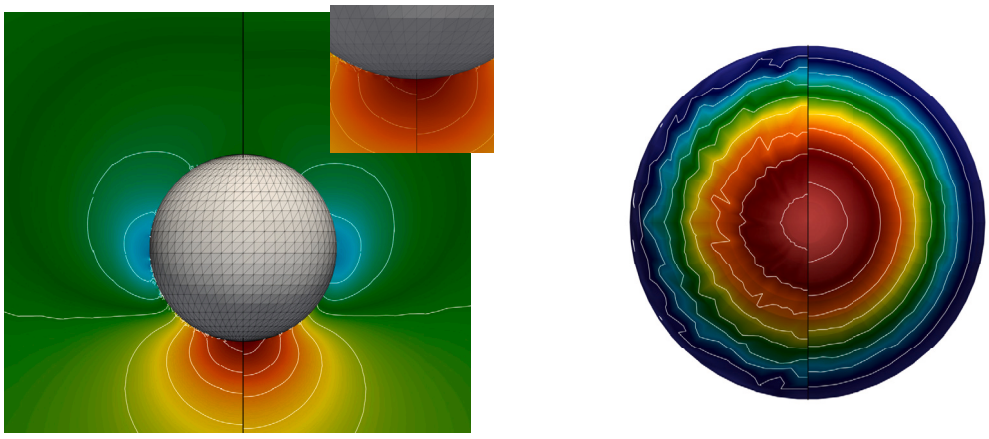


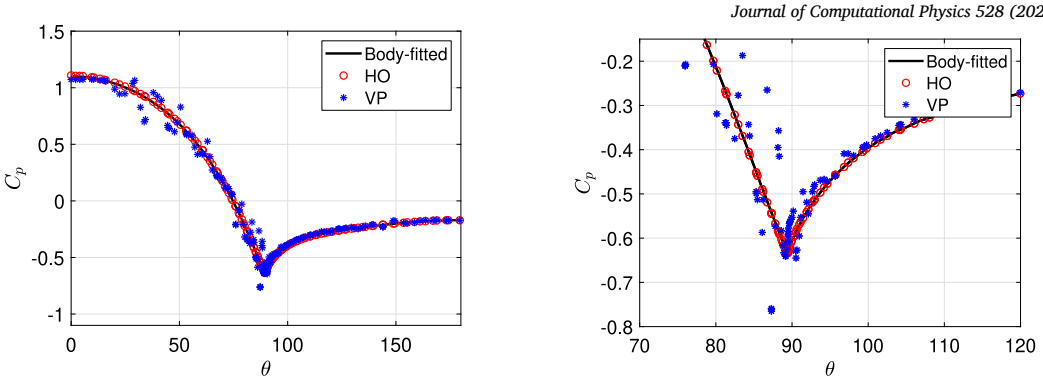
Fig. 19. Sphere at Re = 100: Comparison between velocity streamlines. The standard volume penalization is on the left side, the high-order is on the right side of the figure.



(a) Pressure contours. (b) Pressure contours on the sphere.

Fig. 20. Sphere at Re = 100: Comparison between pressure contours. The standard volume penalization is in the left side, the high-order is in the right side of the figure. A close up near the surface is also provided and the pressure contours on the body are plotted.

The most significant difference is, again, the pressure distribution close to the body and, consequently, the pressure coefficient C_p and the drag c_D. In particular, Fig. 20 highlights this difference between the two analyzed methods and reports the spurious pressure oscillation characteristic of the volume penalization approach, while no oscillations appear in the proposed high-order method. For completeness, the difference in the pressure distribution is shown through the pressure contours lines in Fig. 20b. The pressure oscillations observed previously affect the pressure coefficient C_p computed on the surface of the body, which is depicted in Fig. 21, as a function of the azimuth angle θ along with a close-up in Fig. 21b.



(a) Coefficient of pressure.

(b) Coefficient of pressure close up.

Fig. 21. Sphere at Re = 100: Pressure coefficient comparison between high-order immersed boundary and volume penalization.

Table 2
Sphere at Re = 100: Comparison between the drag coefficient c_D.

Source	c_D
Horses3D Body-fitted	1.087
Horses3D IBM VP	1.045
Horses3D High-order IBM	1.081
Fadlun et al. [13]	1.0794
Fornberg [49]	1.0852

Finally, Table 2 summarizes the drag coefficients of the volume penalization and the new methodology together with published data. The drag predicted with the new methodology is closer to the reference values, than the one obtained with the volume penalization approach.

5. Conclusions

We have proposed a IBM capable of preserving high-order accuracy, particularly in the treatment of boundary conditions. Our approach maintains the formal order of the scheme $(P + 1)$ when using polynomials of order P, avoiding the degradation near geometries observed in other unfitted boundary methods (e.g., volume penalization). By using Lagrangian polynomials along the body normal to reconstruct the state on specific faces, the method efficiently exploits Cartesian conforming meshes to simulate curved geometries. Implemented within a discontinuous Galerkin framework, using the open source Horses3D solver, this strategy has been tested on the convection-diffusion equation, the Poisson equation, and the compressible Navier-Stokes equations. These tests confirmed the method's ability to achieve high-order accuracy near the body, even in the presence of cut elements including curved surfaces. Furthermore, our study demonstrated that the proposed method outperforms volume penalization for the compressible Navier-Stokes equations, as evidenced by the improved pressure distribution on the body and more accurate drag coefficients. In conclusion, this work presents a truly high-order IBM. Future research will focus on applying this methodology to more complex flow regimes, including turbulence and moving geometries, to further validate and extend its applicability.

CRediT authorship contribution statement

S. Colombo: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **G. Rubio:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Funding acquisition, Formal analysis. **J. Kou:** Writing – review & editing, Writing – original draft, Supervision. **E. Valero:** Project administration, Funding acquisition. **R. Codina:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Formal analysis. **E. Ferrer:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Stefano Colombo reports financial support was provided by European Union. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

SC acknowledges the funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska Curie grant agreement No 955923-SSECOID. RC gratefully acknowledges the support received from the ICREA Acadèmia Program, from the Catalan Government. EF would like to thank the support of Agencia Estatal de Investigación (for the grant "Europa Excelencia 2022" Proyecto EUR2022-134041/AEI/10.13039/501100011033) y del Mecanismo de Recuperación y Resiliencia de la Unión Europea. EF and GR acknowledge the funding received by the Grant DeepCFD (Project No. PID2022-137899OB-I00) funded by MCIN/ AEI/10.13039/501100011033 and by ERDF A way of making Europe. This research has received funding from the European Union (ROSAS, project number 101138319). This research has received funding from the European Union (ERC, Off-coustics, project number 101086075). Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. Finally, all authors gratefully acknowledge the Universidad Politécnica de Madrid (www.upm.es) for providing computing resources on Magerit Supercomputer and the computer resources at MareNostrum and the technical support port provided by Barcelona Supercomputing Center (RES-IM-2024-1-0003).

Appendix A. Volume penalization immersed boundary

A brief explanation of the volume penalization immersed boundary is here proposed for clarity.

The volume penalization IBM consists in imposing a source term to the Navier-Stokes equations to simulate the presence of a body. As is common in this method, a mask $\zeta(\boldsymbol{x},t)$ is introduced in order to distinguish the region of the domain Ω inside the body Ω_b from the one outside Ω_f, with $\Omega = \Omega_b \cup \Omega_f$:

$$\zeta(\boldsymbol{x},t) = \begin{cases} 1, & \text{if } \boldsymbol{x} \in \Omega_b \\ 0 & \text{if } \boldsymbol{x} \in \Omega_f. \end{cases} \tag{A.1}$$

The mask is computed using a ray-tracing approach that, according to the number of intersections, allows one to check if a degree of freedom is inside the body (odd number of intersections) or outside (even number of intersections).

The Navier-Stokes equations, with the addition of the source term, become:

$$\tilde{\boldsymbol{u}}_t + \nabla_{\boldsymbol{x}} \cdot \left(\tilde{\boldsymbol{F}}_e - \tilde{\boldsymbol{F}}_v\right) = \tilde{\boldsymbol{S}}, \quad \boldsymbol{x} \in \Omega, \quad t > 0. \tag{A.2}$$

For the volume penalization, the source term for Dirichlet boundary conditions is:

$$\tilde{\boldsymbol{S}} = \frac{\zeta}{\eta} \begin{pmatrix} 0 \\ \rho u_s - \rho u \\ \rho v_s - \rho v \\ \rho w_s - \rho w \\ \frac{\rho}{2}(u_s^2 + v_s^2 + w_s^2) - \frac{\rho}{2}(u^2 + v^2 + w^2) \end{pmatrix}, \tag{A.3}$$

where $\boldsymbol{v}_s = (u_s, v_s, w_s)^T$ is the velocity to be imposed on the body. For no-slip boundary conditions, $\boldsymbol{v}_s = (0,0,0)^T$; η is the penalization parameter. The penalization parameter should be high enough to properly simulate the porosity of the body. However, this increases the stiffness of the problem and reduces the time step of the simulation. The common practice in the immersed boundary community is to set the penalization parameter equal to the explicit time step of the scheme, *i.e.,* $\eta = \Delta t$. More details can be found in [23], [22], [50], [51], among others.

Appendix B. Bassi-Rebay 1

As it is common practice for the Bassi-Rebay 1 formulation (BR1) [52], an additional variable is added to the Navier-Stokes equations, $\boldsymbol{g} = \nabla_{\boldsymbol{x}} \boldsymbol{u}$, leading to a coupled system of equations:

$$\boldsymbol{J}\boldsymbol{g} - \nabla_{\boldsymbol{\xi}} \boldsymbol{u} = \boldsymbol{0} \tag{B.1}$$

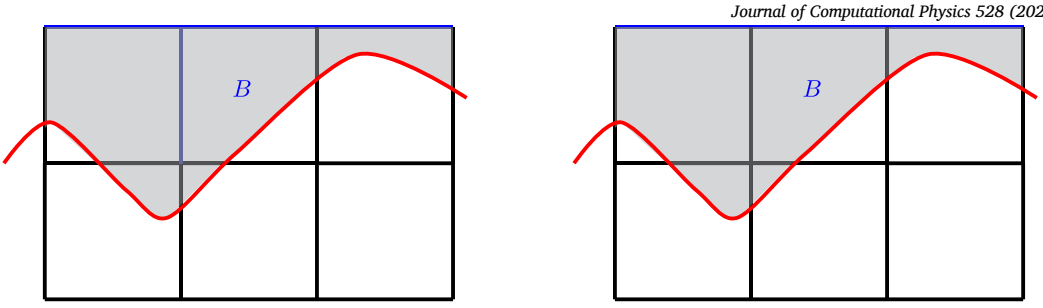
$$\boldsymbol{J}\boldsymbol{u}_t + \nabla_{\boldsymbol{\xi}} \cdot (\boldsymbol{F}_e - \boldsymbol{F}_v) - \boldsymbol{J}\boldsymbol{S} = \boldsymbol{0}. \tag{B.2}$$

Considering the weak formulation of Eq. (B.1), a jump in the state $\boldsymbol{u}$ across the interface between two elements appears:

$$\sum_{el} \int_{el} [\phi_i \boldsymbol{J}\boldsymbol{g} + \nabla_{\boldsymbol{\xi}} \phi \cdot \boldsymbol{u}] - \sum_{\partial el} \oint_{\partial el} \phi_i [\![\boldsymbol{u}]\!] \cdot \hat{\boldsymbol{n}} = \boldsymbol{0}. \tag{B.3}$$

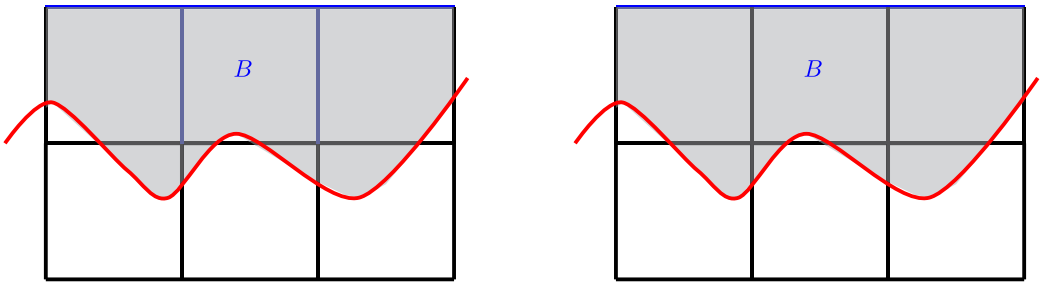
Note that, when transformed back to the physical domain, a factor $1/h$ appears in the last term. The jump appearing in Eq. (B.3) is defined, for the BR1 case, as:

$$[\![\boldsymbol{u}]\!] = \frac{1}{2}\left(\boldsymbol{u}^+ - \boldsymbol{u}^-\right). \tag{B.4}$$



(a) Without check. (b) After check.

Fig. 22. First case of interest in the evaluation of the shifted faces; the shifted faces are depicted in blue.



(a) Without check. (b) After check.

Fig. 23. Second case of interest in the evaluation of the shifted faces; the shifted faces are depicted in blue.

When a shifted face is considered, the jump appearing in Eq. (B.4) is modified and replaced by the computed shifted state $\boldsymbol{u}_{sb}$:

$$[\![\boldsymbol{u}]\!]_{sb} = \frac{1}{2}\left(\boldsymbol{u}_{sb} - \boldsymbol{u}^{-}\right). \tag{B.5}$$

Consequently, the gradient $\boldsymbol{g} = \frac{1}{J}\nabla_{\xi}\boldsymbol{u}$ is evaluated taking into account the presence of the body and is used to compute the viscous numerical fluxes $\boldsymbol{F}_{v}^{\star}(\boldsymbol{u}^{+},\boldsymbol{u}^{-},\boldsymbol{g}^{+},\boldsymbol{g}^{-},\hat{\boldsymbol{n}})$; in particular, when considering a shifted face, we set $\boldsymbol{g}_{sb} = \boldsymbol{g}^{-}$, obtaining $\boldsymbol{F}_{v}^{\star}(\boldsymbol{u}_{sb},\boldsymbol{u}^{-},\boldsymbol{g}_{sb},\boldsymbol{g}^{-},\hat{\boldsymbol{n}})$. Note that similar approaches can be followed for other viscous fluxes, such as interior penalty or Bassi-Rebay 2.

Appendix C. Some consideration about the identification of the shifted faces

In order to clarify some aspects about the identification of the shifted faces, some interesting cases are here reported and analyzed. In particular, we provide some examples of the two possible conditions that can occur during the process of selection of the shifted faces, *i.e.* 1 - when a face is tagged twice as a shifted face and 2 - when one element has one face with at least one degree of freedom inside the fluid domain (see section 3.1).

An example of the first situation is reported in Fig. 22. We focus on the left face of element $\boldsymbol{B}$ highlighted in blue in Fig. 22a. Following the procedure reported in section 3.1, this face is tagged twice as a shifted face and, therefore it is discarded, leading to the final shifted boundary Γ_s is shown in Fig. 22b.

The other possible condition is reported in Fig. 23. Here, the right and left faces of element $\boldsymbol{B}$ are tagged as shifted faces (Fig. 23a) leading to a shifted boundary that is not closed. If this occurs, these two faces are deleted from Γ_s as they are not in front of the face that has one or more degrees of freedom in the fluid domain. The final boundary Γ_s is reported in Fig. 23b.

It is important to notice that these conditions arise when complex geometries are discretized using coarse meshes. Refining the mesh locally helps to alleviate the occurrence of these problems and also allows to better capture the shape of the body.

Appendix D. Effect of the variation of the parameter α

This section is dedicated to a description of the behavior of the proposed method as a function of the parameter α. In particular, we use as a reference the steady heat equation, which is described in detail in Section 4.4, and perform simulations for various parameters α, on a mesh with 25×25 elements. The results are shown in Fig. 24 where $\alpha = \epsilon\left(1 + \frac{D}{h}\right)\sqrt{d}\cdot h$.

Fig. 24 shows that the value of α slightly affects the error. When α decreases, the error reduces for the same polynomial order. Note that there is a limitation on the minimum achievable value of the parameter. In Fig. 24 the black diamonds represent the minimum value of α (α_m) below which the scheme does not converge: notably, the higher the polynomial order, the greater the minimum allowable value of α. Furthermore, as we reduce the value of α, the position of the stencil points gets closer to the surface. The use

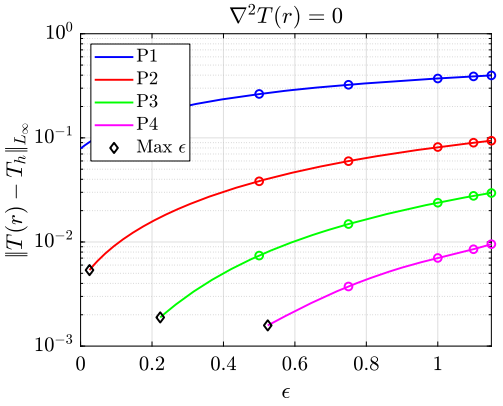


Fig. 24. Behavior of the error as α changes.

of a α_m is justified by the fact that a recursive dependency is expected as we move closer to the body, and this factor helps to recover stability and convergence. Regarding $P = 1$, no value of α_m is found, which suggests that the term $d\,L$ (see Section 3.2) is always large enough to avoid a recursive dependency. In fact, we can check that reducing the value of $d\,L$ leads to the same behavior as reported for higher polynomial orders. In this work, the value of α corresponds to $\epsilon = 1$, which is a good compromise in terms of error and safety margin from α_m, since the same can be used for all polynomials.

Data availability

Data will be made available on request.

References

[1] N. Chalmers, G. Agbaglah, M. Chrust, C. Mavriplis, A parallel *hp*-adaptive high order discontinuous Galerkin method for the incompressible Navier-Stokes equations, J. Comput. Phys. X 2 (2019) 100023, https://doi.org/10.1016/j.jcpx.2019.100023.
[2] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, M. Visbal, High-order CFD methods: current status and perspective, Int. J. Numer. Methods Fluids 72 (8) (2013) 811–845, https://doi.org/10.1002/fld.3767, https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.3767.
[3] G. Karniadakis, S. Sherwin, Spectral/hp Element Methods for Computational Fluid Dynamics, Oxford University Press, 2005, https://doi.org/10.1093/acprof:oso/9780198528692.001.0001.
[4] G. Aparicio-Estrems, A. Gargallo-Peiró, X. Roca, Defining metric-aware size-shape measures to validate and optimize curved high-order meshes, Comput. Aided Des. 168 (2024) 103667, https://doi.org/10.1016/j.cad.2023.103667.
[5] G. Aparicio-Estrems, A. Gargallo-Peiró, X. Roca, Combining high-order metric interpolation and geometry implicitization for curved r-adaption, Comput. Aided Des. 157 (2023) 103478, https://doi.org/10.1016/j.cad.2023.103478.
[6] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (2) (1972) 252–271, https://doi.org/10.1016/0021-9991(72)90065-4.
[7] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.
[8] F. Sotiropoulos, X. Yang, Immersed boundary methods for simulating fluid-structure interaction, Prog. Aerosp. Sci. 65 (2014) 1–21.
[9] B.E. Griffith, N.A. Patankar, Immersed methods for fluid–structure interaction, Annu. Rev. Fluid Mech. 52 (2020) 421–448.
[10] T. Ye, R. Mittal, H. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. 156 (2) (1999) 209–240.
[11] H. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, J. Comput. Phys. 174 (1) (2001) 345–380.
[12] P. Fu, G. Kreiss, S. Zahedi, A bound preserving cut discontinuous Galerkin method for one dimensional hyperbolic conservation laws, arXiv:2404.13936, https://arxiv.org/abs/2404.13936, 2024.
[13] E. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (1) (2000) 35–60.
[14] N. Zhang, Z. Zheng, An improved direct-forcing immersed-boundary method for finite difference applications, J. Comput. Phys. 221 (1) (2007) 250–268, https://doi.org/10.1016/j.jcp.2006.06.012.
[15] H. Luo, H. Dai, P.J.F. de Sousa, B. Yin, On the numerical oscillation of the direct-forcing immersed-boundary method for moving boundaries, Comput. Fluids 56 (2012) 61–76.
[16] 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.
[17] S. Majumdar, G. Iaccarino, P. Durbin, RANS solvers with adaptive structured boundary non-conforming grids, Annu. Res. Br. 1 (2001).
[18] Y.-H. Tseng, J.H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2) (2003) 593–623, https://doi.org/10.1016/j.jcp.2003.07.024.
[19] P. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, Numer. Math. 81 (4) (1999) 497–520.
[20] E. Brown-Dymkoski, N. Kasimov, O.V. Vasilyev, A characteristic based volume penalization method for general evolution problems applied to compressible viscous flows, J. Comput. Phys. 262 (2014) 344–357, https://doi.org/10.1016/j.jcp.2013.12.060.
[21] R. Abgrall, H. Beaugendre, C. Dobrzynski, An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques, J. Comput. Phys. 257 (2014) 83–101.
[22] J. Kou, S. Joshi, A.H. de Mendoza, K. Puri, C. Hirsch, E. Ferrer, Immersed boundary method for high-order flux reconstruction based on volume penalization, J. Comput. Phys. 448 (2022) 110721, https://doi.org/10.1016/j.jcp.2021.110721.

[23] J. Kou, E. Ferrer, A combined volume penalization/selective frequency damping approach for immersed boundary methods applied to high-order schemes, J. Comput. Phys. 472 (2023) 111678.
[24] J. Kou, A.H. de Mendoza, S. Joshi, S. Le Clainche, E. Ferrer, Eigensolution analysis of immersed boundary method based on volume penalization: applications to high-order schemes, J. Comput. Phys. 449 (2022) 110817, https://doi.org/10.1016/j.jcp.2021.110817.
[25] V.J. Llorente, J. Kou, E. Valero, E. Ferrer, A modified equation analysis for immersed boundary methods based on volume penalization: applications to linear advection–diffusion equations and high-order discontinuous Galerkin schemes, Comput. Fluids 257 (2023) 105869, https://doi.org/10.1016/j.compfluid.2023.105869.
[26] J. Kou, E. Ferrer, A combined volume penalization/selective frequency damping approach for immersed boundary methods: application to moving geometries, Phys. Fluids 35 (12) (2023) 121702, https://doi.org/10.1063/5.0179779.
[27] T. Engels, D. Kolomenskiy, K. Schneider, J. Sesterhenn, Numerical simulation of fluid–structure interaction with the volume penalization method, J. Comput. Phys. 281 (2015) 96–115, https://doi.org/10.1016/j.jcp.2014.10.005.
[28] A. Massing, M. Larson, A. Logg, M. Rognes, A Nitsche-based cut finite element method for a fluid–structure interaction problem, Commun. Appl. Math. Comput. Sci. 10 (11 2013), https://doi.org/10.2140/camcos.2015.10.97.
[29] B. Liu, A Nitsche stabilized finite element method: application for heat and mass transfer and fluid–structure interaction, Comput. Methods Appl. Mech. Eng. 386 (2021) 114101, https://doi.org/10.1016/j.cma.2021.114101.
[30] E. Burman, P. Hansbo, Fictitious domain finite element methods using cut elements: II. a stabilized Nitsche method, Appl. Numer. Math. 62 (2012) 328–341.
[31] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, J. Comput. Phys. 171 (1) (2001) 132–150, https://doi.org/10.1006/jcph.2001.6778.
[32] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2) (2005) 448–476, https://doi.org/10.1016/j.jcp.2005.03.017.
[33] F. Sotiropoulos, X. Yang, Immersed boundary methods for simulating fluid–structure interaction, Prog. Aerosp. Sci. 65 (01 2013), https://doi.org/10.1016/j.paerosci.2013.09.003.
[34] 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, https://doi.org/10.1016/j.jcp.2017.10.026.
[35] N.M. Atallah, C. Canuto, G. Scovazzi, Analysis of the shifted boundary method for the Poisson problem in general domains, arXiv:2006.00872, 2020.
[36] M. Funada, T. Imamura, High-order immersed boundary method for inviscid flows applied to flux reconstruction method on a hierarchical Cartesian grid, Comput. Fluids 265 (2023) 105986, https://doi.org/10.1016/j.compfluid.2023.105986.
[37] S. Péron, C. Benoit, T. Renaud, I. Mary, An immersed boundary method on Cartesian adaptive grids for the simulation of compressible flows around arbitrary geometries, Eng. Comput. 37 (07 2021), https://doi.org/10.1007/s00366-020-00950-y.
[38] B. Constant, S. Péron, H. Beaugendre, C. Benoit, An improved immersed boundary method for turbulent flow simulations on Cartesian grids, J. Comput. Phys. 435 (2021) 110240, https://doi.org/10.1016/j.jcp.2021.110240.
[39] R. Codina, J. Baiges, Approximate imposition of boundary conditions in immersed boundary methods, Int. J. Numer. Methods Eng. 80 (2009) 1379–1405.
[40] E. Ferrer, G. Rubio, G. Ntoukas, W. Laskowski, O. Mariño, S. Colombo, A. Mateo-Gabín, H. Marbona, F.M. de Lara, D. Huergo, J. Manzanero, A. Rueda-Ramírez, D. Kopriva, E. Valero, A high-order discontinuous Galerkin solver for flow simulations and multi-physics applications, Comput. Phys. Commun. 287 (2023) 108700, https://doi.org/10.1016/j.cpc.2023.108700.
[41] D.A. Kopriva, Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers, 1st edition, Springer Publishing Company, Incorporated, 2009.
[42] G.J. Gassner, A.R. Winters, F.J. Hindenlang, D.A. Kopriva, The br1 scheme is stable for the compressible Navier–Stokes equations, J. Sci. Comput. 77 (2018) 154–200.
[43] B. Riviere, Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation, vol. 35, 2008.
[44] B. Cockburn, G.E. Karniadakis, C.-W. Shu, Discontinuous Galerkin Methods: Theory, Computation and Applications, 1st edition, Springer Publishing Company, Incorporated, 2011.
[45] T. Akenine-Möller, E. Haines, N. Hoffman, A. Pesce, M. Iwanicki, S. Hillaire, Real-Time Rendering, 4th edition, A K Peters/CRC Press, Boca Raton, FL, USA, 2018.
[46] S. Dennis, G.-z. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100, J. Fluid Mech. 42 (3) (1970) 471–489, https://doi.org/10.1017/S0022112070001428, cited by: 777, https://www.scopus.com/inward/record.uri?eid=2-s2.0-0014938217&doi=10.1017%2fS0022112070001428&partnerID=40&md5=adcab63a521b38ba40ad46d52736529e.
[47] B. Fornberg, A numerical study of steady viscous flow past a circular cylinder, J. Fluid Mech. 98 (4) (1980) 819–855.
[48] J.-I. Choi, R.C. Oberoi, J.R. Edwards, J.A. Rosati, An immersed boundary method for complex incompressible flows, J. Comput. Phys. 224 (2) (2007) 757–784, https://doi.org/10.1016/j.jcp.2006.10.032.
[49] B. Fornberg, Steady viscous flow past a sphere at high Reynolds numbers, J. Fluid Mech. 190 (1988) 471–489, https://doi.org/10.1017/S0022112088001417.
[50] O. Boiron, G. Chiavassa, R. Donat, A high-resolution penalization method for large Mach number flows in the presence of obstacles, Comput. Fluids 38 (3) (2009) 703–714, https://doi.org/10.1016/j.compfluid.2008.07.003.
[51] P. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in viscous flows, Numer. Math. 81 (1999) 497–520, https://doi.org/10.1007/s002110050401.
[52] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, J. Comput. Phys. 131 (2) (1997) 267–279, https://doi.org/10.1006/jcph.1996.5572.