A Shifted Boundary Method for the Compressible Euler Equations

Xianyi Zeng[a], Ting Song[b], Guglielmo Scovazzi[c,*]

[a]*Department of Mathematics, Lehigh University, Bethlehem, 18015, PA, United States*
[b]*ExxonMobil Upstream Research Company, Spring, 77389, TX, United States*
[c]*Department of Civil and Environmental Engineering, Duke University, Durham, 27705, NC, United States*

Abstract

The Shifted Boundary Method (SBM) is applied to compressible Euler flows, with and without shock discontinuities. The SBM belongs to the class of unfitted (or immersed, or embedded) finite element methods and avoids integration over cut cells (and the associated implementation/stability issues) by reformulating the original boundary value problem over a surrogate (approximate) computational domain. Accuracy is maintained by modifying the original boundary conditions using Taylor expansions. Hence the name of the method, that shifts the location and values of the boundary conditions. We specifically discuss the advantages the proposed method offers in avoiding spurious numerical artifacts in two scenarios: (a) when curved boundaries are represented by body-fitted polygonal approximations and (b) when the Kutta condition needs to be imposed in immersed simulations of airfoils. An extensive suite of numerical tests is included.

Keywords: Shifted Boundary Method, explicit time integration, Euler equations, compressible flow, finite element method
2020 MSC: 65M85, 35L04, 65M60

1. Introduction

In this article, the Shifted Boundary Method (SBM) is applied to compressible Euler equations, first in the case of smooth flows problems without discontinuities and then in the case of flows with strong shocks. The SBM falls in the category of immersed (or embedded, or unfitted) finite element methods, in which the computational grids do not conform to the geometry of the shapes to be simulated. The idea of immersing shapes in computational grids can be beneficial when the geometrical complexity becomes a dominant factor in simulations.

Immersed methods for compressible flow have been widely used in the context of finite difference and finite volume methods, after the initial developments in the context of the incompressible Navier-Stokes equations [1–3]. We attempt to mention the most relevant contributions to this work, knowing that ours could only be an incomplete account: we focus almost exclusively on contributions on compressible flows, because of the immense literature on immersed/embedded methods.

Cartesian Cut-Cell finite volume methods [4–8] became popular in the late ’90s, as a way to attack very complex geometry computations of compressible and incompressible flows. At about the same time, so-called Discrete Forcing Methods [9] were proposed. In Discrete Forcing Methods, the immersed boundary is modeled through appropriate forcing terms that sharply enforce the embedded boundary conditions, as an alternative of the Immersed Boundary Method of Peskin [1]. Shortly after the Ghost-Cell Method was proposed [10, 11], in which boundary conditions are applied using fictitious cells in proximity of the embedded boundary. Parallel to these developments are Cartesian Embedded Methods [12–14], which share some similarity in the implementation. In a different thread of developments, Overset Grids Methods [15, 16] (akin the Chimera Method [17–19] implemented in the NASA OVERFLOW solver [20]) were extended to compressible flow problems. While most of the previous developments utilize Cartesian

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

or mapped Cartesian grids, the FIVER algorithm [21–23] is a prime example of embedded finite volume methods constructed on fully unstructured grids.

The literature is much scarcer in the context of finite element methods for compressible flows. The Implicit Boundary Method, a type of CutFEM method, was recently explored [24, 25]. CutFEM discretizations, although appeared earlier than the Implicit Boundary Method, have only recently been applied to systems of hyperbolic conservation laws, typically leveraging the Discontinuous Galerkin (DG) framework [26–28]. CutFEMs proposed to date rely on XFEM-type data structures to integrate the governing equations on cut cells, a strategy that produces a number of challenging situations. First, if particularly small cut cells are produced, cutFEMs may induce numerical instabilities that need stabilization or linear system of equations that have poor condition numbers. To obviate the occurrence of these numerical instabilities in unfitted finite element methods, Burman [29] introduced the *ghost penalty method*, in which the variational form is stabilized by introducing a penalization of the solution gradients at the interface separating cut and uncut elements. A different approach introduced for B-spline variational formulations, known as *Extended B-splines* [30, 31], involves eliminating via an extrapolation procedure cut B-splines with small support. This technique was applied to the Navier-Stokes equations for moving boundary problems in [32] and [33]. One additional challenge for cutFEM method is somewhat complex data structures and integration procedures on the cut cells, where the canonical finite element approaches need to be forfeited. Sub-triangulations and adaptive quadratures are often used [34, 35], leading to a non-negligible portion of the overall wall-clock time is spent in the treatment of the embedded/immersed boundary.

To obviate the problematic issues mentioned in the application of the cutFEM paradigm, the SBM eliminates cut cells altogether from the simulation. In the SBM, the location where boundary conditions are applied is *shifted* from the true to a surrogate boundary, and, at the same time, modified (*shifted*) boundary conditions are applied in order to avoid a reduction in the convergence rates of the overall formulation. In fact, if the boundary conditions associated to the true domain are not appropriately modified on the surrogate domain, only first-order convergence is to be expected. The appropriate (modified) boundary conditions are then applied weakly, using a Nitsche strategy. This process yields a method which is simple, robust, accurate and efficient.

The SBM was proposed for the Poisson and Stokes flow problems in [36] and was generalized in [37] to the advection-diffusion and Navier-Stokes equations. The SBM was extended to problems involving internal interfaces in [38], and the benefits of its application in the context of reduced order modeling was analyzed in [39–41]. Further rigorous mathematical analysis was pursued in [42, 43] for the Poisson and the Stokes flow problems, and in [42] for the treatment of general domains with corners. A high-order version of the SBM was presented in [44], together with its mathematical analysis, for the Poisson and Stokes flow problems. An arbitrary-order penalty-free version of the SBM for Poisson and linear elasticity problems appeared in [45]. The SBM was extended to complex-geometry problems in linear and nonlinear elasticity in [46, 47] and to Iso-Geometric Analysis (IGA) in [48].

For problems involving hyperbolic systems of equations, much less work has been done. A recent effort in this direction [49] extended the SBM to the equations of acoustics and shallow water flows. In this reference, the authors documented the handling of various types of wave structures at the embedded boundaries and tested the SBM for problems with relatively weak discontinuities.

As already mentioned, the present work is focused on the Euler equations of inviscid compressible flow, with particular emphasis on the interplay between the SBM and the hyperbolic wave structure near boundaries. For example, we numerically analyze for the first time how strong shocks interact with the geometrically complex features of the SBM approximate boundary.

The baseline numerical method we use is the stabilized finite element method (FEM) described in [50, 51], which relies on globally continuous, piecewise-linear basis functions. The possible presence of strong shocks is controlled by means of two of the most popular and recent discontinuity capturing approaches [52] for finite elements: the entropy viscosity proposed by Guermond et al. [53, 54] and a Lax-type viscosity equipped with a modified Ducros sensor [55, 56]. We compare the SBM results with benchmark conforming mesh computations (CMC, i.e., computations using body-fitted grids). An extensive suite of numerical examples shows that the SBM and CMC yield very similar results if the mesh resolution is comparable, whether the artificial viscosity is active or not. Similar results can be expected when problem-specific and more sophisticated discontinuity-capturing operators are used when flux limiting strategies are used. We want to stress that it is not the aim of this article to find the most accurate discontinuity-capturing operator to be combined with the SBM, but instead to show that the SBM performs with the same accuracy and robustness of a body-fitted FEM on grids of similar resolutions.

We also discuss additional advantages of the SBM framework, even when the grids are conforming (body-fitted). First of all, the SBM can improve the accuracy of lower-order body-fitted discretizations in the presence of curved boundaries. A polygonal interpolation of the boundary might induce spurious effects, but the SBM can easily cure those, by perturbing the boundary conditions and keeping into account the true geometry of boundaries. More details are offered in Section 3.5 and the numerical examples in Section 4.3 and Section 4.4. Second, the solution of inviscid compressible flows may lose uniqueness at a geometrical singularity, such as at the trailing edge of an airfoil. In conventional aerodynamics, one often needs to deliberately impose the Kutta condition [57] of flow continuity at the trailing edge, either implicitly, aligning body-fitted grids with the camber line, or explicitly, using additional numerical instruments [58]. In the SBM, instead, we implement the Kutta condition on arbitrary immersed grids by extending the camber line of an airfoil as a slip interface beyond the trailing edge (see Section 4.5 for more details).

This paper is organized as follows: Section 2 briefly reviews the governing equations and the underlying FEM for conforming mesh computations with weak enforcement of various boundary conditions; Section 3 presents the general framework of the SBM and the detailed derivation of shifted wall boundary conditions, as well as the use of the SBM to improve the accuracy of CMCs over curved geometries (SBM+CMC); Section 4 assesses the numerical performance of the proposed method, by comparing the SBM, the CMC, and the SBM+CMC for a number of challenging benchmark tests and, finally, Section 5 summarizes conclusions and future directions.

2. Governing equations, a baseline body-fitted variational formulation, and boundary conditions

2.1. Governing equations

Let Ω be a connected open set in $\mathbb{R}^d$ with Lipschitz boundary $\Gamma = \partial\Omega$, and let $\boldsymbol{n}$ denote the outer-pointing normal to Γ. The first-order form of the hyperbolic system of the compressible Euler equations reads:

$$\boldsymbol{U}_{,t} + \nabla \cdot (\boldsymbol{U} \otimes \boldsymbol{u}(\boldsymbol{U}) + \boldsymbol{G}(\boldsymbol{U})) + \boldsymbol{B} = \boldsymbol{0}, \tag{2.1}$$

where $\boldsymbol{U}(\boldsymbol{x},t) : \Omega \times \mathbb{R}^+ \mapsto \mathbb{R}^m$ is the vector of conserved variables, with $m = d+2$ and $\boldsymbol{a} \otimes \boldsymbol{b} = \boldsymbol{a}\boldsymbol{b}^T$, that is $(\boldsymbol{a} \otimes \boldsymbol{b})_{ij} = a_i b_j$. Here, $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{U}) \in \mathbb{R}^d$ denotes the advection velocity, $\boldsymbol{G} = \boldsymbol{G}(\boldsymbol{U}) \in \mathbb{R}^{m \times d}$ the non-advective part of the flux, and $\boldsymbol{B} = \boldsymbol{B}(\boldsymbol{x},t;\boldsymbol{U}) \in \mathbb{R}^m$ a source term. In particular, the divergence $\nabla \cdot$ of a matrix $\boldsymbol{C} \in \mathbb{R}^{m \times d}$ applies to the second index, that is $\nabla \cdot \boldsymbol{C} = \partial_j C_{ij}$. $\boldsymbol{U}$ and $\boldsymbol{G}$ are defined as

$$\boldsymbol{U} = \begin{Bmatrix} \rho \\ \rho \boldsymbol{u} \\ \rho E \end{Bmatrix} = \begin{Bmatrix} \rho \\ \rho u_1 \\ \vdots \\ \rho u_d \\ \rho E \end{Bmatrix}, \qquad \boldsymbol{G}(\boldsymbol{U}) = \begin{bmatrix} \boldsymbol{0}^T \\ p\boldsymbol{I}_d \\ p\boldsymbol{u}^T \end{bmatrix}, \tag{2.2}$$

where curly brackets indicate vectors and square brackets indicate matrices. We favor the vector notation to encompass both the two- and three-dimensional cases, that will both be explored in the numerical examples. More specifically, ρ is the density, $\boldsymbol{u} \in \mathbb{R}^d$ is the velocity, E is the specific total energy, $\boldsymbol{I}_d \in \mathbb{R}^{d \times d}$ is the identity matrix in dimension d, and $\boldsymbol{w}^T$ indicates the row-vector transpose of a column-vector $\boldsymbol{w} \in \mathbb{R}^d$.

The total energy is the sum of the internal and kinetic energy according to $E = e + \boldsymbol{u} \cdot \boldsymbol{u}/2$. The specific internal energy e, the density ρ, and the pressure p are related by the equation of state $p = p(\rho, e)$ of the fluid, assumed for simplicity to be an ideal gas:

$$p = (\gamma - 1)\rho e, \tag{2.3}$$

with γ the isentropic exponent. Equation (2.1) can be more compactly written as:

$$\boldsymbol{U}_t + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}) + \boldsymbol{B} = \boldsymbol{0}, \tag{2.4}$$

with $\boldsymbol{F}(\boldsymbol{U}) := \boldsymbol{U} \otimes \boldsymbol{u}(\boldsymbol{U}) + \boldsymbol{G}(\boldsymbol{U})$, but we prefer the former to the latter in this work.

Initial conditions at $t = 0$ can be specified for the solution vector field $\boldsymbol{U}$, in terms of an initial velocity $\boldsymbol{u}_0$, and two fields out of the triplet initial pressure p_0, density ρ_0, and internal energy e_0, which are connected by (2.3). Namely:

$$\boldsymbol{U}(\boldsymbol{x}, t = 0) = \boldsymbol{U}_0. \tag{2.5}$$

The discussion about boundary conditions, which is central to the present paper, is postponed until Section 2.3.

2.2. *Baseline body-fitted variational formulation*

As a point of departure in the discussion, we consider now a Galerkin formulation constructed on a computational grid fitted to all boundaries (body-fitted grid/mesh) of the domain Ω. Let $\mathscr{T}^h$ be an admissible and shape-regular tessellation of Ω, on which the following piecewise-linear, globally continuous, finite element trial space $\mathscr{S}^h$ and test space $\mathscr{V}^h$ are defined:

$$\mathscr{S}^h = \mathscr{V}^h = \{\boldsymbol{W}^h \in (\mathscr{C}(\overline{\Omega}))^m : \boldsymbol{W}^h_{|T} \in (\mathscr{P}^1(T))^m,\ \forall T \in \mathscr{T}^h\}, \tag{2.6}$$

where $\mathscr{P}^1(T)$ is the set of linear polynomials over the element T and $(\mathscr{C}(\overline{\Omega}))^m$ is the set of functions with values in $\mathbb{R}^m$ that are continuous over the closure of the set Ω. We choose the trial space $\mathscr{S}^h$ to coincide with the test space $\mathscr{V}^h$ since boundary conditions will be enforced weakly. In particular, we consider the following variational formulation of (2.2):

Find $\boldsymbol{U}^h : \mathbb{R}^+ \mapsto \mathscr{V}^h$ such that for all $t \in \mathbb{R}^+$ and $\boldsymbol{W}^h \in \mathscr{S}^h$:

$$\begin{aligned}(\boldsymbol{W}^h, \boldsymbol{U}^h_{,t})_\Omega - (\nabla \boldsymbol{W}^h, \boldsymbol{U}^h \otimes \boldsymbol{u}(\boldsymbol{U}^h) + \boldsymbol{G}(\boldsymbol{U}^h))_\Omega + (\boldsymbol{W}^h, \boldsymbol{B}^h)_\Omega + \mathscr{B}_{\mathrm{bc}}(\boldsymbol{W}^h, \boldsymbol{U}^h)&\\ + \mathscr{B}_{\mathrm{dc}}(\boldsymbol{W}^h, \boldsymbol{U}^h) + \mathscr{B}_{\mathrm{vms}}(\boldsymbol{W}^h, \boldsymbol{U}^h) \,=\, 0\,,&\end{aligned} \tag{2.7}$$

in which integration by parts has been applied to the flux term. The bilinear forms $\mathscr{B}_{\mathrm{bc}}$, $\mathscr{B}_{\mathrm{dc}}$, and $\mathscr{B}_{\mathrm{vms}}$ implement, respectively, the boundary conditions/contributions, a shock/discontinuity-capturing operator, and the variational multiscale stabilization (or VMS term, required for the numerical stabilization of equal-order interpolation approximations).

2.3. *Boundary conditions*

We consider next the imposition of various boundary conditions. Let the boundary be partitioned with the non-overlapping decomposition $\partial\Omega = \overline{\Gamma}_I \cup \overline{\Gamma}_O \cup \overline{\Gamma}_S \cup \overline{\Gamma}_F$, where the subscripts "*I*", "*O*", "*S*", and "*F*" stand for inlet, outlet, slip, and far-field boundaries, respectively. Furthermore, $\overline{\Gamma}_I = \overline{\Gamma}_{I;sub} \cup \overline{\Gamma}_{I;sup}$ and $\overline{\Gamma}_O = \overline{\Gamma}_{O;sub} \cup \overline{\Gamma}_{O;sup}$, where the subscript "*sub*" indicates that flow is subsonic (i.e., $||\boldsymbol{u}|| < c_s$ with $c_s = c_s(\boldsymbol{U}) = \sqrt{\gamma p/\rho}$ the isentropic speed of sound) while the subscript "*sup*" indicates that the flow is supersonic (i.e., $||\boldsymbol{u}|| > c_s$).

Far-field boundary conditions are introduced to prevent undesirable wave reflections at a distance from bodies that are immersed in an airflow, like airfoils. In the far field there may not be a fixed number of characteristics traversing the boundary and we handle this situation by introducing a numerical flux between the numerical solution and a prescribed ambient fluid state vector. This procedure is similar to what is usually done in finite volume methods [59] or residual redistribution methods [60].

Then, the boundary term $\mathscr{B}_{\mathrm{bc}}(\cdot,\cdot)$ can be decomposed as

$$\mathscr{B}_{\mathrm{bc}}(\cdot,\cdot) \,=\, \mathscr{B}_{I;sub}(\cdot,\cdot) + \mathscr{B}_{I;sup}(\cdot,\cdot) + \mathscr{B}_{O;sub}(\cdot,\cdot) + \mathscr{B}_{O;sup}(\cdot,\cdot) + \mathscr{B}_S(\cdot,\cdot) + \mathscr{B}_F(\cdot,\cdot)\,, \tag{2.8}$$

where each term is discussed below:

- At a supersonic inlet $\Gamma_{I;sup}$, all characteristics leave the domain and the full fluid state vector is specified, as

  $$\boldsymbol{U} = \boldsymbol{U}_b\,, \quad \forall \boldsymbol{x} \in \Gamma_{I;sup}\,, \tag{2.9}$$

  with the corresponding boundary term given by:

  $$\mathscr{B}_{I;sup}(\boldsymbol{W}^h, \boldsymbol{U}^h) = \langle \boldsymbol{W}^h, (\boldsymbol{U}_b \otimes \boldsymbol{u}(\boldsymbol{U}_b) + \boldsymbol{G}(\boldsymbol{U}_b))\boldsymbol{n}\rangle_{\Gamma_{I;sup}}\,. \tag{2.10}$$

- At a subsonic inlet $\Gamma_{I;sub}$, all but one characteristic exit the boundary, and consequently all but one of the associated Riemann invariants need to be specified. Thus, we impose the boundary condition

  $$\rho = \rho_b\,, \quad \boldsymbol{u} = \boldsymbol{u}_b\,, \quad \forall \boldsymbol{x} \in \Gamma_{I;sub}\,, \tag{2.11}$$

  with the corresponding boundary term given by

  $$\mathscr{B}_{I;sub}(\boldsymbol{W}^h, \boldsymbol{U}^h) = \left\langle \boldsymbol{W}^h \otimes \boldsymbol{n}\,, \left\{ \begin{array}{c} \rho_b \\ \rho_b\, \boldsymbol{u}_b \\ \frac{\gamma p(\boldsymbol{U}^h)}{\gamma - 1} + \frac{\rho_b\, \boldsymbol{u}_b \cdot \boldsymbol{u}_b}{2} \end{array} \right\} \otimes \boldsymbol{u}_b + \left\{ \begin{array}{c} \boldsymbol{0}^T \\ p(\boldsymbol{U}^h)\, \boldsymbol{I}_d \\ p(\boldsymbol{U}^h)\, \boldsymbol{u}_b^T \end{array} \right\} \right\rangle_{\Gamma_{I;sub}} \,. \tag{2.12}$$

- At a supersonic outlet $\Gamma_{O;sup}$, all characteristics exit the boundary and consequently no boundary data is required. The boundary term is simply a result of integration-by-part of the flux term

$$\mathscr{B}_{O;sup}(\boldsymbol{W}^h, \boldsymbol{U}^h) = \langle \boldsymbol{W}^h \otimes \boldsymbol{n}, \boldsymbol{U}^h \otimes \boldsymbol{u}(\boldsymbol{U}^h) + \boldsymbol{G}(\boldsymbol{U}^h) \rangle_{\Gamma_{O;sup}} . \tag{2.13}$$

- At a subsonic outlet $\Gamma_{O;sub}$, only one characteristic exits the boundary and a single Riemann invariant needs to be specified. Common choices involve imposing the pressure or the dynamic pressure (the latter is especially favored in many aerodynamics applications). Here we illustrate the idea by specifying

$$p = p_b , \quad \forall \boldsymbol{x} \in \Gamma_{O;sub} , \tag{2.14}$$

so that the boundary term now reads

$$\mathscr{B}_{O;sub}(\boldsymbol{W}^h, \boldsymbol{U}^h) = \left\langle \boldsymbol{W}^h \otimes \boldsymbol{n}, \left\{ \begin{matrix} \rho(\boldsymbol{U}^h) \\ \rho(\boldsymbol{U}^h)\,\boldsymbol{u}(\boldsymbol{U}^h) \\ \frac{\gamma p_b}{\gamma - 1} + \frac{\rho(\boldsymbol{U}^h)\,\boldsymbol{u}(\boldsymbol{U}^h)\cdot\boldsymbol{u}(\boldsymbol{U}^h)}{2} \end{matrix} \right\} \otimes \boldsymbol{u}(\boldsymbol{U}^h) + \left\{ \begin{matrix} \boldsymbol{0}^T \\ p_b \boldsymbol{I}_d \\ p_b\,\boldsymbol{u}(\boldsymbol{U}^h)^T \end{matrix} \right\} \right\rangle_{\Gamma_{O;sub}} . \tag{2.15}$$

- At a slip boundary Γ_S, the normal component of the velocity is specified:

$$\boldsymbol{u} \cdot \boldsymbol{n} = u_n , \quad \forall \boldsymbol{x} \in \Gamma_S . \tag{2.16}$$

Such boundary conditions include a fixed wall, a plane of symmetry with $u_n \equiv 0$, or a moving boundary, either rigid or prescribed by structural motion in the context of fluid-structure interaction problems. Unlike the previous boundary conditions, we also introduce a penalty term, to improve numerical stability [61]:

$$\mathscr{B}_S(\boldsymbol{W}^h, \boldsymbol{U}^h) = \left\langle \boldsymbol{W}^h, u_n \boldsymbol{U}^h + \alpha_{\text{slip}}\, c_s(\boldsymbol{U}^h) \left\{ \begin{matrix} 0 \\ \rho(\boldsymbol{U}^h)\,(\boldsymbol{u}(\boldsymbol{U}^h)\cdot\boldsymbol{n} - u_n)\,\boldsymbol{n} \\ 0 \end{matrix} \right\} + \left\{ \begin{matrix} 0 \\ p(\boldsymbol{U}^h)\,\boldsymbol{n} \\ p(\boldsymbol{U}^h)\,u_n \end{matrix} \right\} \right\rangle_{\Gamma_S} . \tag{2.17}$$

Here $c_s(\boldsymbol{U}^h)$ is the local speed of sound and $\alpha_{\text{slip}} = O(1)$ is a parameter that is fixed to $\alpha_{\text{slip}} = 0.2$ in all tests, unless otherwise stated.

- Lastly, at a far-field boundary Γ_F, we assume the entire ambient fluid state vector $\boldsymbol{U}_b$ is prescribed and the boundary term is given by

$$\mathscr{B}_F(\boldsymbol{W}^h, \boldsymbol{U}^h) = \left\langle \boldsymbol{W}^h, \boldsymbol{F}_{\text{Roe}}(\boldsymbol{U}^h, \boldsymbol{U}_b; \boldsymbol{n}) \right\rangle_{\Gamma_F} , \tag{2.18}$$

where $\boldsymbol{F}_{\text{Roe}}(\cdot, \cdot; \boldsymbol{n})$ is the classical Roe flux in the direction $\boldsymbol{n}$ [62]. Alternatively, other numerical fluxes can be used instead of the Roe flux, see for example the text on approximated Riemann solvers [59].

2.4. *The discontinuity-capturing operator*

As shocks appear frequently in compressible inviscid flows, we use artificial viscosities to capture discontinuities, implemented as

$$\mathscr{B}_{\text{dc}}(\boldsymbol{W}^h, \boldsymbol{U}^h) = (\nabla \boldsymbol{W}^h, \nu_h \nabla \boldsymbol{U}^h)_\Omega , \tag{2.19}$$

where ν_h is the artificial viscosity that scales also with the local element size h. Two recently proposed artificial viscosities are considered in this work. For most computations we adopt the residual-consistent entropy viscosity proposed by Guermond et al. [53, 54], which is strong in the vicinity of shocks and vanishes in smooth regions of the flow, where second-order accuracy is then maintained. However, when there are complex wave structure or very intense shocks, the residual-consistent entropy viscosity requires specific tuning, and we preferred in this case to use a simpler Lax-type viscosity, equipped with a modified Ducros sensor/limiter [55, 56]. This specific sensor/limiter targets the viscosity only in regions where the flow is compressive. We describe these two approaches next.

2.4.1. *The entropy viscosity*

Denote the entropy of Euler flow by $S = \frac{\rho}{\gamma-1}\log\left(\frac{p}{\rho^\gamma}\right)$, then $(S, S\boldsymbol{u})$ is an entropy pair and the entropy residual:

$$\mathrm{Res}_S = S_t + \nabla\cdot(S\boldsymbol{u}) + (S_{,\boldsymbol{U}})^T\boldsymbol{B} \tag{2.20}$$

is zero in smooth-flow region and it is positive across a shock. At the semi-discrete level, for each element $T \in \mathscr{T}^h$ we compute the semi-discrete entropy residual:

$$\mathrm{Res}_S^h(\boldsymbol{x}_T) := S(\boldsymbol{U}^h(\boldsymbol{x}_T))_{,t} + \nabla\cdot(S(\boldsymbol{U}^h(\boldsymbol{x}_T))\boldsymbol{u}(\boldsymbol{U}^h(\boldsymbol{x}_T))) + (S_{,\boldsymbol{U}}(\boldsymbol{U}^h(\boldsymbol{x}_T)))^T\boldsymbol{B}^h\ , \tag{2.21}$$

where $\boldsymbol{x}_T$ is the centroid of T. The approximation to the temporal derivative in the first term on the right hand side follows the same procedure in the predictor/multi-corrector method described later in Section 2.5. A provisional viscosity ν_h^{ent} is then calculated for the element T as:

$$\nu_h^{\mathrm{ent}} = \nu_T^{\mathrm{ent}} := \frac{c_{\mathrm{ent}} h_T^2 \left|\mathrm{Res}_S^h(\boldsymbol{x}_T)\right|}{\left|\left|S - \overline{S}\right|\right|_\infty}\ , \tag{2.22}$$

where $\overline{S} = \int_\Omega S(\boldsymbol{U}^h(\boldsymbol{x}))d\boldsymbol{x}/\left|\Omega\right|$ is the mean entropy over the entire domain and $\left|\left|S - \overline{S}\right|\right|_\infty = \max_A \left|S(\boldsymbol{U}^h(\boldsymbol{x}_A)) - \overline{S}\right|$, with A running over all nodes in Ω and $\boldsymbol{x}_A$ the nodal coordinates. In general, the tunable parameter $c_{\mathrm{ent}} = O(1)$ depends on the choice of entropy pair and for the one used here we fix $c_{\mathrm{ent}} = 4.0$ in all computations. To avoid overly large artificial viscosity at strong shocks due to large entropy residual and to avoid large artificial viscosity in a single layer of elements, the provisional viscosity is first capped by the classical Lax viscosity $\nu_h^{\mathrm{Lax}} = \nu_T^{\mathrm{Lax}} := \frac{1}{2}h_T\left[\left|\left|\boldsymbol{u}(\boldsymbol{U}^h(\boldsymbol{x}_T))\right|\right| + c_s(\boldsymbol{U}^h(\boldsymbol{x}_T))\right]$ and then smoothened to obtain the final entropy viscosity:

$$\nu_h = \mathbb{S}(\tilde{\nu}_h)\ , \quad \tilde{\nu}_h = \min\left(\nu_h^{\mathrm{ent}}, \nu_h^{\mathrm{Lax}}\right)\ . \tag{2.23}$$

Here $\mathbb{S}$ is a smoothing operator defined as follows. First, we compute at each node A a nodal viscosity $\hat{\nu}_A$ that is the maximum of all $\tilde{\nu}_{h|T}$ such that A is a vertex of the element T; next the smoothed elemental viscosity ν_T is computed as the average of nodal viscosities $\hat{\nu}_A$, where A is a vertex of T.

2.4.2. *The Lax viscosity with a Ducros-type sensor/limiter*

While the entropy residual viscosity does an excellent job in localizing the discontinuity capturing operator near shocks, in our experience it is too aggressive for problems with complex wave structure and very strong shocks, and requires more sophisticated tuning. In these cases, we prefer a more conservative artificial viscosity that activates the Lax viscosity in compressive flow regions (which includes shocks), using a Ducros-type sensor:

$$\nu_h = \max\left(0, \frac{\nabla\cdot\boldsymbol{u}}{\sqrt{(\nabla\cdot\boldsymbol{u})^2 + |c_s|^2}}\right) c_{\mathrm{Lax}} h(||\boldsymbol{u}|| + c_s)\ , \tag{2.24}$$

where $c_{\mathrm{Lax}} \leq 0.5$ is a tunable parameter.

Remark 2.1 (On the use of artificial viscosities). We point out that the object of our contribution is not to find the perfect artificial viscosity to use in computations, but to show the performance of the shifted boundary approach in combination with the prevailing shock capturing operators for finite element discretizations.

2.5. *Explicit predictor/multi-corrector time marching method*

Let t_n and $t_{n+1} = t_n + \Delta t_n$ be two consecutive times instants in the simulation. Denoting the discrete solution at t_j by $\boldsymbol{U}_j^h$, with $j = n, n+1$, we describe below an explicit predictor/multi-corrector (PMC) time marching method. The iterate $\boldsymbol{U}_{n+1}^{h;(k)}$ is recursively constructed for iteration $k = 0, 1, \cdots, k_{\max}$, according to

$$\left(\boldsymbol{W}^h, \frac{\boldsymbol{U}_{n+1}^{h;(k+1)} - \boldsymbol{U}_n^h}{\Delta t_n}\right)_\Omega - (\nabla\boldsymbol{W}^h, \boldsymbol{U}_{n+1/2}^{h;(k)} \otimes \boldsymbol{u}_{n+1/2}^{(k)} + \boldsymbol{G}(\boldsymbol{U}_{n+1/2}^{h;(k)}))_\Omega + (\boldsymbol{W}^h, \boldsymbol{B}_{n+1/2}^{(k)})_\Omega$$

$$+\mathscr{B}_{\text{vms}}(\boldsymbol{W}^h, \boldsymbol{U}_*^{h;(k)}) + \mathscr{B}_{\text{dc}}(\boldsymbol{W}^h, \boldsymbol{U}_{n+1/2}^{h;(k)}) + \mathscr{B}_{\text{bc}}(\boldsymbol{W}^h, \boldsymbol{U}_{n+1/2}^{h;(k)}) = 0\,, \tag{2.25}$$

where $\boldsymbol{U}_{n+1}^{h;(0)} = \boldsymbol{U}_n^h$, $\boldsymbol{U}_{n+1/2}^{h;(k)} = (\boldsymbol{U}_{n+1}^{h;(k)} + \boldsymbol{U}_n^h)/2$, $\boldsymbol{u}_{n+1/2}^{(k)} = \boldsymbol{u}(\boldsymbol{U}_{n+1/2}^{h;(k)})$, and $\boldsymbol{B}_{n+1/2}^{(k)}$ is defined similarly. In this work, $k_{\max} = 3$ is used for all numerical tests. The asterisk in $\mathscr{B}_{\text{vms}}(\boldsymbol{W}^h, \boldsymbol{U}_*^{h;(k)})$ indicates that $\mathscr{B}_{\text{vms}}(\boldsymbol{W}^h, \cdot)$ depends on an appropriate nonlinear expression depending on $\boldsymbol{U}_n^h$ and $\boldsymbol{U}_{n+1}^{h;(k)}$, as shown in the next section. The time step Δt_n is determined by the standard Courant condition:

$$\Delta t_n = \alpha_{\text{cfl}} \min_{T \in \mathscr{T}^h} \left(\min_{A \in \mathscr{N}(T)} \frac{h_T^2}{\nu_T^2 + \sqrt{\nu_T^2 + (\left|\left|\boldsymbol{u}(\boldsymbol{U}_{n;A}^h)\right|\right| + c_s(\boldsymbol{U}_{n;A}^h))^2 h_T^2}} \right), \tag{2.26}$$

where the first minimum runs over all elements T in the mesh, the second minimum runs over the set $\mathscr{N}(T)$ of nodes A of element T, h_T denotes the size of element T computed as the diameter of the largest circle/sphere inscribed to the element, ν_T is the artificial viscosity computed for the element T (see the previous section), $\boldsymbol{U}_{n;A}^h$ is the numerical solution at node A and time instant t_n, and c_s is the speed of sound, previously defined. The parameter α_{cfl} designates the user-specified Courant number with a range between zero and one for an explicit time integrator. This formula is motivated by the one derived and studied in [50], and we pick $\alpha_{\text{cfl}} = 0.9$ for all numerical tests in the current work.

2.6. *The variational multiscale stabilization term*

The Variational Multiscale Stabilization (VMS) was originally designed to stabilize standard Galerkin discretizations of the transport equations [63–66]. Later on, it was applied to the compressible Euler equations and hyperbolic systems of conservation laws, including moving boundaries [50, 51, 67–77]. The basic idea is to take into account the effect of an under-resolved component of the solution $\boldsymbol{U}'$ in the flux term $\boldsymbol{F}(\boldsymbol{U}_{n+1/2}^{h;(k)})$. In particular, instead of using $\boldsymbol{F}(\boldsymbol{U}_{n+1/2}^{h;(k)}) = \boldsymbol{U}_{n+1/2}^{h;(k)} \otimes \boldsymbol{u}(\boldsymbol{U}_{n+1/2}^{h;(k)}) + \boldsymbol{G}(\boldsymbol{U}_{n+1/2}^{h;(k)})$, one considers $\boldsymbol{F}(\boldsymbol{U}_{n+1/2}^{h;(k)} + \boldsymbol{U}')$, introducing the linearization

$$(\nabla \boldsymbol{W}^h, \boldsymbol{F}(\boldsymbol{U}_{n+1/2}^{h;(k)} + \boldsymbol{U}'))_\Omega \approx (\nabla \boldsymbol{W}^h, \boldsymbol{F}(\boldsymbol{U}_{n+1/2}^{h;(k)}))_\Omega + (\nabla \boldsymbol{W}^h, \boldsymbol{F}_{,\boldsymbol{U}}(\boldsymbol{U}_{n+1/2}^{h;(k)})\boldsymbol{U}')\,.$$

By scaling arguments and in analogy with *a posteriori* error estimation, a typical model for $\boldsymbol{U}$ is given as

$$\boldsymbol{U}' = -\frac{\Delta t_n}{2\alpha_{\text{cfl}}} \text{Res}(\boldsymbol{U}_*^{h;(k)})\,. \tag{2.27}$$

Here $\alpha_{\text{cfl}} \in (0,\, 1)$ is the Courant number that is used to compute the time step size (hence $\Delta t_n / \alpha_{\text{cfl}}$ is approximately the element size times the maximum characteristic speed), and $\text{Res}(\boldsymbol{U}_*^{h;(k)})$ is the residual of the Euler equations, written only in terms of the coarse-scale component of the solution $\boldsymbol{U}_*^{h;(k)}$, namely:

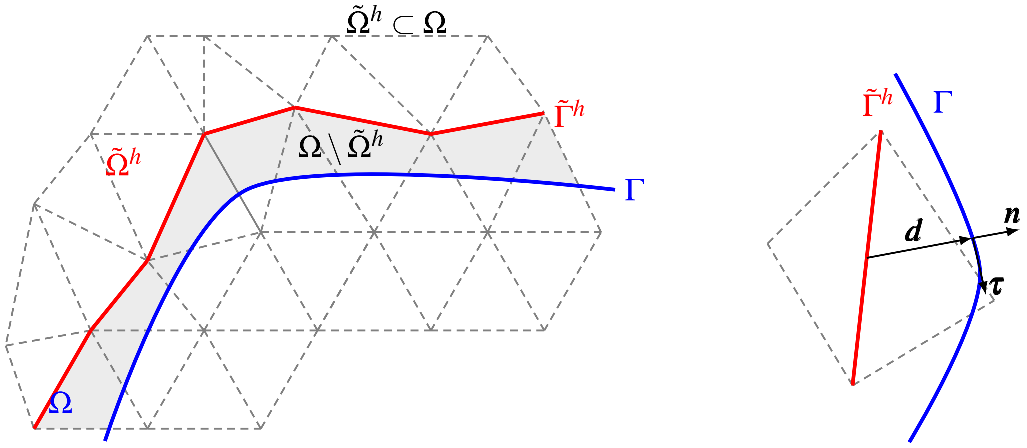
$$\text{Res}(\boldsymbol{U}_*^{h;(k)}) = \frac{\boldsymbol{U}_{n+1}^{h;(k)} - \boldsymbol{U}_n^h}{\Delta t_n} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}_{n+1/2}^{h;(k)}) + \boldsymbol{B}_{n+1/2}^{(k)} \tag{2.28}$$

Hence the VMS term in (2.7) is given by:

$$\mathscr{B}_{\text{vms}}(\boldsymbol{W}^h, \boldsymbol{U}_*^{h;(k)}) = \left(\nabla \boldsymbol{W}^h, \frac{\Delta t_n}{2\alpha_{\text{cfl}}} \boldsymbol{F}_{,\boldsymbol{U}}(\boldsymbol{U}_{n+1/2}^{h;(k)}) \text{Res}(\boldsymbol{U}_*^{h;(k)}) \right)_\Omega. \tag{2.29}$$

3. The Shifted Boundary Method

We describe now the general strategy by which immersed boundary condition are enforced with the SBM, and the implementation of boundary conditions specific to the Euler equations.



(a) The true domain Ω, the surrogate domain $\tilde{\Omega}^h \subset \Omega$ and their boundaries $\tilde{\Gamma}_h$ and Γ.

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

Figure 3.1: The surrogate domain, its boundary, and the distance vector $\boldsymbol{d}$.

3.1. The true domain, the surrogate domain and maps

Recall that Ω is a connected open set in $\mathbb{R}^d$ of Lipschitz boundary $\Gamma = \partial\Omega$ with normal $\boldsymbol{n}$ and consider a closed domain $\mathscr{D}$ such that $\text{clos}(\Omega) \subseteq \mathscr{D}$. We introduce a family $\mathscr{T}^h$ of admissible and shape-regular tessellations of $\mathscr{D}$ (note that before $\mathscr{T}^h$ was used to indicate a family of tessellations of Ω). We indicate by h_T the size of element $T \in \mathscr{T}^h$ and by h the piecewise constant function such that $h_{|T} = h_T$. We restrict each triangulation by selecting those elements that are contained in $\text{clos}(\Omega)$, i.e., we form

$$\tilde{\mathscr{T}}_h := \{T \in \mathscr{T}^h : T \subset \text{clos}(\Omega)\}\,,$$

which identifies the *surrogate domain*

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

with *surrogate boundary* $\tilde{\Gamma}^h := \partial\tilde{\Omega}^h$ and outward-oriented unit normal vector $\tilde{\boldsymbol{n}}$ to $\tilde{\Gamma}^h$. Obviously, $\tilde{\mathscr{T}}_h$ is an admissible, shape-regular triangulation of $\tilde{\Omega}^h$ (see Figure 3.1a). We now introduce a mapping

$$\boldsymbol{M}^h : \tilde{\Gamma}^h \to \Gamma\,, \tag{3.1a}$$

$$\tilde{\boldsymbol{x}} \mapsto \boldsymbol{x}\,, \tag{3.1b}$$

which associates to any point $\tilde{\boldsymbol{x}} \in \tilde{\Gamma}^h$ on the surrogate boundary a point $\boldsymbol{x} = \boldsymbol{M}^h(\tilde{\boldsymbol{x}})$ on the physical boundary Γ. Whenever uniquely defined, the closest-point projector of $\tilde{\boldsymbol{x}}$ upon Γ is a natural choice for $\boldsymbol{M}^h$, as shown e.g. in Figure 3.1b. Note that, in general, $\boldsymbol{n}(\boldsymbol{M}^h(\tilde{\boldsymbol{x}})) \neq \tilde{\boldsymbol{n}}(\tilde{\boldsymbol{x}})$. Through $\boldsymbol{M}^h$, a distance vector function $\boldsymbol{d}_{\boldsymbol{M}^h}$ can be defined as

$$\boldsymbol{d}_{\boldsymbol{M}^h}(\tilde{\boldsymbol{x}}) = \boldsymbol{x} - \tilde{\boldsymbol{x}} = [\boldsymbol{M}^h - \boldsymbol{I}](\tilde{\boldsymbol{x}})\,. \tag{3.2}$$

For the sake of simplicity, we set $\boldsymbol{d} = \boldsymbol{d}_{\boldsymbol{M}^h}$ where $\boldsymbol{d} = \|\boldsymbol{d}\|\boldsymbol{\nu}$ and $\boldsymbol{\nu}$ is a unit vector. Similarly, we will write $\boldsymbol{n}$ instead of $\boldsymbol{n}(\boldsymbol{M}^h(\tilde{\boldsymbol{x}}))$ and $\tilde{\boldsymbol{n}}$ instead of $\tilde{\boldsymbol{n}}(\tilde{\boldsymbol{x}})$, whenever there is no ambiguity.

Remark 3.1. If $\boldsymbol{x} = \boldsymbol{M}^h(\tilde{\boldsymbol{x}})$ does not belong to corners or edges, then the closest-point projection implies $\boldsymbol{\nu} = \boldsymbol{n}$, where $\boldsymbol{n}$ was defined as the outward pointing normal to Γ. More sophisticated choices may be locally preferable in the presence of corners or edges and we refer to [42] for more details.

Remark 3.2. The map $\boldsymbol{M}^h$ and the distance $\boldsymbol{d}$ can be defined with strategies other than the closest-point projection, such as level sets, for which $\boldsymbol{d}$ is defined by means of a signed distance function.

In case the boundary is partitioned as $\Gamma = \operatorname{clos}\left(\cup_{j=1}^{N}\Gamma_j\right)$, with $\Gamma_j \cap \Gamma_k = \emptyset$ for any $j \neq k$, also the surrogate boundary is partitioned as $\tilde{\Gamma}^h = \operatorname{clos}\left(\cup_{j=1}^{\tilde{N}}\tilde{\Gamma}_j^h\right)$, with $\tilde{\Gamma}_j^h \cap \tilde{\Gamma}_k^h = \emptyset$ for any $j \neq k$, using again the map $\boldsymbol{M}^h$:

$$\tilde{\Gamma}_j^h = \{\tilde{e} \subseteq \tilde{\Gamma}^h : \boldsymbol{M}^h(\tilde{e}) \subseteq \Gamma_j\}\,, \tag{3.3}$$

where $\tilde{e}$ is an edge/face of the grid on the surrogate domain. It will also be convenient to introduce the projections of a vector $\boldsymbol{w}$ along the normal $\boldsymbol{n}$ and tangent $\boldsymbol{\tau}$ to the true boundary, namely:

$$\boldsymbol{\pi_n}(\boldsymbol{w}) := (\boldsymbol{w}\cdot\boldsymbol{n})\boldsymbol{n}\,, \quad \boldsymbol{\pi_\tau}(\boldsymbol{w}) := \boldsymbol{w} - \boldsymbol{\pi_n}(\boldsymbol{w})\,, \quad \forall \boldsymbol{w} \in \mathbb{R}^d\,. \tag{3.4}$$

3.2. *General strategy*

In the SBM, the governing equations are discretized in $\tilde{\Omega}^h$ rather than in Ω, with the challenge of accurately imposing boundary conditions on $\tilde{\Gamma}^h$. To this end, boundary conditions are *shifted* from Γ to $\tilde{\Gamma}^h$, by performing a Taylor expansion of the variable of interest at the surrogate boundary. Under the assumption that u is sufficiently smooth in the strip between $\tilde{\Gamma}^h$ and Γ, let

$$\mathsf{D}_{\boldsymbol{d}}^i \boldsymbol{u} = \sum_{\boldsymbol{\alpha}\in\mathbb{N}^n, |\boldsymbol{\alpha}|=i} \frac{i!}{\boldsymbol{\alpha}!}\frac{\partial^i \boldsymbol{u}}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}}\boldsymbol{d}^{\boldsymbol{\alpha}}$$

denote the ith-order directional derivative along $\boldsymbol{d}$. Then, we can write

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}(\tilde{\boldsymbol{x}} + \boldsymbol{d}(\tilde{\boldsymbol{x}})) = \boldsymbol{u}(\tilde{\boldsymbol{x}}) + \sum_{i=1}^{m} \frac{\mathsf{D}_{\boldsymbol{d}}^i \boldsymbol{u}(\tilde{\boldsymbol{x}})}{i!} + (\mathsf{R}^m(\boldsymbol{u},\boldsymbol{d}))(\tilde{\boldsymbol{x}})\,, \tag{3.5}$$

where the remainder $\mathsf{R}^m(\boldsymbol{u},\boldsymbol{d})$ satisfies $|\mathsf{R}^m(\boldsymbol{u},\boldsymbol{d})| = o(\|\boldsymbol{d}\|^m)$ as $\|\boldsymbol{d}\| \to 0$.

In this work, we consider piece-wise linear approximations of the solution fields, and for this reason we limit the Taylor expansions to the first two terms, that is,

$$\boldsymbol{u}(\boldsymbol{x}) \approx \mathsf{S}_{\boldsymbol{d}}\boldsymbol{u}(\tilde{\boldsymbol{x}})\,, \tag{3.6}$$

where

$$\mathsf{S}_{\boldsymbol{d}}\boldsymbol{u}(\tilde{\boldsymbol{x}}) := \boldsymbol{u}(\tilde{\boldsymbol{x}}) + \boldsymbol{d}\cdot\nabla\boldsymbol{u} \tag{3.7}$$

is the *shift* operator for every $\tilde{\boldsymbol{x}} \in \tilde{\Gamma}^h$.

It is now important to make an observation, in the context of compressible Euler equations: the far-field boundary is usually rather regular in shape, and it makes less sense to treat it as immersed. Hence, we will not consider the case of embedded far-field boundary conditions, and the shifted boundary terms in the variational formulation becomes:

$$\mathscr{B}_{\mathrm{bc}}(\cdot,\cdot) = \mathscr{B}_{I/O/F/S}(\cdot,\cdot) + \tilde{\mathscr{B}}_{I/O/S}(\cdot,\cdot)\,, \tag{3.8}$$

where $\mathscr{B}_{I/O/F/S}(\cdot,\cdot) := \mathscr{B}_{I;sub}(\cdot,\cdot) + \mathscr{B}_{I;sup}(\cdot,\cdot) + \mathscr{B}_{O;sub}(\cdot,\cdot) + \mathscr{B}_{O;sup}(\cdot,\cdot) + \mathscr{B}_{F}(\cdot,\cdot) + \mathscr{B}_{S}(\cdot,\cdot)$ and $\tilde{\mathscr{B}}_{I/O/S}(\cdot,\cdot) = \tilde{\mathscr{B}}_{I;sub}(\cdot,\cdot) + \tilde{\mathscr{B}}_{I;sup}(\cdot,\cdot) + \tilde{\mathscr{B}}_{O;sub}(\cdot,\cdot) + \tilde{\mathscr{B}}_{O;sup}(\cdot,\cdot) + \tilde{\mathscr{B}}_{S}(\cdot,\cdot)$. In Section 3.3, we will discuss in detail how to construct the slip-wall shifted boundary conditions. Other types of shifted boundary conditions are addressed, in less detail, in Section 3.4 (see also [49]).

3.3. *Enforcing the slip boundary condition at a surrogate boundary*

The main idea is to shift the slip boundary condition $\boldsymbol{u}\cdot\boldsymbol{n} = u_n$ on Γ_S to a condition for $\boldsymbol{u}\cdot\tilde{\boldsymbol{n}}$ on its surrogate $\tilde{\Gamma}_S$, using Taylor expansions and the projections given in (3.4). In particular, letting $\tilde{\boldsymbol{x}} \in \tilde{\Gamma}_S$, assuming sufficient smoothness in the velocity field $\boldsymbol{u}$, we have:

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}(\tilde{\boldsymbol{x}} + \boldsymbol{d}) \approx \boldsymbol{u}(\tilde{\boldsymbol{x}}) + \boldsymbol{d}\cdot\nabla\boldsymbol{u}(\tilde{\boldsymbol{x}})$$

and consequently

$$
\begin{aligned}
\boldsymbol{u}(\tilde{\boldsymbol{x}})\cdot\tilde{\boldsymbol{n}} &= [\boldsymbol{\pi}_{\boldsymbol{n}}(\boldsymbol{u}(\tilde{\boldsymbol{x}}))+\boldsymbol{\pi}_{\boldsymbol{\tau}}(\boldsymbol{u}(\tilde{\boldsymbol{x}}))]\cdot\tilde{\boldsymbol{n}} = (\boldsymbol{u}(\tilde{\boldsymbol{x}})\cdot\boldsymbol{n})(\boldsymbol{n}\cdot\tilde{\boldsymbol{n}})+\boldsymbol{\pi}_{\boldsymbol{\tau}}(\boldsymbol{u}(\tilde{\boldsymbol{x}}))\cdot\tilde{\boldsymbol{n}}\\
&\approx [\boldsymbol{u}(\boldsymbol{x})\cdot\boldsymbol{n}-(\boldsymbol{d}\cdot\nabla\boldsymbol{u}(\tilde{\boldsymbol{x}}))\cdot\boldsymbol{n}]\,(\boldsymbol{n}\cdot\tilde{\boldsymbol{n}})+\boldsymbol{\pi}_{\boldsymbol{\tau}}(\boldsymbol{u}(\tilde{\boldsymbol{x}}))\cdot\tilde{\boldsymbol{n}}\\
&= [u_n-(\boldsymbol{d}\cdot\nabla\boldsymbol{u}(\tilde{\boldsymbol{x}}))\cdot\boldsymbol{n}]\,(\boldsymbol{n}\cdot\tilde{\boldsymbol{n}})+\boldsymbol{\pi}_{\boldsymbol{\tau}}(\boldsymbol{u}(\tilde{\boldsymbol{x}}))\cdot\tilde{\boldsymbol{n}}\,.
\end{aligned}
$$

Here we use the short-hand notation $\boldsymbol{x}=\boldsymbol{M}^h(\tilde{\boldsymbol{x}})$, $\tilde{\boldsymbol{n}}=\tilde{\boldsymbol{n}}(\tilde{\boldsymbol{x}})$, $\boldsymbol{n}=\boldsymbol{n}(\boldsymbol{M}^h(\tilde{\boldsymbol{x}}))$, and $\boldsymbol{d}=\boldsymbol{d}(\tilde{\boldsymbol{x}})$. Note that both $\nabla\boldsymbol{u}(\tilde{\boldsymbol{x}})$ and $\boldsymbol{\pi}_{\boldsymbol{\tau}}(\boldsymbol{u}(\tilde{\boldsymbol{x}}))$ can be computed by processing the discrete solutions on the surrogate boundary. Hence, the shifted slip boundary condition is given by:

$$
\boldsymbol{u}\cdot\tilde{\boldsymbol{n}}=\tilde{u}_n := u_n\boldsymbol{n}\cdot\tilde{\boldsymbol{n}}-[(\boldsymbol{d}\cdot\nabla\boldsymbol{u})\cdot\boldsymbol{n}]\,(\boldsymbol{n}\cdot\tilde{\boldsymbol{n}})+\boldsymbol{\pi}_{\boldsymbol{\tau}}(\boldsymbol{u})\cdot\tilde{\boldsymbol{n}}\,,\quad \text{on }\tilde{\Gamma}_S\,, \tag{3.9}
$$

and the boundary term $\tilde{\mathscr{B}}_S$ is:

$$
\tilde{\mathscr{B}}_S(\boldsymbol{W}^h,\boldsymbol{U}^h)=\left\langle \boldsymbol{W}^h,\,\tilde{u}_n\boldsymbol{U}^h+\alpha_{\mathrm{slip}}c_s(\boldsymbol{U}^h)\begin{Bmatrix}0\\ \rho(\boldsymbol{U}^h)(\boldsymbol{u}(\boldsymbol{U}^h)\cdot\tilde{\boldsymbol{n}}-\tilde{u}_n)\tilde{\boldsymbol{n}}\\ 0\end{Bmatrix}+\begin{Bmatrix}0\\ p(\boldsymbol{U}^h)\tilde{\boldsymbol{n}}\\ p(\boldsymbol{U}^h)\tilde{u}_n\end{Bmatrix}\right\rangle_{\tilde{\Gamma}_S}. \tag{3.10}
$$

As a last note, one may argue that (3.9) will not approximate $\boldsymbol{u}\cdot\boldsymbol{n}=u_n$ well if $\boldsymbol{n}\cdot\tilde{\boldsymbol{n}}$ is close to zero; however, this will rarely happen for triangular or tetrahedral elements, see for example the analysis in [42].

3.4. Enforcing other types of boundary conditions at a surrogate boundary

Here we list the boundary terms at a surrogate boundary for other types of boundary conditions.

- At a supersonic inlet $\Gamma_{I;sup}$, the full fluid state vector $\boldsymbol{U}_b$ is prescribed. At a quadrature point on the corresponding surrogate boundary $\tilde{\boldsymbol{x}}\in\tilde{\Gamma}_{I;sup}$ we shift the boundary condition as

  $$
  \boldsymbol{U}(\tilde{\boldsymbol{x}})=\tilde{\boldsymbol{U}}_b := \boldsymbol{U}_b-\boldsymbol{d}\cdot\nabla\boldsymbol{U}(\tilde{\boldsymbol{x}})
  $$

  and, setting $\tilde{\boldsymbol{u}}_b := \boldsymbol{u}(\tilde{\boldsymbol{U}}_b)$ for simplicity, the boundary term $\tilde{\mathscr{B}}_{I;sup}$ is given by

  $$
  \tilde{\mathscr{B}}_{I;sup}(\boldsymbol{W}^h,\boldsymbol{U}^h)=\langle\boldsymbol{W}^h,\,(\tilde{\boldsymbol{U}}_b\otimes\tilde{\boldsymbol{u}}_b+\tilde{\boldsymbol{G}}_b)\tilde{\boldsymbol{n}}\rangle_{\tilde{\Gamma}_{I;sup}}\,. \tag{3.11}
  $$

- At a subsonic inlet $\Gamma_{I;sub}$, the flow density ρ_b and the flow velocity $\boldsymbol{u}_b$ are prescribed. We define the shifted data at every $\tilde{\boldsymbol{x}}\in\tilde{\Gamma}_{I;sub}$ as:

  $$
  \begin{aligned}
  \rho(\tilde{\boldsymbol{x}})&=\tilde{\rho}_b := \rho_b-\boldsymbol{d}\cdot\nabla\rho(\tilde{\boldsymbol{x}})\,,\\
  \boldsymbol{u}(\tilde{\boldsymbol{x}})&=\tilde{\boldsymbol{u}}_b := \boldsymbol{u}_b-\boldsymbol{d}\cdot\nabla\boldsymbol{u}(\tilde{\boldsymbol{x}})\,.
  \end{aligned}
  $$

  Then, the boundary term $\tilde{\mathscr{B}}_{I;sub}$ is given by:

  $$
  \tilde{\mathscr{B}}_{I;sub}(\boldsymbol{W}^h,\boldsymbol{U}^h)=\left\langle\boldsymbol{W}^h,\,\tilde{\boldsymbol{u}}_b\cdot\tilde{\boldsymbol{n}}\begin{Bmatrix}\tilde{\rho}_b\\ \tilde{\rho}_b\tilde{\boldsymbol{u}}_b\\ \frac{\gamma p(\boldsymbol{U}^h)}{\gamma-1}+\frac{\tilde{\rho}_b\tilde{\boldsymbol{u}}_b\cdot\tilde{\boldsymbol{u}}_b}{2}\end{Bmatrix}+\begin{Bmatrix}0\\ p(\boldsymbol{U}^h)\tilde{\boldsymbol{n}}\\ p(\boldsymbol{U}^h)\tilde{\boldsymbol{u}}_b\cdot\tilde{\boldsymbol{n}}\end{Bmatrix}\right\rangle_{\tilde{\Gamma}_{I;sub}}. \tag{3.12}
  $$

- At a supersonic outlet $\Gamma_{O;sup}$, no data is prescribed and the shifted boundary term is essentially the same as in body-fitted case:

  $$
  \tilde{\mathscr{B}}_{O;sup}(\boldsymbol{W}^h,\boldsymbol{U}^h)=\langle\boldsymbol{W}^h,\,(\boldsymbol{U}^h\otimes\boldsymbol{u}(\boldsymbol{U}^h)+\boldsymbol{G}(\boldsymbol{U}^h))\tilde{\boldsymbol{n}}\rangle_{\tilde{\Gamma}_{O;sup}}\,. \tag{3.13}
  $$

- At a subsonic outlet $\Gamma_{O;sub}$, the pressure p_b is prescribed and the boundary condition is shifted to $\tilde{\boldsymbol{x}}\in\tilde{\Gamma}_{O;sub}$ as

  $$
  p(\tilde{\boldsymbol{x}})=\tilde{p}_b := p_b-\boldsymbol{d}\cdot\nabla p(\tilde{\boldsymbol{x}})\,.
  $$

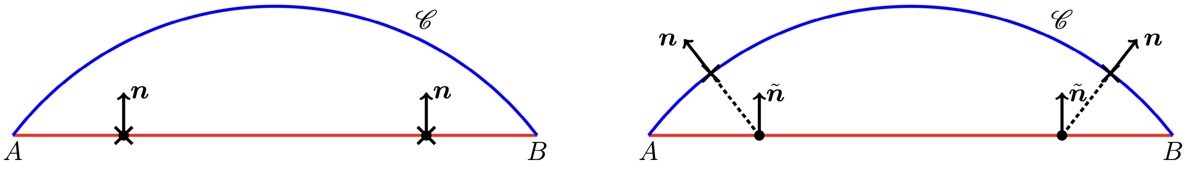
  Hence, the boundary term is

  $$
  \tilde{\mathscr{B}}_{O;sub}(\boldsymbol{W}^h,\boldsymbol{U}^h)=\left\langle\boldsymbol{W}^h,\,\boldsymbol{u}(\boldsymbol{U}^h)\cdot\tilde{\boldsymbol{n}}\begin{Bmatrix}\rho^h\\ \rho^h\boldsymbol{u}(\boldsymbol{U}^h)\\ \frac{\gamma\tilde{p}_b}{\gamma-1}+\frac{\rho^h\boldsymbol{u}(\boldsymbol{U}^h)\cdot\boldsymbol{u}(\boldsymbol{U}^h)}{2}\end{Bmatrix}+\begin{Bmatrix}0\\ \tilde{p}_b\tilde{\boldsymbol{n}}\\ \tilde{p}_b\boldsymbol{u}(\boldsymbol{U}^h)\cdot\tilde{\boldsymbol{n}}\end{Bmatrix}\right\rangle_{\tilde{\Gamma}_{O;sub}}. \tag{3.14}
  $$

3.5. *Enhancing computations with body-fitted grids at curved boundaries*

For computations with both body-fitted and embedded grids, we adopt quadrature rules that integrate exactly cubic functions, that is, the two-point Gauss-Legendre quadrature for edges and the four-point quadrature rule for triangles. Here we briefly discuss the potential of using the SBM to improve the accuracy in the case when boundaries are curved and are represented by body-fitted, affine grids (e.g., when the piecewise-linear finite elements described in Section 2 are used).

To explain the idea, let us consider a two-dimensional edge with its two end points lying on a curved portion $\mathscr{C}$ of the slip boundary Γ_S, as shown in Figure 3.2. In the same figure, the two Gauss-Legendre quadrature points are marked by solid dots. We assume that the location of $\mathscr{C}$ is known: for example, it can be given in analytical form, by a much finer boundary grid, or a CAD geometry representation. In the body-fitted computation, the boundary condition is enforced weakly at the points marked by × in Figure 3.2a, which coincide with the quadrature points. Instead, in the case of the SBM, due to the shifting mechanism, we actually weakly enforce the boundary condition on the true geometry (up to second-order accuracy), at the point marked by × in Figure 3.2b.



(a) Boundary normals for a body-fitted grid.

(b) Boundary normals computed with the SBM.

Figure 3.2: Comparing boundary integrations between a classical method with body-fitted grids and the SBM: • - quadrature points; × - points at which the slip boundary condition is enforced (i.e., normals and velocity values).

We also want to point out an additional issue, which was previously discussed in [78] and addressed in the context of high-order discontinuous Galerkin methods. Because the body-fitted grid matches the actual boundary $\mathscr{C}$ only at the nodes, it enforces the normal velocity in a direction that is orthogonal to the edge $\overline{AB}$, indicated by $\boldsymbol{n}$ in Figure 3.2a. In contrast, the SBM takes the exact normal to $\mathscr{C}$ into consideration, again indicated by $\boldsymbol{n}$ in Figure 3.2b. We will assess the performance improvement of this strategy with specific numerical tests in the next section.

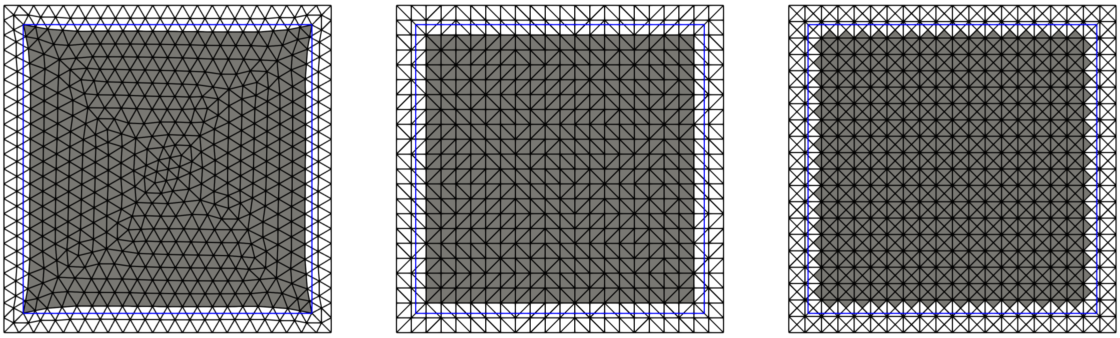
4. Numerical tests

In this section, we assess the numerical performance of the SBM for both transient and steady problems. All tests are run with a CFL condition $\alpha_{\text{cfl}} = 0.9$. In the case of transient computations, $k_{\max} = 3$ corrector steps are used in the PMC time integrator. Steady solutions are computed by integrating in time from an initial state and, since temporal accuracy is not important, only a single corrector step is used ($k_{\max} = 1$). In all tests except the double Mach reflection problem (Section 4.6), the discontinuity capturing operator is defined by the entropy viscosity in Section 2.4.1 with $c_{\text{ent}} = 4.0$; instead, the Lax viscosity with Ducros-type sensor with $c_{\text{Lax}} = 0.1$, as described in Section 2.4.2, is chosen for the double Mach reflection problem of Section 4.6. The computations presented in Section 4.1, 4.2, 4.3, and 4.4 were performed on an in-house workstation that is equipped with a 32-core 2.8 GHz Intel Xeon Scalable Platinum 8362 processor. The computations presented in Section 4.5, 4.6, and 4.7 were performed on a mid-range high-performance cluster composed of 32 nodes with 24 Intel Xeon E5-2650v4 2.2GHz cores.

4.1. *The two-dimensional Taylor-Green vortex test*

The two-dimensional Taylor-Green vortex problem [51] is defined on the square domain $\Omega = [0,\,1]^2$ depicted in Figure 4.1. The fluid has a specific heat capacity ratio $\gamma = 5/3$ and the source term in the governing equations is given by

$$\boldsymbol{B}(\boldsymbol{x},t) = \left\{ \begin{array}{c} 0 \\ \boldsymbol{0} \\ -\rho(\boldsymbol{x},t)\frac{3\pi}{8}(\cos(3\pi x)\cos(\pi y) - \cos(\pi x)\cos(3\pi y)) \end{array} \right\} . \tag{4.1}$$



(a) Mesh 2 of Sequence I.

(b) Mesh 2 of Sequence II.

(c) Mesh 2 of Sequence III.

Figure 4.1: Two-dimensional Taylor-Green vortex test. Examples of grids from Sequence I, II, and III. True boundaries are marked in blue and surrogate domains $\tilde{\Omega}$ are shaded in gray.

Table 1: Two-dimensional Taylor-Green vortex test. Mesh information of the three sequences of triangular grids described in Figure 4.1 and Section 4.1: total area, number of nodes, and number of elements in the surrogate domain.

	Sequence I			Sequence II			Sequence III		
	Area	Nodes	Elements	Area	Nodes	Elements	Area	Nodes	Elements
Mesh 1	0.917336	120	198	0.85983	100	162	0.82204	145	256
Mesh 2	0.917336	437	792	0.85983	361	648	0.873417	609	1,088
Mesh 3	0.979051	1,772	3,378	0.95802	1,521	2,888	0.9553	2,517	4,760
Mesh 4	0.988189	6,994	13,658	0.95802	5,929	11,552	0.983397	9,941	19,600
Mesh 5	0.993284	27,808	54,958	0.983397	24,025	47,432	0.990421	40,041	78,960
Mesh 6	0.994818	110,784	220,254	0.996321	96,721	192,200	0.997496	159,613	318,096

Shifted slip-wall boundary conditions ($\boldsymbol{u} \cdot \boldsymbol{n} = 0$) are set along the entire perimeter of the square domain and are enforced according to the discussion in Section 3.3. This problem admits a smooth and time-independent solution given by

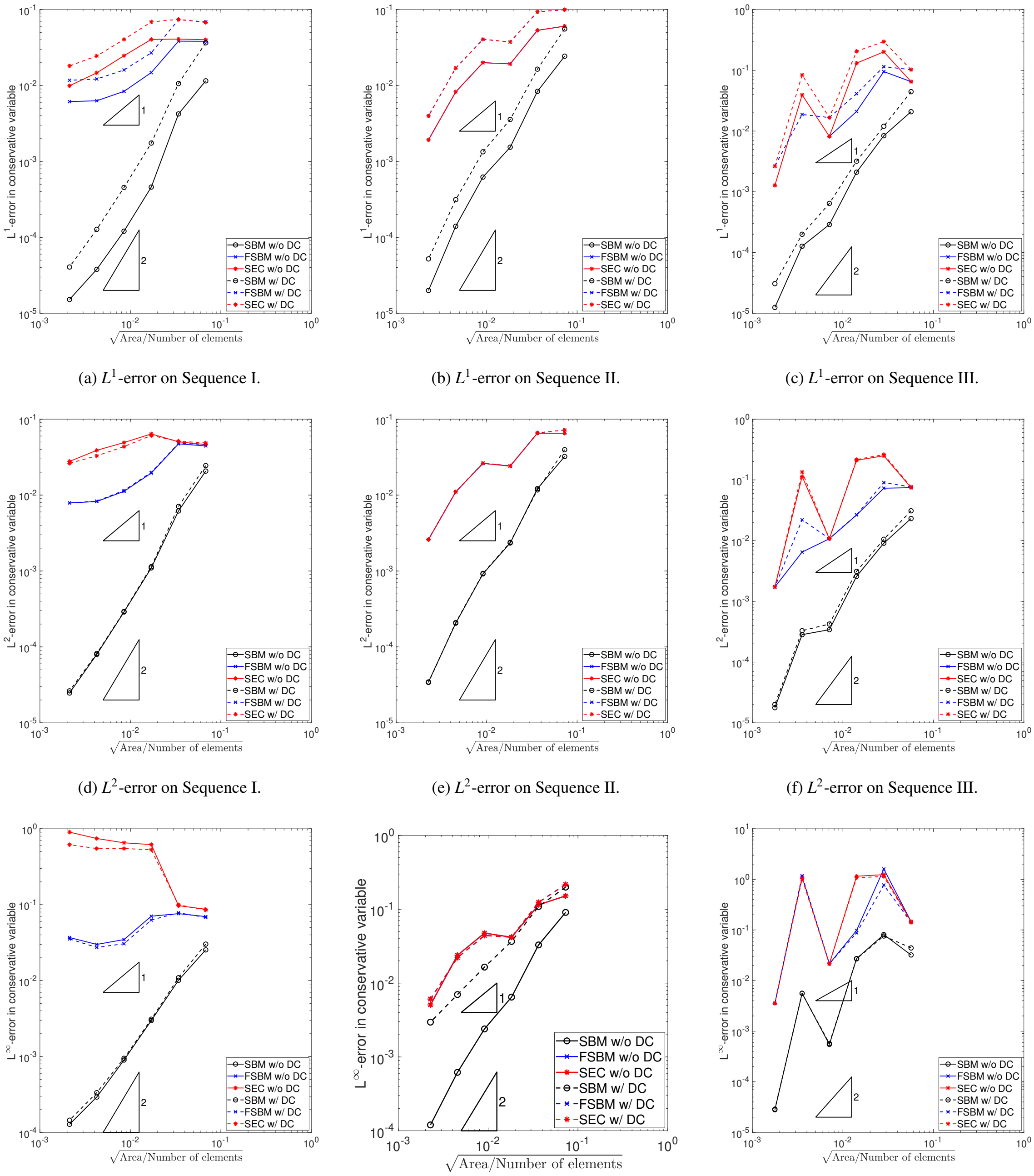
$$\rho(\boldsymbol{x},t) = 1\,, \qquad \boldsymbol{u}(\boldsymbol{x},t) = \begin{Bmatrix} \sin(\pi x)\cos(\pi y) \\ -\cos(\pi x)\sin(\pi y) \end{Bmatrix}\,, \qquad p(\boldsymbol{x},t) = \frac{\cos(2\pi x)+\cos(2\pi y)}{4} + 1\,, \tag{4.2}$$

which will be used to compute the norm of the numerical errors.

We performed a series of tests to assess the performance of the SBM, its mesh-independence, and the impact of artificial viscosity. In particular, we consider three sequences of triangular grids, as illustrated in Figure 4.1 and Table 1, and solve the problem until $T = 0.5$. The first sequence (Sequence I) contains six unstructured triangular grids with (approximate) size ranging from $h = 10^{-1}$ to $h = 3 \times 10^{-3}$. The next two sequences (Sequence II and III) are both obtained from a base sequence of uniform quadrilateral grids (i.e., square elements), again with element size ranging from $h = 10^{-1}$ to $h = 3 \times 10^{-3}$. In particular, grids in Sequence II are obtained by dividing each base square element into two triangles, whereas meshes in Sequence III are obtained by dividing each base square element into four triangles. Note that the surrogate boundary $\tilde{\Gamma}$ tends to be smooth in Sequence II but it could be very rough in Sequence III.

Remark 4.1. It is important to realize that grids in Sequence III are not of good quality, from a mesh generation perspective, since they alternate nodes with a 4-element connectivity with nodes with an 8-element connectivity, producing widely oscillating patterns in the entries of the mass and stiffness matrices associated with the FEM discretization. We also show numerical results for these grids to test the effectiveness of shifted boundary conditions even when the topology of the grids is suboptimal.

In order to demonstrate the effect of incorporating the distance vector $\boldsymbol{d}$, we consider the following three numerical methods: (1) the SBM as described in Section 3, (2) a first-order SBM (FSBM) in which a zeroth-order Taylor series is



(a) L^1-error on Sequence I.

(b) L^1-error on Sequence II.

(c) L^1-error on Sequence III.

(d) L^2-error on Sequence I.

(e) L^2-error on Sequence II.

(f) L^2-error on Sequence III.

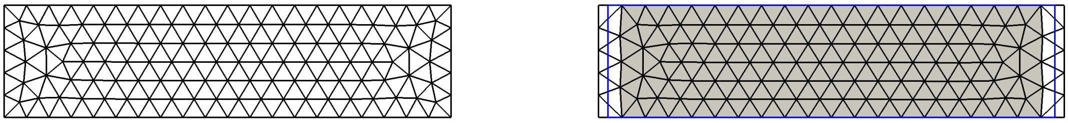
(g) L^∞-error on Sequence I.

(h) L^∞-error on Sequence II.

(i) L^∞-error on Sequence III.

Figure 4.2: Two-dimensional Taylor-Green vortex test. L^1-norms (upper row), L^2-norms (middle row), and L^∞-norms (lower row) of the numerical errors plotted in logarithmic scale against a representative element size for the three sequences of grids. The legends are as follows: SBM indicates the Shifted Boundary Method described in Section 3; FSBM indicates the Shifted Boundary Method with zeroth-order Taylor series expansion ($\boldsymbol{d} = \mathbf{0}$); and SEC indicates a simple embedded computation, in which the FSBM is implemented enforcing $\boldsymbol{n} = \tilde{\boldsymbol{n}}$.

used, and (3) a simple embedded computation (SEC) in which the boundary terms of the body-fitted formulation (2.7) are integrated on the surrogate boundary rather than the true boundary. For each method, we consider computations that have the discontinuity-capturing operator turned on and off, respectively.



(a) Mesh 1 of the sequence of the CMC grids.

(b) Mesh 1 of the sequence of the SBM grids.

Figure 4.3: Channel flow test. Examples of grids used for the CMC and the SBM simulations described in Section 4.2. In the SBM grid (right panel), the true boundary is described by the blue box and the surrogate domain is represented by the shaded region. Detailed information on the grids is summarized in Table 2.

Remark 4.2. With respect to the definition of the SBM variational formulation of Section 3, the FSBM is equivalent to setting $\boldsymbol{d} = \boldsymbol{0}$ and the SEC is equivalent to setting both $\boldsymbol{d} = \boldsymbol{0}$ and $\boldsymbol{n} = \tilde{\boldsymbol{n}}$.

The L^1-norms, L^2-norms, and L^∞-norms of the numerical errors obtained with the SBM, the FSBM, and the SEC are plotted in logarithmic scale in Figure 4.2 for the three sequences of grids. Computations are performed with and without discontinuity-capturing, to highlight differences. The representative element size in the horizontal axis is computed as the square root of the average area of elements in the surrogate domain. Note that the L^1-norms of the errors are qualitatively similar to the L^2-norms; thus in the remaining tests we only consider the L^2-norm and the L^∞-norm.

Looking at the L^2-norms of the errors in Figure 4.2d–Figure 4.2f first, the error curves for the SBM are generally second-order accurate on all three sequences of grids, whether the artificial viscosity is applied or not, as expected. In contrast, both the FSBM (blue curves) and the SEC (red curves) demonstrate at best first-order convergence and are much less accurate than the SBM computations, with or without the artificial viscosity. The FSBM and the SEC also show strong mesh dependency, especially on Sequence II and Sequence III. Particularly, depending on the refinement level, Sequence III contains grids with either zigzagging (see Figure 4.1c) or straight surrogate boundaries and, as a consequence, both FSBM and SEC produce non-monotone convergence curves as shown in Figure 4.2f. In contrast, the L^1-norms and L^2-norms of errors computed by SBM decrease monotonically on all sequences of grids.

Now let us focus on the L^∞-norms of the errors depicted in Figure 4.2g–Figure 4.2i. Overall, the L^∞-errors are larger than the L^2-errors, but have very similar behaviors, except in the situations described next. On Sequence III, the L^∞-error curve by SBM is non-monotone, just like FSBM and SEC; this is again due to the fact that on Sequence III, the surrogate interface switches between straight lines and zigzagging ones as the mesh is refined. Nevertheless, if we compare the L^∞-errors computed over coarsest and finest grids of Sequence III, we observe that the SBM appears to show second-order convergence, while the FSBM and SEC are at best first-order accurate.

Lastly, we mention that when using Sequence II, the straight surrogate boundaries imply $\tilde{\boldsymbol{n}} = \boldsymbol{n}$ and thus the FSBM and the SEC compute exactly the same solutions (see the middle column of Figure 4.2); the same happens on a number but not all grids in Sequence III, as shown in the right column of Figure 4.2.

The conclusion that can be drawn from this first battery of tests is that the SBM is the most accurate and robust approach, whether the discontinuity-capturing operator is active or not.

4.2. Flows in a straight channel

In this second test, we consider horizontal flows in a straight channel, which is given by the rectangular domain $[-2, 2] \times [0, 1]$. Particularly, we first consider a "patch" test for the SBM, solving uniform subsonic and supersonic flows, and then a convergence test for subsonic inlet and outlet boundaries. In all tests, the upper and lower sides of the rectangular domain are slip walls and the left and right edges are inlet and outlet with appropriate boundary conditions to be specified later. For the convergence test, we compare the performance of both the conforming mesh computation (or the CMC) and the SBM on two sequences of six unstructured grids with comparable resolutions. The first grid of each sequence is shown in Figure 4.3, with the shaded region designating the surrogate domain in the case of the SBM; mesh information of all grids is summarized in Table 2. For the uniform-flow tests, we only present results obtained by the SBM on the two coarsest grids (see Figure 4.3b).

Table 2: Channel flow test. Mesh information of the two sequences of triangular grids described in Section 4.2 and Figure 4.3: total area, number of nodes, and number of elements in the surrogate domain.

	CMC Meshes			SBM Meshes		
	Area	Nodes	Elements	Area	Nodes	Elements
Mesh 1	4.0	146	240	3.76889	130	214
Mesh 2	4.0	526	950	3.98011	470	844
Mesh 3	4.0	1,944	3,686	3.97349	1,775	3,358
Mesh 4	4.0	7,568	14,734	3.98204	6,919	13,452
Mesh 5	4.0	29,872	58,942	3.98734	27,629	54,486
Mesh 6	4.0	119,046	236,490	3.99729	109,935	218,325

Table 3: The L^2-norm of errors computed by the SBM for uniform subsonic and supersonic flows in a straight channel, with and without discontinuity capturing.

	$M = 0.3$ w.o. DC	$M = 0.3$ with DC	$M = 1.3$ w.o. DC	$M = 1.3$ with DC
Mesh 1	7.859910940971e-16	6.720757895425e-16	1.987480817517e-15	1.784232331159e-15
Mesh 2	1.037911364231e-15	9.092033378590e-16	4.286117123925e-15	4.074105818579e-15

4.2.1. Uniform subsonic and supersonic flows

First we use the SBM to solve a Mach 0.3 flow and a Mach 1.3 flow going from left to right of the domain, with and without the artificial viscosity. The initial (and thus exact) solution is given by:

$$\rho(\boldsymbol{x},t) = 1.4\,, \quad \boldsymbol{u}(\boldsymbol{x},t) = \begin{Bmatrix} M \\ 0 \end{Bmatrix}\,, \quad p(\boldsymbol{x},t) = 1.0\,, \tag{4.3}$$

where M is the Mach number of the problem. The subsonic inlet and outlet for the Mach 0.3 flow and the supersonic inlet and outlet for the Mach 1.3 flow are applied at the left boundary and right boundary, respectively, as described in Section 3.4. The L^2-norms of the numerical errors are summarized in Table 3 and are at the level of machine precision.

4.2.2. Smooth flow with subsonic inlet and outlet

This test is motivated by [49], where we use the method of manufactured solutions and consider a smooth unsteady flow with specific heat capacity ratio $\gamma = 1.4$ and the source term:

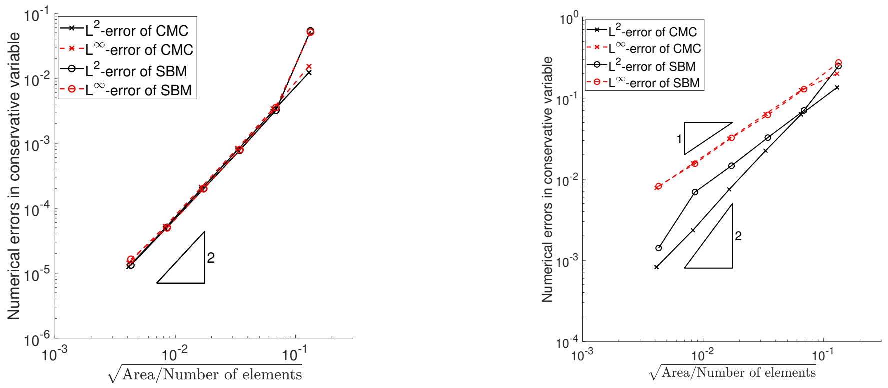
$$\boldsymbol{B}(\boldsymbol{x},t) = \lambda\pi \begin{Bmatrix} \sin(\pi t)\sin(\pi x) - \cos(\pi x)[1+\cos(\pi t)] - \lambda\cos(\pi t)\sin(2\pi x) \\ -B_1[\cos(\pi x)B_2 - \sin(\pi t)\sin(\pi x)] \\ 0 \\ -\dfrac{\cos(\pi x)\{28 + B_1^2[1+\lambda\cos(\pi t)\sin(\pi x)]\} - B_1^2[\sin(\pi t)\sin(\pi x) - \cos(\pi x)B_2]}{2} \end{Bmatrix} \tag{4.4}$$

where $B_1 = 1 + \lambda\sin(\pi x)$ and $B_2 = 2 + \cos(\pi t) + 3\lambda\cos(\pi t)\sin(\pi x)$, so that the smooth flow solution is given by:

$$\rho(\boldsymbol{x},t) = 1 + \lambda\sin(\pi x)\cos(\pi t)\,, \quad \boldsymbol{u}(\boldsymbol{x},t) = \begin{Bmatrix} 1 + \lambda\sin(\pi x) \\ 0 \end{Bmatrix}\,, \quad p(\boldsymbol{x},t) = 4\,. \tag{4.5}$$

When $\lambda = 0.1$, the flow is subsonic everywhere in the channel and we specify the subsonic inlet condition

$$\rho_b(-2,t) = 1\,, \quad \boldsymbol{u}_b(-2,t) = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix}\,, \tag{4.6}$$



(a) Computations without artificial viscosity.

(b) Computations with artificial viscosity.

Figure 4.4: Channel flow test. The L^2-norms and the L^∞-norms of the numerical errors of the test described in Section 4.2, plotted in logarithmic scale against the representative element sizes. CMC: conforming mesh computation; SBM: the Shifted Boundary Method.

at $x = -2$ and the subsonic outlet condition (prescribed pressure)

$$p_b(2,t) = 4\,, \tag{4.7}$$

at $x = 2$. See also Section 2.3 and Section 3.4 for more details about the implementation of the boundary conditions on the body-fitted case and the SBM case, respectively.

Figure 4.4 shows the L^2-norms and the L^∞-norms of the numerical errors obtained with the CMC and the SBM, with and without artificial viscosity, in logarithmic scale. As shown in the plots, the numerical accuracy of the SBM is similar to the CMC, whether the artificial viscosity is applied or not. Particularly, if the artificial viscosity is not applied, we obtain second-order convergence for both the SBM and the CMC in all norms. If instead the artificial viscosity is applied, the convergence of L^2-errors is close to second-order and about first-order for the L^∞-errors.

4.3. Isentropic supersonic vortex in an annulus

Next we consider the stationary flow that describes an isentropic supersonic vortex inside a quarter of an annulus [78] with inner radius $R_i = 1.0$ and outer radius $R_o = 1.384$, shown in Figure 4.5a. We use the sequences of triangular grids depicted in Figure 4.5b and Figure 4.5c. The analytical stationary solution is smooth, and given by

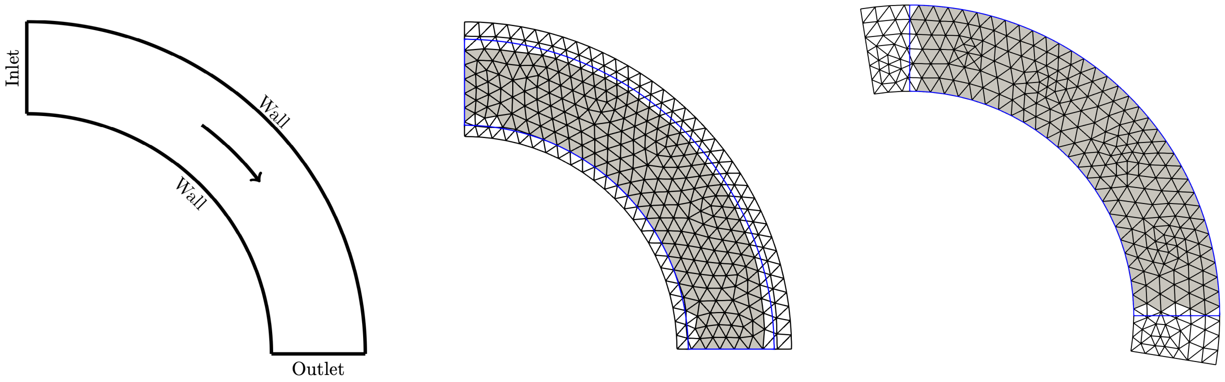
$$\rho(\boldsymbol{x},t) = \left[1 + \frac{\gamma-1}{2} M^2 \left(1 - \frac{R_i^2}{|\boldsymbol{x}|^2}\right)\right]^{\frac{1}{\gamma-1}}, \qquad \boldsymbol{u}(\boldsymbol{x},t) = \left\{ \begin{array}{c} \dfrac{yM}{|\boldsymbol{x}|^2} \\ -\dfrac{xM}{|\boldsymbol{x}|^2} \end{array} \right\}, \qquad p(\boldsymbol{x},t) = \frac{\rho(\boldsymbol{x},t)^\gamma}{\gamma}\,, \tag{4.8}$$

where $M = 2.25$ is the Mach number on the inner circular wall. Nonetheless, we will still consider computations with the artificial viscosity both active and inactive. The fluid state vector is initialized with the exact values (4.8) and numerical computations are driven to a steady state, until $\max_A \left|\left|(\boldsymbol{U}_A^{n+1} - \boldsymbol{U}_A^n)/\Delta t^n\right|\right|_\infty \leq 10^{-10}$, where A runs over all active nodes and $\boldsymbol{U}_A^{n+1}$ and $\boldsymbol{U}_A^n$ are the solution vectors at this node at t^{n+1} and t^n, respectively.

In all computations, the supersonic inlet and the supersonic outlet boundary conditions are specified at the left vertical edge and the lower horizontal edge, respectively, and slip-wall boundary conditions are specified at the inner and outer circular walls. These boundary conditions are implemented either in the body-fitted or SBM fashion, according to the discussion in Section 2.3, Section 3.3, and Section 3.4.

4.3.1. Fitted and embedded curved boundaries

First, we assess the accuracy of the SBM in the presence of curved boundaries, with particular emphasis on the discussion in Section 3.5. The numerical performance of three strategies with piecewise-linear finite elements is compared: (1) geometrically conforming (i.e., body-fitted) mesh computations (CMC); (2) conforming mesh computations



(a) Domain and boundary conditions.

(b) Mesh 1 of the SBM grids used in Section 4.3.1.

(c) Mesh 1 of the SBM grids used in Section 4.3.2.

Figure 4.5: Isentropic supersonic vortex in an annulus. Computational domain and examples of the sequences of grids used in computations where the circular walls are embedded (Section 4.3.1) or the inlet and outlet are embedded (Section 4.3.2). The true boundaries are indicated by the blue curves and the surrogate domains by the shaded regions shaded in gray.

Table 4: Isentropic supersonic vortex in an annulus. Mesh information (area, number of nodes, and number of elements in the surrogate domain) of the three sequences of triangular grids; the CMC grids are used for both tests discussed in Section 4.3.1 and Section 4.3.2.

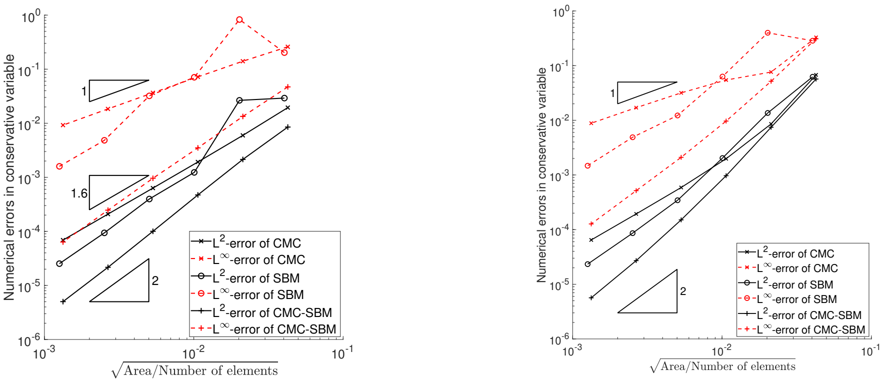
	CMC/CMC-SBM			SBM (Section 4.3.1)			SBM (Section 4.3.2)		
	Area	Nodes	Elements	Area	Nodes	Elements	Area	Nodes	Elements
Mesh 1	0.719023	234	396	0.642321	233	393	0.695393	236	402
Mesh 2	0.719004	863	1,584	0.674759	901	1,650	0.707707	886	1,631
Mesh 3	0.718999	3,309	6,336	0.69603	3,578	6,860	0.712723	3,423	6,564
Mesh 4	0.718998	12,953	25,344	0.707748	14,235	27,882	0.71565	13,457	26,348
Mesh 5	0.718998	51,249	101,376	0.713146	56,767	112,362	0.717431	53,375	105,618
Mesh 6	0.718997	203,873	405,504	0.715838	226,711	451,084	0.718225	212,580	422,895

enhanced with the SBM to treat the curved geometry (CMC-SBM) as described in Section 3.5; and (3) the SBM. All methods are tested on a sequence of six successively refined grids; the grids used for the SBM are shown in Figure 4.5 and, correspondingly, body-fitted grids of analogous sizes are employed for the CMC and the CMC-SBM. The mesh information is summarized in Table 4.

The L^2-norm and the L^∞-norm of the error of the time-converged solutions are plotted in logarithmic scale in Figure 4.6, with the left and right panels showing the plots for computations without and with artificial viscosity, respectively. Figure 4.6a shows that both the SBM and the CMC-SBM are more accurate than the CMC. In particular, the CMC-SBM provides numerical errors that are at least one order of magnitude smaller than the CMC, while the SBM has errors in between the previous two. Both the CMC-SBM and the SBM produce second-order convergence rates, thanks to the accurate capturing of the curved geometry by the shifted boundary condition.

We can obtain essentially the same conclusions for computations with the artificial viscosity included, see Figure 4.6b. Both the SBM and the CMC-SBM produce second-order solutions, whereas the CMC solutions show slower convergence due to polygonal approximation of the curved boundaries.

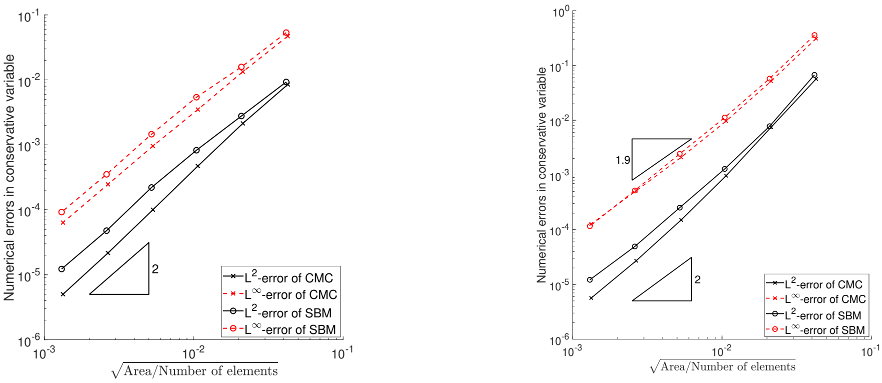
Remark 4.3. In practice, one can use a different (generally more accurate) quadrature rule on edges of the surrogate boundary from the one used for interior edges. For the test under consideration, the 3-point Gauss-Legendre quadrature rule gives very similar results to the default 2-point quadrature rule, for the CMC-SBM. This is possibly due to the circular geometry of the true boundary. If the geometries are more complex, like those with curvature spanning a large range of values, the CMC-SBM offers an attractive strategy to accurately capture the boundary's geometrical features with relatively coarse background grids. Further investigation is warranted to fully confirm this point.



(a) Computations without artificial viscosity.

(b) Computations with artificial viscosity.

Figure 4.6: Isentropic supersonic vortex in an annulus. The L^2-norms and the L^∞-norms of the numerical errors in Section 4.3.1 plotted in logarithmic scale against a representative element size. CMC: conforming mesh computation; SBM: the Shifted Boundary Method on embedded grids; CMC-SBM: conforming mesh computation with SBM-enhanced boundary conditions.



(a) Computations without artificial viscosity.

(b) Computations with artificial viscosity.

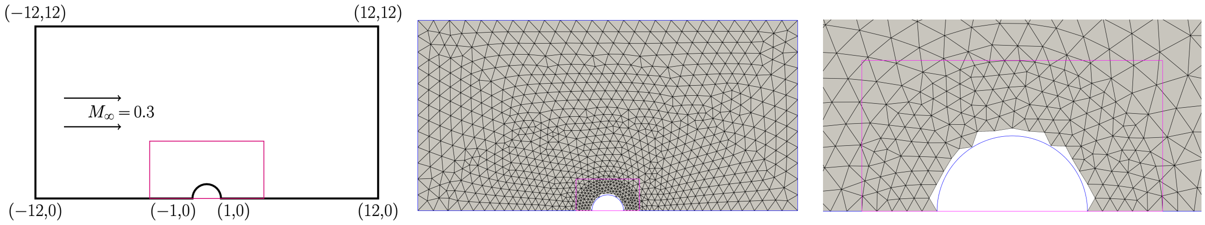
Figure 4.7: Isentropic supersonic vortex in an annulus. The L^2-norms and the L^∞-norms of the numerical errors for the computations described in Section 4.3.2, plotted in logarithmic scale against a representative element size. CMC: conforming mesh computations with piecewise linear elements; SBM: the Shifted Boundary Method on embedded grids.

4.3.2. Fitted and embedded supersonic inlets and outlets

In this second version of the supersonic vortex problem, we consider embedded computations of the inlet and outlet. To minimize the impact on the error of the curved walls, they are treated by conformal grids and enhanced by the SBM. The grids used in the SBM computations share similar resolution with the meshes used in previous conforming computations; the first of this sequence is illustrated in Figure 4.5c and the mesh information is summarized in the last columns of Table 4. As the inlet and outlet are straight lines, we do not consider the CMC-SBM in this test. The convergence plots for the CMC are identical to the CMC-SBM ones reported in Section 4.3.2 and the SBM plots are shown in Figure 4.7a and Figure 4.7b, for computations without and with artificial viscosity, respectively. It is quite clear that the SBM computation delivers very similar results to the CMC results.

4.4. Subsonic flow past a stationary cylinder

In this test, a uniform flow impinges on a stationary cylinder until it reaches steady state. Due to the symmetry of the problem, only the upper half of the computational domain is considered. Specifically, the computational domain consists of a rectangle $[-12, 12] \times [0, 12]$ with a cylinder of radius 1 sitting at the origin, as shown in Figure 4.8a. A sample grid for embedded computations is shown in Figure 4.8b, with element size approximately $h = 0.2$ near



(a) Flow past cylinder problem setup.

(b) An embedded grid with $h = 0.2$ near the cylinder.

(c) Zoomed view near the cylinder.

Figure 4.8: Subsonic flow past a cylinder. Problem configuration. Left panel: domain setup, with a view box in magenta. Middle panel: an embedded domain with reference element length $h = 0.2$ near the cylinder. Right panel: the zoomed view around the embedded geometry.

the cylinder and $h = 1.0$ near the outer boundaries and a zoomed view around the embedded geometry is provided in Figure 4.8c. This grid is for demonstration only, since finer grids are actually used in computations.

A slip wall boundary condition is enforced at the lower edges $[-12, -1] \times \{0\}$ and $[1, 12] \times \{0\}$, to respect symmetry, and the three exterior boundaries $\{-12\} \times [0, 12]$, $\{12\} \times [0, 12]$, and $[-12, 12] \times \{12\}$ have the far-field condition (2.18) specified with the ambient fluid-state vector given by a Mach 0.3 flow. The cylinder surface is either treated as a conforming slip wall, in the CMC computations, or an embedded slip wall, in the SBM and the CMC-SBM computations. The heat capacity ratio is $\gamma = 1.4$ and at time $t = 0.0$ the initial data is given by a uniform Mach 0.3 flow. Namely:

$$\rho_\infty = 1.4, \quad \boldsymbol{u}_\infty = \begin{Bmatrix} 0.3 \\ 0.0 \end{Bmatrix}, \quad p_\infty = 1\,. \tag{4.9}$$

After a transient, a steady-state flow is achieved and is determined when $\max_A \left|\left|(\boldsymbol{U}_A^{n+1} - \boldsymbol{U}_A^n)/\Delta t^n\right|\right|_\infty \leq 10^{-10}$, where A runs over all active nodes.

In the subsonic regime considered here, the steady flow is symmetric about $x = 0$; however, the artificial viscosity tends to tilt the Mach contours towards the right, especially on coarser grids. In fact, it is well-known that p-refinement is much more effective than h-refinement for this test [78, 79]. Hence it is expected to observe asymmetry in all numerical solutions presented here. Nevertheless, our goal is to assess how the SBM handles embedded boundaries, and we compare in Figure 4.9 the time-converged flow computed with the CMC, the SBM, and the CMC-SBM, on grids of comparable resolution. Because the converged Mach number plots are very similar for all methods, we present the results of the most accurate simulations (i.e., the CMC-SBM on the finest grid) in Figure 4.9a, together with the relative discrepancy of the CMC and SBM with respect to the CMC-SBM (on both the coarse and fine grids), which is defined as

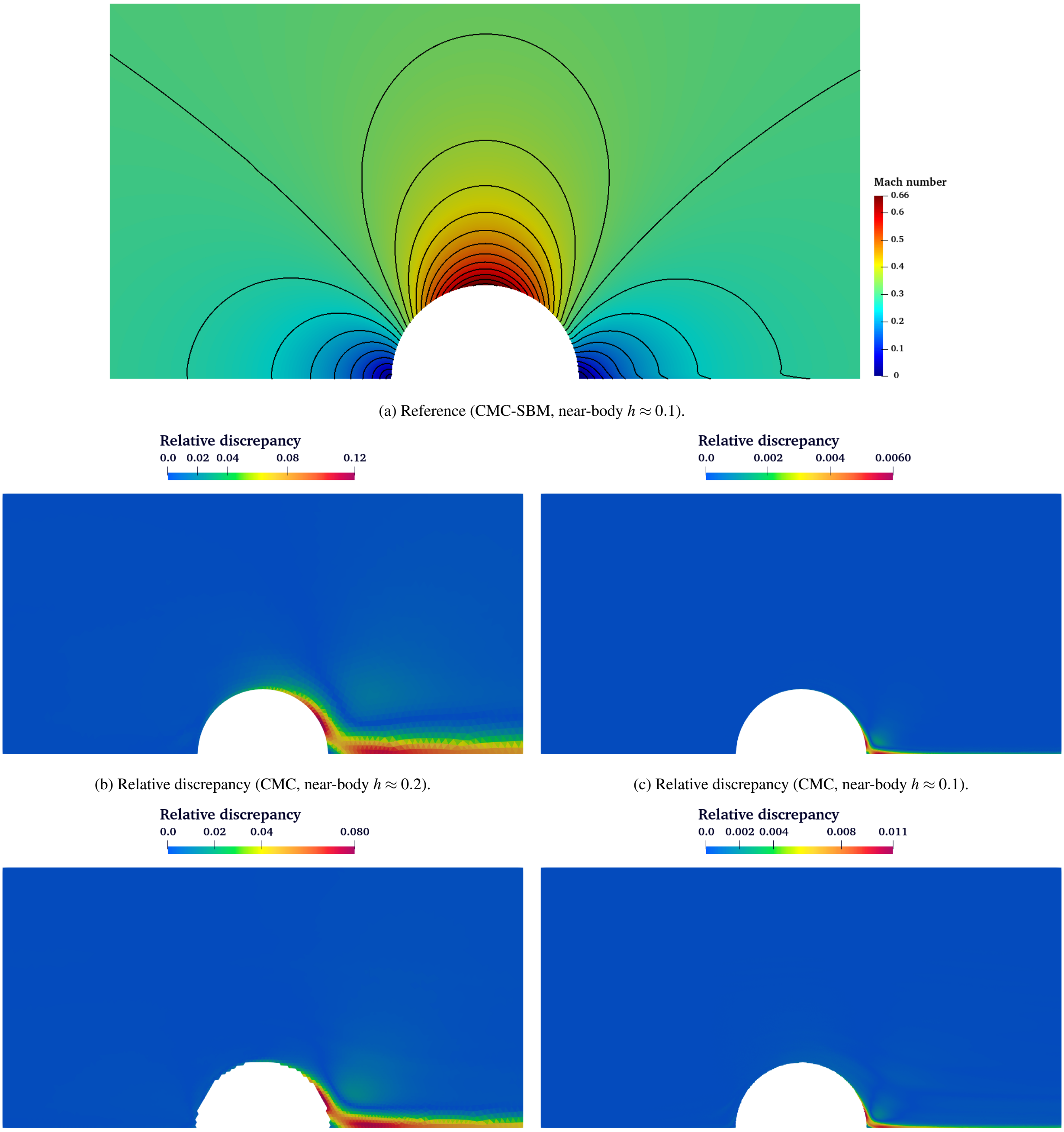
$$\frac{|M - M_{\text{ref}}|}{||M_{\text{ref}}||_\infty}\,. \tag{4.10}$$

Here M_{ref} is a resampling of the reference CMC-SBM solution on the grid of the CMC or SBM simulation under consideration, and M is the Mach number of the CMC or SBM simulation, interpolated at each cell centroid (see also the caption of Figure 4.9).

First of all, we see that due to the effect of artificial viscosity, the accuracy is largely determined by the mesh resolution, see the discussion before Remark 4.3. Second, the relative discrepancy on the fine grids are ten times smaller than that on the coarse grids. This, however, should not be treated as the actual numerical error as the reference solution itself is computed on the fine grid; nevertheless, it makes our point that the numerical error due to the SBM treatment at the embedded boundary is comparable to that of the conformal grid computation. A similar conclusion can be obtained by comparing the pressure coefficient

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty\,||\boldsymbol{u}_\infty||^2} \tag{4.11}$$

along the cylinder surface, shown in Figure 4.10. Note that C_p is computed at the nodes on the boundary for the CMC, while in the case of the SBM and the CMC-SBM a Taylor extrapolation of the pressure is evaluated at quadrature



(a) Reference (CMC-SBM, near-body $h \approx 0.1$).

(b) Relative discrepancy (CMC, near-body $h \approx 0.2$).

(c) Relative discrepancy (CMC, near-body $h \approx 0.1$).

(d) Relative discrepancy (SBM, near-body $h \approx 0.2$).

(e) Relative discrepancy (SBM, near-body $h \approx 0.1$).

Figure 4.9: Subsonic flow past a cylinder: contours of the Mach number and relative discrepancies. The reference solution (Figure 4.9a) is computed by the CMC-SBM using the fine mesh ($h \approx 0.1$ near the cylinder), whereas the relative discrepancy in all the other plots is computed with (4.10). The actual range of the Mach relative discrepancy in each test is given by the corresponding legend on top of the plot.

points on the surrogate boundary. Specifically, we extrapolate the pressure from a quadrature point $\tilde{\boldsymbol{x}}$ on the surrogate surface to its projection $\boldsymbol{M}^h(\tilde{\boldsymbol{x}})$ on the true boundary geometry according to

$$p(\boldsymbol{M}^h(\tilde{\boldsymbol{x}})) \approx p(\tilde{\boldsymbol{x}}) + (\boldsymbol{M}^h(\tilde{\boldsymbol{x}}) - \tilde{\boldsymbol{x}}) \cdot \nabla p(\tilde{\boldsymbol{x}}) \,, \tag{4.12}$$

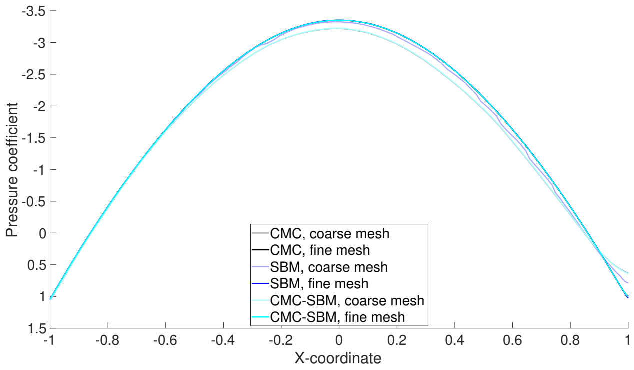
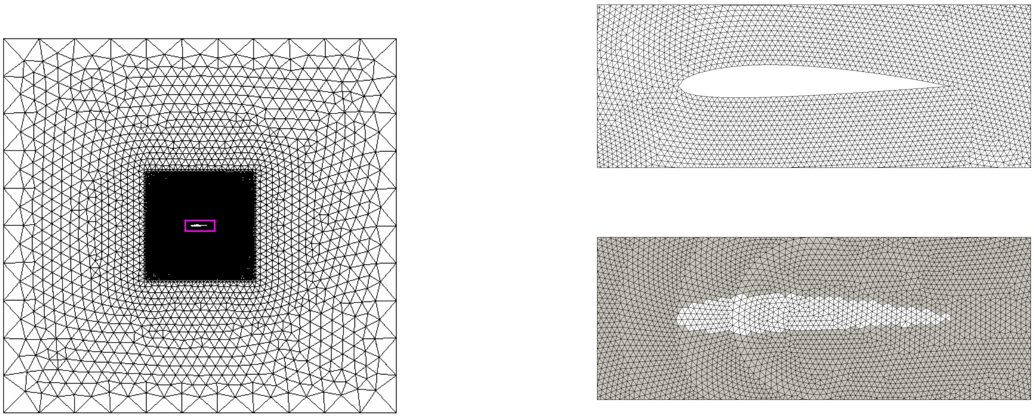


Figure 4.10: Subsonic flow past cylinder. The pressure coefficient along the cylinder surface computed by the CMC, the SBM, and the CMC-SBM. For the SBM and the CMC-SBM, pressures are extrapolated on the exact geometry from close-by surrogate boundaries, as discussed at the end of Section 4.4. Note that the curves of the CMC and the CMC-SBM are almost overlapped, for both grids.



(a) A grid for the SBM with $h = 0.02$ near the airfoil and $h = 2.0$ in the far field.

(b) Zoomed views of the CMC grid (above) and the SBM grid (below, with the dark-gray shaded region indicating the surrogate domain).

Figure 4.11: Transonic flow past a NACA 2412 airfoil: computational grids. Left panel: an embedded grid with $h = 0.02$ near the airfoil and $h = 2.0$ in the far-field Right panel: zoomed view (in the region $[-0.3,\, 1.3] \times [-0.3,\, 0.3]$ of the the magenta box in the left panel. On the top, a body-fitted grid used in the CMC simulations, On the bottom, the surrogate domain (dark gray) associated with the active elements of the SBM simulations. Details are presented in Table 5.

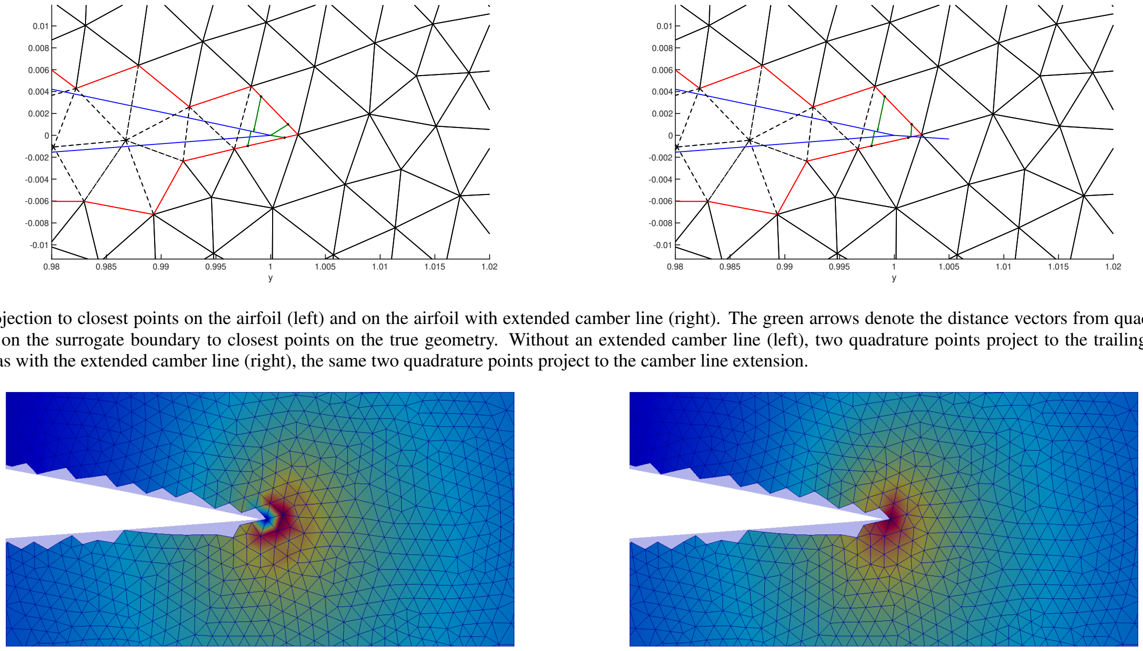
where the gradient is computed the same way as in Section 3.

4.5. *Transonic flow past a NACA 2412 airfoil*

To assess the numerical performance of the SBM in the transonic regime, we consider a flow at Mach 0.8 past a NACA 2412 airfoil at zero angle of attack. The computational domain is $\Omega = [-12,\, 12]^2$ and the airfoil chord lies along the segment $[0,\, 1] \times \{0\}$. The heat capacity ratio is $\gamma = 1.4$, and the initial condition at time $T = 0.0$ is given by

$$\rho_\infty = 1.4\,, \qquad \boldsymbol{u}_\infty = \begin{Bmatrix} 0.8 \\ 0.0 \end{Bmatrix}\,, \qquad p_\infty = 1\,, \tag{4.13}$$

which exactly corresponds to a Mach number of 0.8. The fluid state vector (4.13) is also used to enforce a far-field boundary condition at all four exterior edges; the airfoil surface is treated with a slip-wall boundary condition. Heterogeneous grids are employed for both the CMC and the SBM computations, with element size near the airfoil



(a) Projection to closest points on the airfoil (left) and on the airfoil with extended camber line (right). The green arrows denote the distance vectors from quadrature points on the surrogate boundary to closest points on the true geometry. Without an extended camber line (left), two quadrature points project to the trailing edge; whereas with the extended camber line (right), the same two quadrature points project to the camber line extension.

(b) Converged pressure near the trailing edge computed by the SBM without extended camber line (left) and with extended camber line (right).

Figure 4.12: Comparison between the SBM computations without (left) and with (right) enhancement for the Kutta condition. The upper row demonstrates the different computations of the projection to true geometry at the trailing edge and the lower row compares the pressure computed by the SBM at the trailing edge.

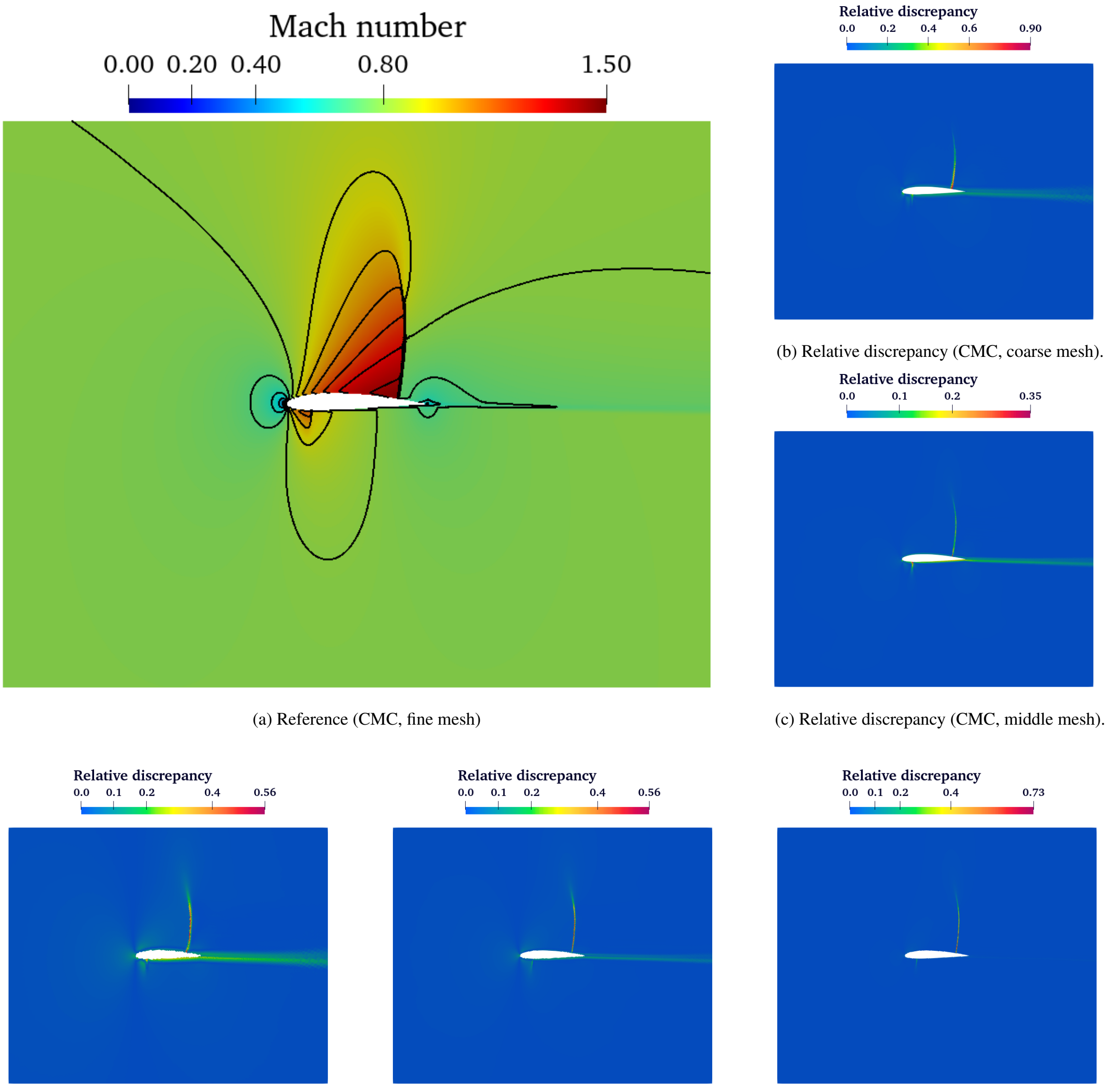
Table 5: Transonic flow past a NACA 2412 airfoil: mesh information, in terms of the number of nodes and number of elements in the computational domain for three pairs of grids.

	CMC Meshes		SBM Meshes	
	Nodes	Elements	Nodes	Elements
Mesh 1	32,607	65,071	32,935	65,716
Mesh 2	130,773	261,262	131,693	263,079
Mesh 3	527,067	1,053,567	526,672	1,052,732

about 100 times smaller than in the far field. We consider three pairs of grids (up to one million elements) with resolution $h = 0.02$, $h = 0.01$, and $h = 0.005$ near the airfoil, respectively. In Figure 4.11 we present sample grids with near-body resolution $h = 0.02$ used for the CMC and the SBM and the full grid information is summarized in Table 5.

When simulating the aerodynamics of airfoils, the Kutta condition at the trailing edge requires a matching of the velocity direction of the flows coming from the upper and lower surfaces, or equivalently, continuity of the pressure. Modern mesh generators can produce body-fitted grids that implicitly satisfy the Kutta condition, by aligning an edge of the grid with the camber line of the airfoil (H-grids, C-grids patterns). The SBM, instead, provides the flexibility of defining an "imaginary tail" at the trailing edge that extends the camber line, for the purpose of the enforcement of the Kutta condition. In particular, this strategy modifies how the projection, distance, and the normal vector are computed at quadrature points on a few edges of the surrogate boundary, as shown in Figure 4.12a. Note that the surrogate boundary $\widetilde{\Gamma}$ is still computed according to the airfoil geometry, and the extended camber line is only utilized to adjust distances and normal vectors. It is clear from the sample plots in Figure 4.12b, that this enhancement allows the SBM to compute a continuous pressure field at the trailing edge.

For simplicity, we adopt a brute force approach to find the projection of each quadrature point on the surrogate



(a) Reference (CMC, fine mesh)

(b) Relative discrepancy (CMC, coarse mesh).

(c) Relative discrepancy (CMC, middle mesh).

(d) Relative discrepancy (SBM, coarse mesh).

(e) Relative discrepancy (SBM, middle mesh).

(f) Relative discrepancy (SBM, fine mesh).

Figure 4.13: Transonic flow past a NACA 2412 airfoil: contours of Mach number and relative discrepancies. The reference solution (Figure 4.13a) is computed by the CMC on the fine mesh ($h \approx 0.005$ near the airfoil), whereas the relative discrepancy in all the other plots is computed with (4.10). The actual range of the Mach relative discrepancy in each test is given by the corresponding legend on top of the plot.

boundary to the airfoil surface, using 10^6 equally spaced sampling points along the chord on both the upper surface and the lower surface. To speed up computations, we set a larger tolerance to determine the steady state and stop the computation when $\max_A \left|\left|(\boldsymbol{U}_A^{n+1} - \boldsymbol{U}_A^n)/\Delta t^n\right|\right|_\infty \leq 10^{-6}$.

Figure 4.13, similar to the case of the flow past a cylinder, shows only one converged Mach number field (the CMC on the fine grid) as the reference solution in Figure 4.13a and for each of the other tests (the SBM on all three grids, and the CMC on the coarse and middle grids) we plot the discrepancy (4.10). The largest discrepancies, of order $O(1)$, occur near shock fronts, while everywhere else discrepancies are smaller, by orders of magnitude: this is expected, as slight changes in the shock location can produce large values of (4.10). Hence, we conclude that all simulations compute nearly the same shock positions, and the SBM and the CMC solutions are in close agreement on

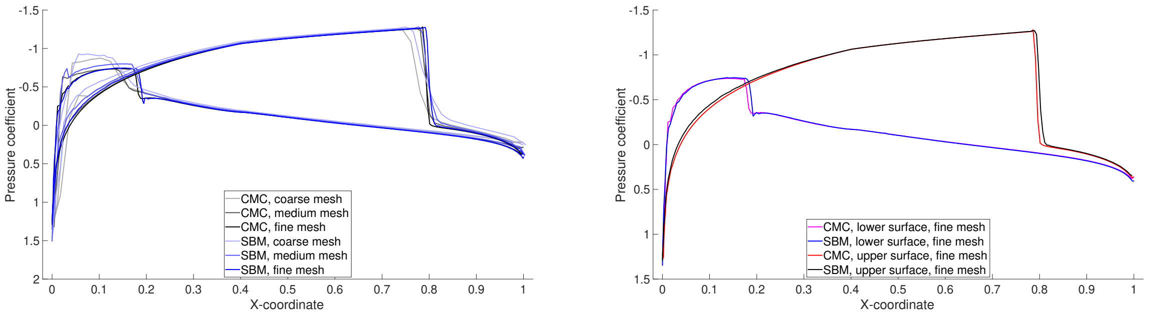


Figure 4.14: Transonic flow past a NACA 2412 airfoil. Left panel: pressure coefficient C_p along the airfoil surface computed by the CMC and the SBM on three pairs of grids; for the SBM, pressures are extrapolated on the exact geometry from close-by surrogate boundaries, as discussed at the end of Section 4.4. Right panel: pressure coefficient computed by the CMC and the SBM on the pair of fine grids, with upper surface and lower surface distinguished by different colors.

Table 6: Aerodynamic coefficients computed by the CMC and the SBM on the two sequences of grids.

	Drag coefficient C_d		
	Coarse mesh	Intermediate mesh	Fine mesh
CMC	0.059243574336755	0.049691976657056	0.043671104264331
SBM	0.118622899641064	0.038597677782040	0.052549780911754
	Lift coefficient C_l		
	Coarse mesh	Intermediate mesh	Fine mesh
CMC	0.526959085624866	0.544370285666716	0.537545645173485
SBM	0.595367964111681	0.548275555352769	0.552622246472996

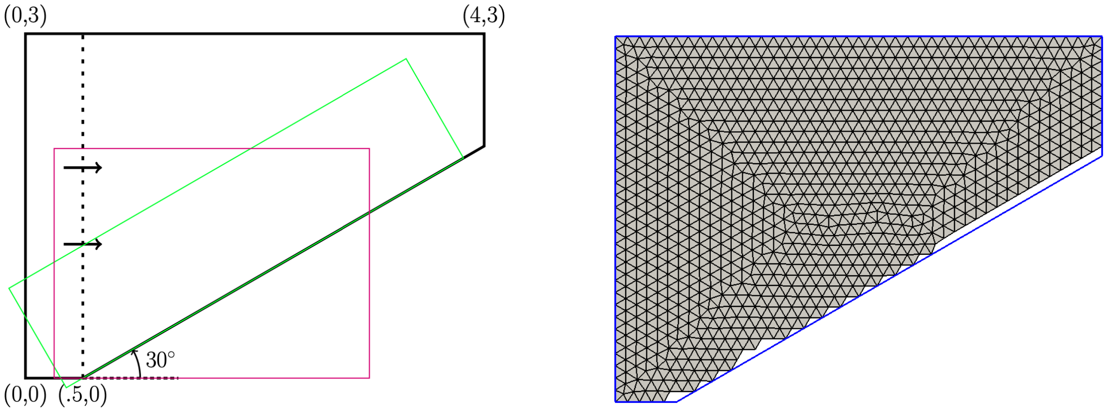
grids with similar resolution. The agreement between the CMC and the SBM can be seen more easily in Figure 4.14, which shows a comparison of the pressure coefficients along the chord of the airfoil, again for the three pairs of grids. The pressure coefficients are calculated according to (4.11) for the CMC and (4.12) for the SBM, respectively.

The drag and lift coefficients of the airfoil are shown in Table 6, and have been computed according to the following formulas, for the CMC and the SBM.

$$\text{CMC}: \quad C_d = \frac{\int_\Gamma p\boldsymbol{n}\cdot\boldsymbol{e}_x d\Gamma}{\frac{1}{2}\rho_\infty\,||\boldsymbol{u}_\infty||^2}, \qquad C_l = \frac{\int_\Gamma p\boldsymbol{n}\cdot\boldsymbol{e}_y d\Gamma}{\frac{1}{2}\rho_\infty\,||\boldsymbol{u}_\infty||^2}; \tag{4.14}$$

$$\text{SBM}: \quad C_d = \frac{\int_{\tilde\Gamma}(p+\boldsymbol{d}\cdot\nabla p)(\boldsymbol{n}\cdot\tilde{\boldsymbol{n}})\boldsymbol{n}\cdot\boldsymbol{e}_x d\tilde\Gamma}{\frac{1}{2}\rho_\infty\,||\boldsymbol{u}_\infty||^2}, \qquad C_l = \frac{\int_{\tilde\Gamma}(p+\boldsymbol{d}\cdot\nabla p)(\boldsymbol{n}\cdot\tilde{\boldsymbol{n}})\boldsymbol{n}\cdot\boldsymbol{e}_y d\tilde\Gamma}{\frac{1}{2}\rho_\infty\,||\boldsymbol{u}_\infty||^2}. \tag{4.15}$$

These coefficients also show good agreement on grids of comparable resolutions, especially on the medium and fine pairs. It is important to realize that the grids utilized in these simulations are most likely under-resolved, and very precise conclusions about the ultimate accuracy of the SBM versus the CMC simulations are hard to make. However, it is fair to say that the SBM results are close to the CMC results, and are predictive. Observe also that the SBM simulation on the finest grid produces slightly sharper shocks than the corresponding CMC simulation (see the right panel of Figure 4.14), an aspect that warrants further investigation.



(a) DMR problem setup.

(b) A sample embedded grid with element size $h = 0.1$.

Figure 4.15: The DMR problem configuration. Left panel: the computational domain with its green boundary, and the viewing/plotting box, with its magenta boundary. Right panel: an embedded domain with reference element length $h = 0.1$.

4.6. Double Mach reflection problem

We consider here the double Mach reflection (DMR) problem, a benchmark test with very strong interacting shocks, first proposed in [80] and extensively used thereafter in the computational fluid dynamics literature. The problem considers a vertical Mach 10 shock pushing along a 30-degree wedge (see Figure 4.15). Traditionally, a rectangular computational grid is aligned with the wedge ramp so that Cartesian grids can easily be deployed (see the green box of Figure 4.15a). In our case, we consider instead triangularizations of the pentagonal domain and the bounding rectangular domain in Figure 4.15a, for the CMC and the SBM computations respectively.

The specific heat capacity ratio is $\gamma = 1.4$ and the initial condition is given by two constant fluid states separated by a Mach 10 shock along the line $x = 0.5$ (the vertical dashed line in Figure 4.15a), To the left of the shock (i.e., $x < 0.5$), the initial data is

$$\rho = 8\,, \quad \boldsymbol{u} = \begin{Bmatrix} 8.25 \\ 0.0 \end{Bmatrix}, \quad p = 116.5\,, \tag{4.16}$$

and to the right of the shock ($x > 0.5$), the initial data is

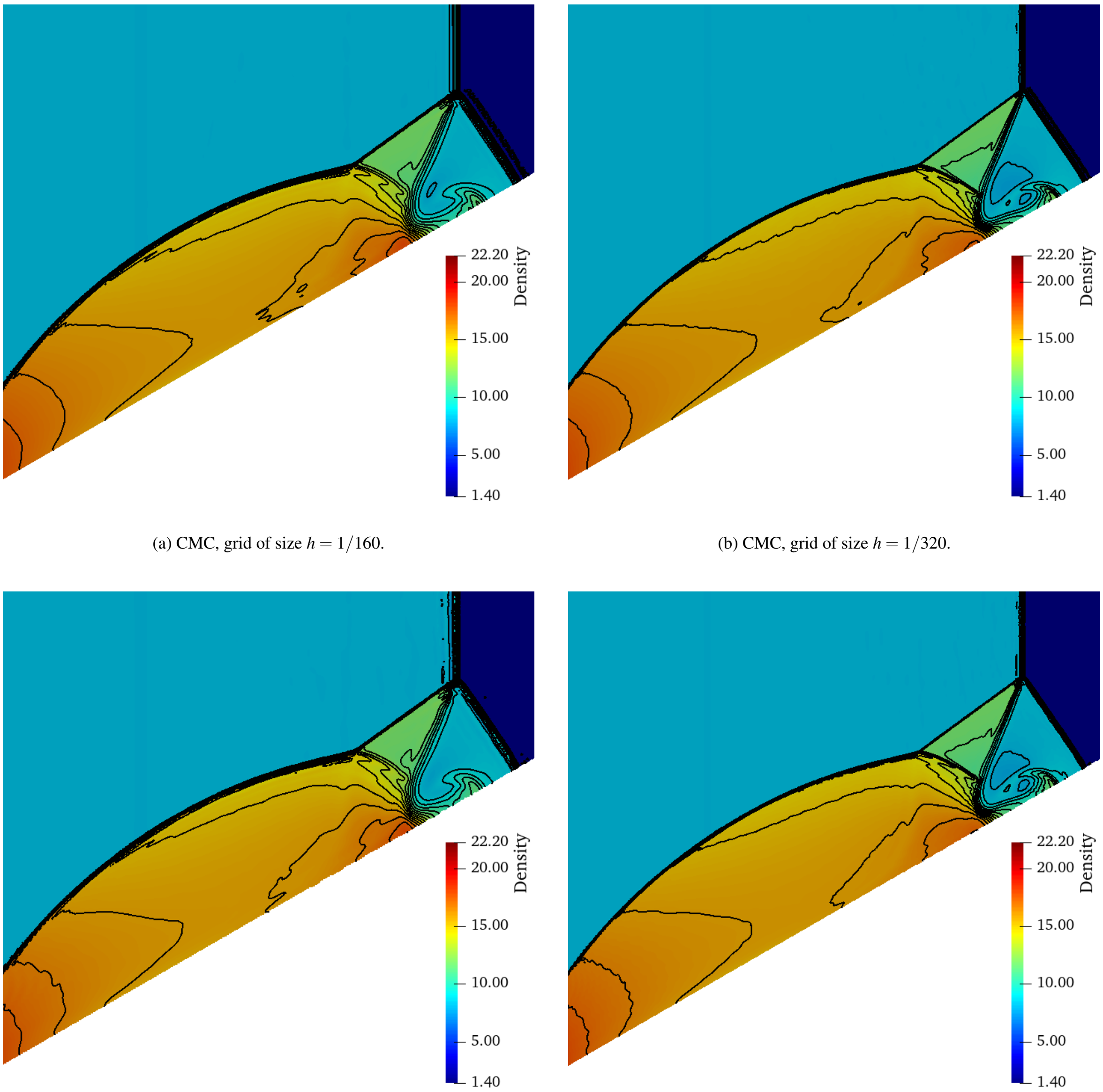
$$\rho = 1.4\,, \quad \boldsymbol{u} = \begin{Bmatrix} 0.0 \\ 0.0 \end{Bmatrix}, \quad p = 1.0\,. \tag{4.17}$$

Slip-wall boundary conditions are applied at the lower edge, the upper edge, and the ramp. Because the state (4.16) is supersonic, we enforce a supersonic inlet boundary condition at the left edge. The problem is computed until $T = 0.2$, at which the shock has yet to reach the right boundary. Hence, a slip boundary condition is also enforced at the edge along $x = 4$.

For this test, we use the Lax viscosity with Ducros-type sensor as described in Section 2.4.2 with parameter $c_{\text{Lax}} = 0.1$. While the DMR test is widely used in high-order computations [81], our focus is to assess the impact of the shifted boundary condition on shocks oblique to the boundary. In other words, instead of attempting to resolve small scale flow features in the reflection region, we compare the density contour plots obtained by the CMC and the SBM on grids with similar resolution, as shown in Figure 4.16. The plots show that the SBM produces very similar results to the CMC.

4.7. Flow past a complex 3D object

Lastly, we compute the supersonic flow past a “Monkey Trefoil”, that is the complex three-dimensional object shown in Figure 4.17 and described in detail in [82]. This simulation demonstrates the versatility of the proposed SBM, since the geometry is provided in STL (Standard Tessellation Language) format, with tiny gaps and overlaps. The



(a) CMC, grid of size $h = 1/160$.

(b) CMC, grid of size $h = 1/320$.

(c) SBM, grid of size $h = 1/160$.

(d) SBM, grid of size $h = 1/320$.

Figure 4.16: DMR problem. Density contours for the CMC (upper row) and the SBM (lower row) on grids with representative element size $h = 1/160$ (left column) and $h = 1/320$ (right column). A total of 30 contours in the range $[1.4, 22.2]$ is plotted. The view region is indicated by the magenta box in Figure 4.15a.

STL format represents surfaces in the three-dimensional Euclidean space as a collection of disconnected (oriented) triangles. The computational domain is the channel $\Omega = [-4, 28] \times [-2, 2] \times [-4, 4]$, with the center of the Monkey Trefoil positioned at the origin of the axes.

Slip-wall boundary conditions are imposed on the four sides of the channel that are parallel to the x-axis. At the remaining sides of the channel domain (i.e., the inflow and the outflow) boundary fluxes are computed according to

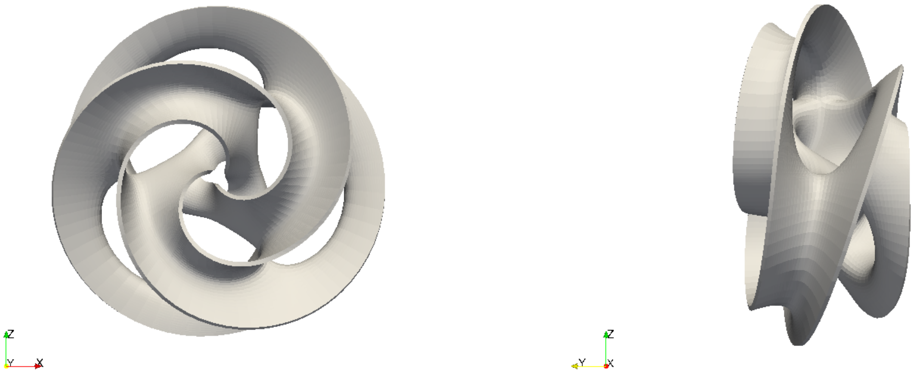


Figure 4.17: The “Monkey Trefoil” object with different viewpoints: (left) into the positive y-direction, and (right) into the positive x-direction. The bounding box of the object is approximately $6.23 \times 3.02 \times 6.29$.

the Roe flux formula (2.18) with an ambient fluid state at Mach 1.2 flow. Namely:

$$\rho_\infty = 1.4, \quad \boldsymbol{u}_\infty = \left\{ \begin{matrix} 1.2 \\ 0.0 \\ 0.0 \end{matrix} \right\}, \quad p_\infty = 1\ . \tag{4.18}$$

This state is also used to set the initial condition of the fluid throughout the domain. Immersed, shifted, slip-wall boundary conditions are enforced on the surface of the Monkey Trefoil shape.

For this three-dimensional test, we adopt stronger penalty at the conformal and embedded slippery boundaries by setting $\alpha_{\text{slip}} = 2.0$. Two tetrahedral grids are used to perform the SBM computation with reference element lengths $h = 0.1$ and $h = 0.05$ near the embedded geometry, respectively. Particularly, the coarser grid has $2,895,945$ elements and $512,925$ nodes, and the finer one has $14,162,737$ elements and $2,478,383$ nodes. In Figure 5.1, Figure 5.2, and Figure 5.3, we plot the pressure contours at time $T = 3$, 6, and 12. In Figure 5.4, Figure 5.5, and Figure 5.6, we plot the Mach number contours, for the same time instants. From these pictures, we can clearly observe that the solution is smooth and free from spurious oscillations, and that gradients become sharper as the grid is refined, an indication that the solution is well behaved, despite we do no have numerical or experimental references to compare with.

5. Conclusions

We have constructed a Shifted Boundary Method for (inviscid) compressible Euler equations on immersed grids. We have investigated the interplay between the SBM and both hyperbolic wave structures and strong shocks, near embedded boundaries. The SBM was compared to conforming mesh computations in extensive numerical tests and found to yield very similar results for comparable mesh resolutions. Similarly, it was found that the SBM is not affected by introduction of a discontinuity-capturing operator, in the form of an artificial viscosity.

We also demonstrated that: (a) in the case of body-fitted computations, the SBM can improve the accuracy at curved boundaries approximated by polygonal interpolation; and (b) in immersed computations of airfoils, the SBM can provide a simple and robust implementation of the Kutta condition at the trailing edge.

Future directions include optimizing the location of surrogate boundaries with the objective to reducing the numerical error. We also plan to extend the method to compressible flow problems with moving boundaries and fluid-structure interaction.

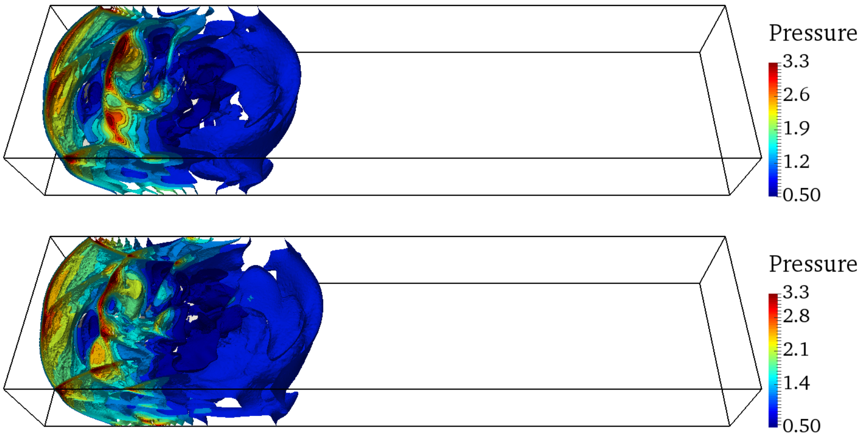


Figure 5.1: Monkey Trefoil test. Pressure contours at $T = 3$, computed with the SBM on a coarser grid (top) and a finer grid (bottom).

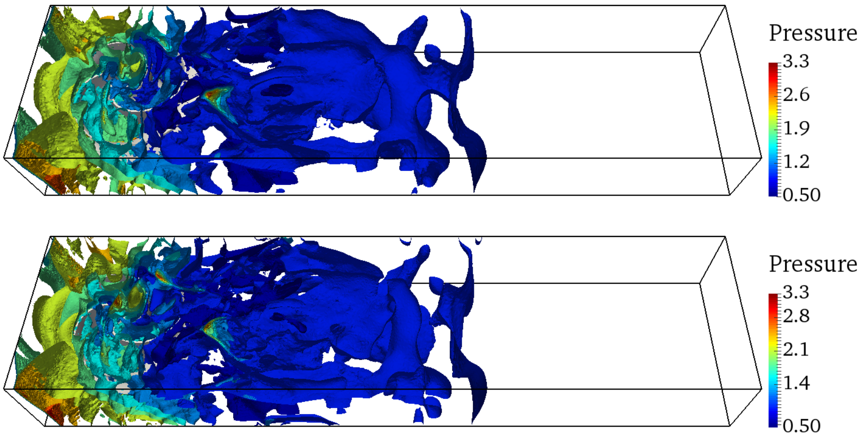


Figure 5.2: Monkey Trefoil test. Pressure contours at $T = 6$, computed with the SBM on a coarser grid (top) and a finer grid (bottom).

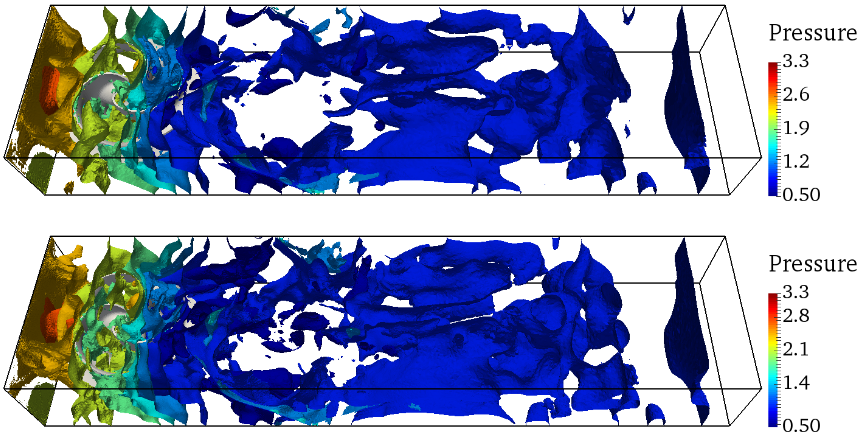


Figure 5.3: Monkey Trefoil test. Pressure contours at $T = 12$, computed with the SBM on a coarser grid (top) and a finer grid (bottom).

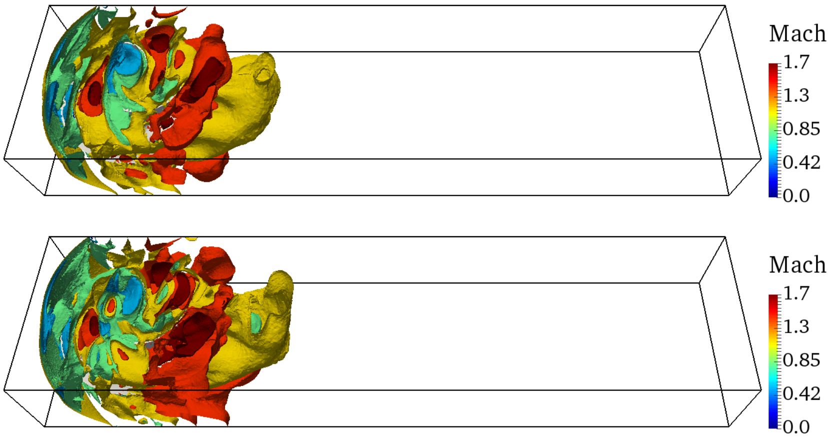


Figure 5.4: Monkey Trefoil test. Mach number contours at $T = 3$, computed with the SBM on a coarser grid (top) and a finer grid (bottom).

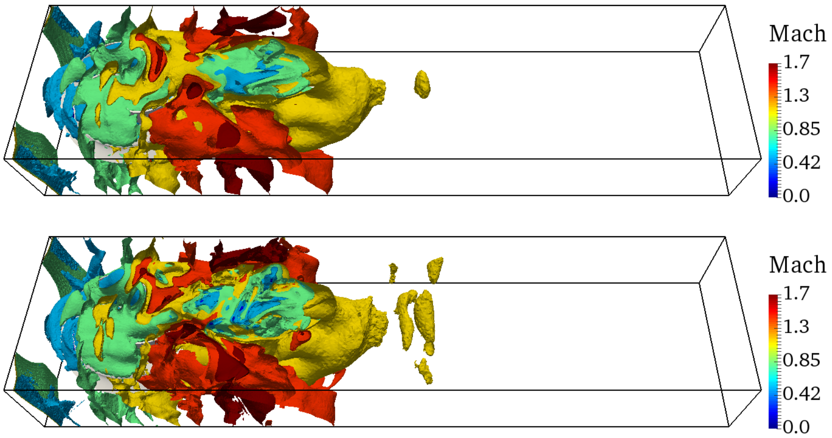


Figure 5.5: Monkey Trefoil test. Mach number contours at $T = 6$, computed with the SBM on a coarser grid (top) and a finer grid (bottom).

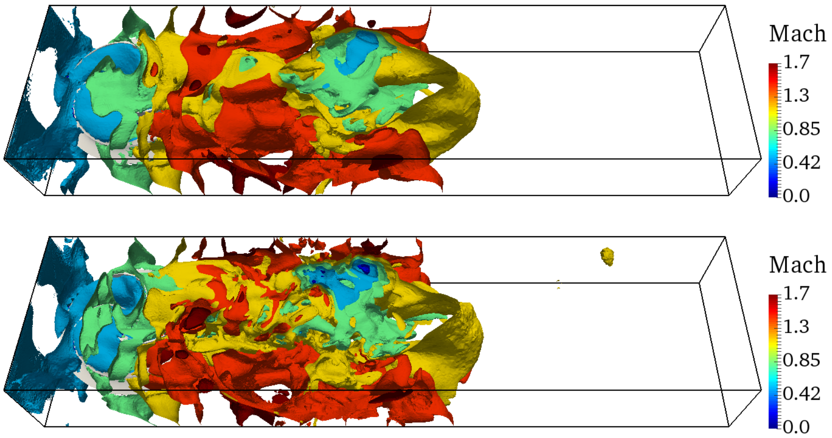


Figure 5.6: Monkey Trefoil test. Mach number contours at $T = 12$, computed with the SBM on a coarser grid (top) and a finer grid (bottom).

Acknowledgements

G. Scovazzi is currently supported by the U.S. National Science Foundation under Grants DMS-2207164 and DMS-2409919 (Computational Mathematics Program, in the Division of Mathematical Sciences). Xianyi Zeng is supported by the U.S. National Science Foundation under Grant DMS-2302080 (Division of Mathematical Sciences). Some of the computations were performed on the "Nivolet" computer cluster at Duke University (owned by G. Scovazzi and sponsored by the Army Research Office through DURIP Grant W911NF1510382). All these sources of funding are gratefully acknowledged.

The authors would also like to thank Prof. Andrea Ferrero, Prof. Francesco La Rocca, and Dr. Michele Errante, all at Politecnico di Torino, for fruitful conversations and suggestions.

References

[1] C. S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (3) (1977) 220–252.

[2] C. S. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.

[3] F. Sotiropoulos, X. Yang, Immersed boundary methods for simulating fluid–structure interaction, Progress in Aerospace Sciences 65 (2014) 1–21.

[4] J. E. Melton, M. J. Berger, M. J. Aftosmis, M. D. Wong, 3d applications of a Cartesian grid euler method, in: 33rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, United States, 1995.

[5] G. Yang, D. Causon, D. Ingram, R. Saunders, P. Battent, A cartesian cut cell method for compressible flows part A: Static body problems, The Aeronautical Journal 101 (1002) (1997) 47–56.

[6] P. G. Tucker, Z. Pan, A cartesian cut cell method for incompressible viscous flow, Applied Mathematical Modelling 24 (8-9) (2000) 591–606.

[7] S. M. Murman, M. J. Aftosmis, M. J. Berger, Implicit approaches for moving boundaries in a 3-D Cartesian method, in: 41th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 2003.

[8] P. Colella, D. T. Graves, B. J. Keen, D. Modiano, A Cartesian grid embedded boundary method for hyperbolic conservation laws, J. Comput. Phys. 211 (1) (2006) 347–366, https://doi.org/10.1016/j.jcp.2005.05.026.

[9] E. A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (1) (2000) 35–60.

[10] 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.

[11] C. Chi, B. J. Lee, H. G. Im, An improved ghost-cell immersed boundary method for compressible flow simulations, Int. J. Num. Meth. Fluids 83 (2) (2017) 132–148.

[12] H.-O. Kreiss, N. A. Petersson, J. Yström, Difference approximations for the second order wave equation, SIAM J. Numer. Anal. 40 (5) (2002) 1940–1967.

[13] M. Kupiainen, B. Sjögreen, A cartesian embedded boundary method for the compressible Navier-Stokes equations, J. Sci. Comput. 41 (2009) 94–117.

[14] K. Nakahashi, Immersed boundary method for compressible Euler equations in the building-cube method, in: 20th AIAA Computational Fluid Dynamics Conference, 2011, p. 3386.

[15] J. W. Banks, D. W. Schwendeman, A. K. Kapila, W. D. Henshaw, A high-resolution godunov method for compressible multi-material flow on overlapping grids, J. Comput. Phys. 223 (1) (2007) 262–297.

[16] J. W. Banks, W. D. Henshaw, J. N. Shadid, An evaluation of the fct method for high-speed flows on structured overlapping grids, J. Comput. Phys. 228 (15) (2009) 5349–5369.

[17] J. L. Steger, F. C. Dougherty, J. A. Benek, A chimera grid scheme, NASA Technical Reports (1983).
URL `https://ntrs.nasa.gov/citations/19840028795`

[18] J. A. Benek, P. G. Buning, J. L. Steger, A 3-D chimera grid embeddeing technique, in: 7th AIAA Computational Physics Conference, Cincinnati, OH, 1985.

[19] J. A. Benek, J. L. Steger, F. C. Dougherty, P. G. Buning, Chimera: A grid-embedding scheme, Tech. Rep. AEDC-TR-85-64, Arnold Engineering Development Center (1986).

[20] D. Jespersen, T. Pulliam, P. Buning, Recent enhancements to OVERFLOW, in: 35th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 1997.

[21] X. Zeng, C. Farhat, A systematic approach for constructing higher-order immersed boundary and ghost fluid methods for fluid–structure interaction problems, J. Comput. Phys. 231 (7) (2012) 2892–2923.

[22] X. Zeng, C. Farhat, A systematic procedure for achieving higher-order spatial accuracy in ghost fluid and other embedded boundary methods for fluid-structure interaction problems, in: 20th AIAA Computational Fluid Dynamics Conference, 2011, p. 3389.

[23] A. Main, X. Zeng, P. Avery, C. Farhat, An enhanced fiver method for multi-material flow problems with second-order convergence rate, J. Comput. Phys. 329 (2017) 141–172.

[24] A. V. Kumar, S. Padmanabhan, R. Burla, Implicit boundary method for finite element analysis using non-conforming mesh or grid, Int. J. Numer. Meth. Eng. 74 (9) (2008) 1421–1447.

[25] A. Obeidat, S. P. Bordas, An implicit boundary approach for viscous compressible high Reynolds flows using a hybrid remeshed particle hydrodynamics method, J. Comput. Phys. 391 (2019) 347–364.

[26] P. Fu, T. Frachon, G. Kreiss, S. Zahedi, High order discontinuous cut finite element methods for linear hyperbolic conservation laws with an interface, J. Sci. Comput. 90 (3) (2022) 84.

[27] R. Zorrilla, A. Larese de Tetto, R. Rossi, A discontinuous Nitsche-based finite element formulation for the imposition of the Navier-slip condition over embedded volumeless geometries, Int. J. Num. Meth. Fluids 93 (9) (2021) 2968–3003.

[28] M. Natarajan, R. Grout, W. Zhang, M. Day, A moving embedded boundary approach for the compressible Navier-Stokes equations in a block-structured adaptive refinement framework, J. Comput. Phys. 465 (2022) 111315.

[29] E. Burman, Ghost penalty, Comptes Rendus Mathematique 348 (21-22) (2010) 1217–1220.

[30] K. Höllig, Finite element methods with B-splines, Vol. 26, SIAM, 2003.

[31] K. Höllig, U. Reif, J. Wipper, Weighted extended B-spline approximation of Dirichlet problems, SIAM J. Numer. Anal. 39 (2) (2001) 442–462.

[32] T. Rüberg, F. Cirak, Subdivision-stabilised immersed b-spline finite elements for moving boundary flows, Comput. Methods Appl. Mech. Eng. 209 (2012) 266–283.

[33] T. Rüberg, F. Cirak, A fixed-grid b-spline finite element technique for fluid–structure interaction, Int. J. Num. Meth. Fluids 74 (9) (2014) 623–660.

[34] J. Parvizian, A. Düster, E. Rank, Finite cell method, Computational Mechanics 41 (1) (2007) 121–133.

[35] A. Düster, J. Parvizian, Z. Yang, E. Rank, The finite cell method for three-dimensional problems of solid mechanics, Comput. Methods Appl. Mech. Eng. 197 (45-48) (2008) 3768–3782.

[36] 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.

[37] A. Main, G. Scovazzi, The Shifted Boundary Method for embedded domain computations. Part II: Linear advection-diffusion and incompressible Navier-Stokes equations, J. Comput. Phys. 372 (2018) 996–1026.

[38] K. Li, N. M. Atallah, G. A. Main, G. Scovazzi, The shifted interface method: A flexible approach to embedded interface computations, Int. J. Numer. Meth. Eng. 121 (3) (2020) 492–518.

[39] E. N. Karatzas, G. Stabile, N. Atallah, G. Scovazzi, G. Rozza, A reduced order approach for the embedded Shifted Boundary FEM and a heat exchange system on parametrized geometries, in: IUTAM Symposium on Model Order Reduction of Coupled Systems, Stuttgart, Germany, May 22–25, 2018, Springer, 2020, pp. 111–125.

[40] E. N. Karatzas, G. Stabile, L. Nouveau, G. Scovazzi, G. Rozza, A reduced basis approach for PDEs on parametrized geometries based on the Shifted Boundary Finite Element Method and application to a Stokes flow, Comput. Methods Appl. Mech. Eng. 347 (2019) 568–587.

[41] E. N. Karatzas, G. Stabile, L. Nouveau, G. Scovazzi, G. Rozza, A Reduced-order Shifted Boundary Method for parametrized incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Eng. 370 (2020) 113273.

[42] N. M. Atallah, C. Canuto, G. Scovazzi, Analysis of the Shifted Boundary Method for the Poisson problem in domains with corners, Math. Comput. 90 (2021) 2041–2069.

[43] N. M. Atallah, C. Canuto, G. Scovazzi, The second-generation Shifted Boundary Method and its numerical analysis, Comput. Methods Appl. Mech. Eng. 372 (2020) 113341.

[44] N. M. Atallah, C. Canuto, G. Scovazzi, The high-order Shifted Boundary Method and its analysis, Comput. Methods Appl. Mech. Eng. 394 (2022) 114885.

[45] J. H. Collins, A. Lozinski, G. Scovazzi, A penalty-free Shifted Boundary Method of arbitrary order, Comput. Methods Appl. Mech. Eng. (2023) 116301.

[46] N. M. Atallah, G. S. Claudio Canuto, The Shifted Boundary Method for solid mechanics, Int. J. Numer. Meth. Eng. 122 (20) (2021) 5935–5970.

[47] N. M. Atallah, G. Scovazzi, Nonlinear elasticity with the Shifted Boundary Method, Computer Methods in Applied Mechanics and Engineering 426 (2024) 116988.

[48] N. Antonelli, R. Aristio, A. Gorgi, R. Zorrilla, R. Rossi, G. Scovazzi, R. Wüchner, The Shifted Boundary Method in isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 430 (2024) 117228.

[49] T. Song, A. Main, G. Scovazzi, M. Ricchiuto, The Shifted Boundary Method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, J. Comput. Phys. 369 (2018) 45–79.

[50] G. Scovazzi, Lagrangian shock hydrodynamics on tetrahedral meshes: A stable and accurate variational multiscale approach, J. Comput. Phys. 231 (24) (2012) 8029–8069.

[51] X. Zeng, G. Scovazzi, A variational multiscale finite element method for monolithic ALE computations of shock hydrodynamics using nodal elements, J. Comput. Phys. 315 (2016) 577–608.

[52] C. Johnson, A. Szepessy, P. Hansbo, On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws, Math. Comput. 54 (189) (1990) 107–129.

[53] J.-L. Guermond, R. Pasquetti, B. Popov, Entropy viscosity method for nonlinear conservation laws, J. Comput. Phys. 230 (11) (2011) 4248–4267.

[54] V. Zingan, J.-L. Guermond, J. Morel, B. Popov, Implementation of the entropy viscosity method with the discontinuous galerkin method, Comput. Methods Appl. Mech. Eng. 253 (2013) 479–490.

[55] F. Ducros, V. Ferrand, F. Nicoud, C. Weber, D. Darracq, C. Gacherieu, T. Poinsot, Large-eddy simulation of the shock/turbulence interaction, J. Comput. Phys. 152 (2) (1999) 517–549.

[56] T. Hendrickson, A. Kartha, G. V. Candler, An improved Ducros sensor for the simulation of compressible flows with shocks, in: Fluid Dynamics Conference, Atlanta, Georgia, 2018.

[57] C. Farhat, Personal communication (2019).

[58] L. J. Ayton, J. R. Gill, N. Peake, The importance of the unsteady Kutta condition when modelling gust-aerofoil interaction, J. Sound Vib. 378 (2016) 28–37.

[59] E. F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics : A Practical Introduction, 3rd Edition, Springer Berlin Heidelberg, 2009.

[60] H. Deconinck, M. Ricchiuto, Residual distribution schemes: Foundation and analysis, Encyclopedia of Computational Mechanics (2007).

[61] T. Song, G. Scovazzi, A Nitsche method for wave propagation problems in time domain, Comput. Methods Appl. Mech. Eng. 293 (2015)

481–521.

[62] P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (2) (1981) 357–372.

[63] T. J. R. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, Comput. Methods Appl. Mech. Eng. 127 (1–4) (1995) 387–401.

[64] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method – a paradigm for computational mechanics, Comput. Methods Appl. Mech. Eng. 166 (1–2) (1998) 3–24.

[65] T. J. Hughes, G. Scovazzi, L. P. Franca, Multiscale and stabilized methods, Encyclopedia of Computational Mechanics (2018) 1–64.

[66] Y. Bazilevs, V. Calo, J. Cottrell, T. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Comput. Methods Appl. Mech. Eng. 197 (2007) 173–201.

[67] T. J. R. Hughes, L. P. Franca, M. Mallet, A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics, Comput. Methods Appl. Mech. Eng. 54 (1986) 223–234.

[68] T. J. R. Hughes, M. Mallet, A. Mizukami, A new finite element formulation for computational fluid dynamics: II. Beyond SUPG, Comput. Methods Appl. Mech. Eng. 54 (1986) 341–355.

[69] T. J. R. Hughes, M. Mallet, A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems, Comput. Methods Appl. Mech. Eng. 58 (1986) 305–328.

[70] T. J. R. Hughes, T. E. Tezduyar, Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations, Comput. Methods Appl. Mech. Eng. 45 (1984) 217–284.

[71] G. Scovazzi, M. A. Christon, T. J. R. Hughes, J. N. Shadid, Stabilized shock hydrodynamics: I. A Lagrangian method, Comput. Methods Appl. Mech. Eng. 196 (4–6) (2007) 923–966.

[72] G. Scovazzi, Stabilized shock hydrodynamics: II. Design and physical interpretation of the SUPG operator for Lagrangian computations, Comput. Methods Appl. Mech. Eng. 196 (4–6) (2007) 966–978.

[73] G. Scovazzi, J. N. Shadid, E. Love, W. J. Rider, A conservative nodal variational multiscale method for Lagrangian shock hydrodynamics, Comput. Methods Appl. Mech. Eng. 199 (49–52) (2010) 3059–3100.

[74] G. Scovazzi, B. Carnes, X. Zeng, S. Rossi, A simple, stable, and accurate linear tetrahedral finite element for transient, nearly, and fully incompressible solid dynamics: A dynamic variational multiscale approach, Int. J. Numer. Meth. Eng. 106 (10) (2016) 799–839.

[75] G. Scovazzi, A discourse on Galilean invariance and SUPG-type stabilization, Comput. Methods Appl. Mech. Eng. 196 (4–6) (2007) 1108–1132.

[76] G. Scovazzi, Galilean invariance and stabilized methods for compressible flows, Int. J. Num. Meth. Fluids 54 (6–8) (2007) 757–778.

[77] G. Scovazzi, E. Love, A generalized view on Galilean invariance in stabilized compressible flow computations, Int. J. Num. Meth. Fluids 64 (10–12) (2010) 1065–1083.

[78] L. Krivodonova, M. Berger, High-order accurate implementation of solid wall boundary conditions in curved geometries, J. Comput. Phys. 211 (2) (2006) 492–512.

[79] M. Ciallella, E. Gaburro, M. L. M. Ricchiuto, Shifted boundary polynomial corrections for compressible flows: High order on curved domains using linear meshes, Appl. Math. Comput. 441 (2023) 127698.

[80] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1) (1984) 115–173, review article.

[81] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1) (1996) 202–228.

[82] C. Séquin, Part description and specifications for monkey trefoil, `https://people.eecs.berkeley.edu/~sequin/SFF/spec.monktref.html`.