ICCFD13-2026-031

Genuinely multi-dimensional equilibria preservation on curved domains with a ghost point method for acoustic equations

Mirco Ciallella*, Davide Torlo⊥, Mario Ricchiuto×
Corresponding author: davide.torlo@uniroma1.it
* Université Paris Cité and Sorbonne Université, CNRS, LJLL, F-75005 Paris, France.
⊥ Dipartimento di Matematica, Università di Roma La Sapienza, Rome, Italy.
× Centre Inria de l'Université de Bordeaux, CNRS UMR 5251, Talence, France.

Abstract: In this work we present a novel stationarity preserving numerical method for the solution of hyperbolic problems, with a focus on the linear hyperbolic acoustic system. The method combines multi-dimensional globalization of the fluxes (GF) on Cartesian grids with the ghost point method (GPM) to handle curved domains. The main feature of the method is the embed discrete stationary solutions with zero velocity divergence, which are not preserved by classical methods. The features of the new method are shown on some classical examples, and compared to a standard Rusanov-GPM method that does not preserve the stationary solutions.

Keywords: Multi-dimensional equilibrium, Curved domains, Ghost Point Method, Global Flux.

Contents

1. Introduction to the problem of stationarity preservation for acoustic equations

We focus on the numerical solution of hyperbolic balance laws in two dimensions. Given a domain $\Omega = [x_\ell, x_r] \times [y_b, y_t] \subset \mathbb{R}^2$, we look for the solution $q : \mathbb{R}^+ \times \Omega \to \mathbb{R}^m$ of the following hyperbolic system of partial differential equations:

$$\partial_t q(t,x,y) + \partial_x f(q(t,x,y)) + \partial_y g(q(t,x,y)) = S(q(t,x,y),t,x,y), \tag{1}$$

where $f : \mathbb{R}^m \to \mathbb{R}^m$ and $g : \mathbb{R}^m \to \mathbb{R}^m$ are the flux functions, while $S : \mathbb{R}^m \times \mathbb{R}^+ \times \Omega \to \mathbb{R}^m$ is a source term. Examples of such systems include the Euler equations of gas dynamics, the shallow water equations and the magnetohydrodynamics equations. In this work, we focus on a liner version of (1): the acoustic equations. For this system $q = (p,u,v)$, where p is the pressure and u, v are the velocity components, while in the simplest setting the fluxes and source term can be written as

$$f(q) = \begin{pmatrix} u \\ p \\ 0 \end{pmatrix}, \qquad g(q) = \begin{pmatrix} v \\ 0 \\ p \end{pmatrix}, \qquad S(q,t,x,y) = 0. \tag{2}$$

Though being a linear system, the acoustic equations present some challenges for numerical methods. In particular, they admit a large class of stationary solutions, which are characterized by the conditions

$$\partial_x u + \partial_y v = 0, \qquad \nabla p = 0. \tag{3}$$

These stationary solutions are related to several practical applications going from the low Mach limit of the Euler equations, to several physically relevant equilibria (e.g. stationary vortical structures). Classical methods using dissipation based on one dimensional differencing are over-diffusive for such states [1, 2].

Several techniques have been proposed in the literature to design numerical methods that preserve these stationary solutions [3, 4, 5, 2, 6]. All these methods exploit the idea that numerical dissipation should not be proportional to the Laplacian operator, but rather involve quantities that vanish along the equilibrium state. For the acoustic equations for example, the appropriate form of the numerical dissipation involves the grad-div operator, which vanishes on states with zero divergence. Most of these methods are of first or second order of accuracy. Arbitrarily high order stationarity preserving finite element based method using multi-dimensional globalization of the fluxes (GF) and two different stabilization approaches has been proposed in [7]. A key point proven in the reference is that using a grad-div operator for the stabilization (as e.g. in the classical SUPG) is not enough to preserve the stationary solutions. Additional constraints must be imposted on the algebraic equations, which can be verified using flux globalization. The present work is based on the Finite Volume (FV) reformulation of this method, proposed in [1].

A common feature of all the stationarity preserving methods recalled is to be designed for Cartesian grids. Their extension to complex curvilinear geometries is not always straightforward. The presence of internal boundaries introduce additional challenges as boundary conditions can affect the structure of the stationary solutions, and introduce additional sources of error. Indeed, the diffusion operators vary along the boundary and must be designed so that they vanish when stationary solutions are reached, which can be nontrivial.

We aim at extending the stationarity preserving methods to complex curved domains and boundary conditions. To do so, we will develop a ghost point method (GPM) [8, 9, 10], which is an immersed boundary method that allows to handle complex geometries and boundary conditions on simple Cartesian grids. The main idea of the GPM is to modify the grid points that lie outside the physical domain but are still used in the stencil of the numerical scheme, so as to enforce the boundary conditions prescribed on the physical boundary, which is defined only by a discrete set of points. This feature not only eliminates the need for complex mesh generation by easily handling curved geometries, but also provides the flexibility to develop stationarity-preserving methods, which are typically designed on Cartesian grids.

To extend the GF-FV method to complex geometries we thus propose a new formulation of the GPM for the acoustic equations that is based on the multi-dimensional globalization of the fluxes, and allows to preserve stationary solutions with zero divergence. We will test our method on various numerical examples and we will compare it with a Rusanov-GPM method that does not preserve the stationary solutions.

The work is structured as follows. In section 2 we will recall the GF-FV formulation for Cartesian grids. In section 3 we will present the extension of the GF-FV method to complex geometries and boundary conditions using the GPM. In section 4 we will present some numerical simulations to test our method. Finally, in section 5 we will draw some conclusions and discuss future perspectives.

2. Global flux finite-volume formulation for cartesian grids

The basic idea of the multi-dimensional GF formulation [7, 11, 1] is to design a numerical scheme that solves the equivalent formulation of (1) given by

$$\partial_t q(t,x,y) + \partial_x \partial_y \mathcal{F}(t,x,y) = 0, \tag{4}$$

where $\mathcal{F}$ is the global flux and it is defined as

$$\mathcal{F}(t,x,y) = \int_{y_b}^{y} f(q(t,x,\eta))\mathrm{d}\eta + \int_{x_\ell}^{x} g(q(t,\xi,y))\mathrm{d}\xi - \int_{x_\ell}^{x}\int_{y_b}^{y} S(q(t,\xi,\eta),t,\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta. \tag{5}$$

Having a unique differential operator in front of the global flux (5) selects equilibria in a different way. Instead of looking, for example, at equilibria of the type $\nabla\cdot u = 0$, we now look for $\partial_x\partial_y\mathcal{F} = 0$, which sums up to looking for global fluxes that are sum of two unidirectional functions, i.e., $\mathcal{F}(t,x,y) = h^x(x)+h^y(y)$. Although this does not change anything at the continuous level, it has a huge impact at the discrete level, as it allows us to design numerical methods that preserve the stationary solutions of the considered hyperbolic problem, which are not preserved by classical methods. Proofs of the impact of the unique differential operator can be found in the finite element discretization of (4) with a SUPG stabilization presented in [7, 11, 12], and in the finite volume discretization proposed in [1]. As mentioned above, we will consider the FV formulation on Cartesian grids presented in [1].

Let us consider a Cartesian grid with N_x cells in the x direction and N_y cells in the y direction, with cell centers located at (x_i, y_j) for $i = 1,\dots,N_x$ and $j = 1,\dots,N_y$. We denote by Δx and Δy the grid spacings in the x and y directions, respectively. We also denote the interface points by $x_{i+\frac{1}{2}} = \frac{x_i+x_{i+1}}{2}$ and $y_{j+\frac{1}{2}} = \frac{y_j+y_{j+1}}{2}$ and the cells $C_{i,j} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$. The numerical solution at time t^n is denoted by $q^n_{i,j}$, which is an approximation of the cell average of the solution over the cell $C_{i,j}$. Now, the equation (4) can be discretized in space using a finite volume method as follows using two times the fundamental theorem of calculus:

$$\frac{q^{n+1}_{i,j} - q^n_{i,j}}{\Delta t} + \frac{\mathcal{F}(t^n, x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) - \mathcal{F}(t^n, x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}) - \mathcal{F}(t^n, x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}}) + \mathcal{F}(t^n, x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}})}{\Delta x \Delta y} = 0. \tag{6}$$

This formulation introduces global fluxes at the 4 corners that need to be approximated. At each corner, we will use a numerical corner flux that gathers information from the 4 surrounding cells and distribute them to the corresponding 4 cell averages, i.e.

$$\widehat{\mathcal{F}}^{(i,j)}_{i+\frac{1}{2},j+\frac{1}{2}} \approx \pm\mathcal{F}(t^n, x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) \tag{7}$$

where, in the previous formula, the sign will be determined by the normal from the corner towards the cell centers. Then, the FV update will read

$$\frac{q^{n+1}_{i,j} - q^n_{i,j}}{\Delta t} + \frac{\widehat{\mathcal{F}}^{(i,j)}_{i+\frac{1}{2},j+\frac{1}{2}} + \widehat{\mathcal{F}}^{(i+1,j+1)}_{i+\frac{1}{2},j+\frac{1}{2}} + \widehat{\mathcal{F}}^{(i+1,j)}_{i+\frac{1}{2},j+\frac{1}{2}} + \widehat{\mathcal{F}}^{(i,j+1)}_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x \Delta y} = 0. \tag{8}$$

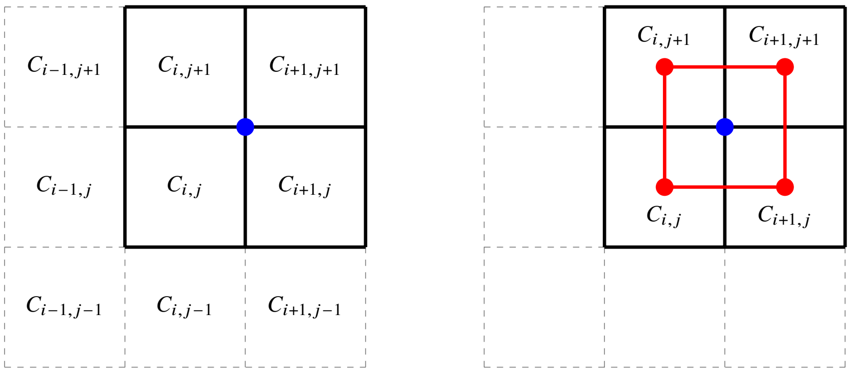

Figure 1: Cell labeling for the 2D grid: main grid (left), dual grid (right).

2.1 Reconstruction of global fluxes in cell centers

Since we are interested in preserving solutions with $\mathcal{F}$ that can be written as the sum of two unidirectional functions, the main challenge of the scheme lies in the definition of the global flux. This can be done through a composite quadrature formula that discretizes (5). We introduce three quantities $F \approx \int_{y_b}^{y} f(q(t,x,\eta))\mathrm{d}\eta$, $G \approx \int_{x_\ell}^{x} g(q(t,\xi,y))\mathrm{d}\xi$, and $R \approx \int_{x_\ell}^{x}\int_{y_b}^{y} S(q(t,\xi,\eta),t,\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta$, defined iteratively by

$$F_{i,j} := F_{i,j-1} + \Delta y \frac{f(q^n_{i,j}) + f(q^n_{i,j-1})}{2}, \tag{9}$$

$$G_{i,j} := G_{i-1,j} + \Delta x \frac{g(q^n_{i,j}) + g(q^n_{i-1,j})}{2}, \tag{10}$$

$$R_{i,j} := R_{i-1,j-1} + \Delta x \Delta y \frac{S(q^n_{i,j}) + S(q^n_{i-1,j}) + S(q^n_{i,j-1}) + S(q^n_{i-1,j-1})}{4}, \tag{11}$$

with $F_{i,0} = 0$ for all $i = 1, \dots, N_x$, $G_{0,j} = 0$ for all $j = 1, \dots, N_y$, and $R_{i,j} = 0$ when either $i = 0$ or $j = 0$. Then, the global flux at the cell center can be reconstructed as

$$\mathcal{F}_{i,j} := F_{i,j} + G_{i,j} - R_{i,j}. \tag{12}$$

Although this appears to be a global definition of the fluxes, the differences in the final update (8) result in only a local dependence on the 3×3 neighboring cells [1].

2.2 Corner fluxes

Corner fluxes are in general based on a different concept than classical interface fluxes. This is mainly due to the ambiguity arising from the need to enforce conservation with 4 states rather than 2. As proven in [13, 14], the conservation property and consequent Lax-Wendroff theorem are satisfied when the corner fluxes from one corner to the 4 cells sum up to zero, i.e.,

$$\widehat{\mathcal{F}}^{(i,j)}_{i+\frac{1}{2},j+\frac{1}{2}} + \widehat{\mathcal{F}}^{(i+1,j+1)}_{i+\frac{1}{2},j+\frac{1}{2}} + \widehat{\mathcal{F}}^{(i+1,j)}_{i+\frac{1}{2},j+\frac{1}{2}} + \widehat{\mathcal{F}}^{(i,j+1)}_{i+\frac{1}{2},j+\frac{1}{2}} = 0. \tag{13}$$

There are various definitions of consistency for corner fluxes [1, 14], and they are clearly linked with the concept of residual distribution [13].

Let us define the GF-SUPG corner flux by a consistent central part and a diffusive part as

$$\widehat{\mathcal{F}}^{(i,j+1)}_{i+\frac{1}{2},j+\frac{1}{2}} = \overline{\mathcal{F}}_{i+\frac{1}{2},j+\frac{1}{2}} n^{(i,j+1)}_{i+\frac{1}{2},j+\frac{1}{2}} + \mathcal{D}^{(i,j+1)}_{i+\frac{1}{2},j+\frac{1}{2}}, \qquad \widehat{\mathcal{F}}^{(i+1,j+1)}_{i+\frac{1}{2},j+\frac{1}{2}} = \overline{\mathcal{F}}_{i+\frac{1}{2},j+\frac{1}{2}} n^{(i+1,j+1)}_{i+\frac{1}{2},j+\frac{1}{2}} + \mathcal{D}^{(i+1,j+1)}_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{14}$$

$$\widehat{\mathcal{F}}^{(i,j)}_{i+\frac{1}{2},j+\frac{1}{2}} = \overline{\mathcal{F}}_{i+\frac{1}{2},j+\frac{1}{2}} n^{(i,j)}_{i+\frac{1}{2},j+\frac{1}{2}} + \mathcal{D}^{(i,j)}_{i+\frac{1}{2},j+\frac{1}{2}}, \qquad \widehat{\mathcal{F}}^{(i+1,j)}_{i+\frac{1}{2},j+\frac{1}{2}} = \overline{\mathcal{F}}_{i+\frac{1}{2},j+\frac{1}{2}} n^{(i+1,j)}_{i+\frac{1}{2},j+\frac{1}{2}} + \mathcal{D}^{(i+1,j)}_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{15}$$

where

$$\overline{\mathscr{F}}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{4}(\mathscr{F}_{i,j} + \mathscr{F}_{i+1,j} + \mathscr{F}_{i+1,j+1} + \mathscr{F}_{i,j+1})$$

is the average of the global fluxes at the corner, $n^{(i+\ell,j+r)}_{i+\frac{1}{2},j+\frac{1}{2}} = 1$ for $(\ell, r) = (0,0)$ or $(\ell, r) = (1,1)$ and $n^{(i+\ell,j+r)}_{i+\frac{1}{2},j+\frac{1}{2}} = -1$ for $(\ell, r) = (0,1)$ or $(\ell, r) = (1,0)$. We define the numerical diffusion as

$$\begin{aligned}
\mathcal{D}^{(i,j+1)}_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{\alpha\Delta}{4}\left(-\frac{J^x}{\Delta x} + \frac{J^y}{\Delta y}\right)\Phi_{i+\frac{1}{2},j+\frac{1}{2}}, \qquad & \mathcal{D}^{(i+1,j+1)}_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{\alpha\Delta}{4}\left(+\frac{J^x}{\Delta x} + \frac{J^y}{\Delta y}\right)\Phi_{i+\frac{1}{2},j+\frac{1}{2}}, \\
\mathcal{D}^{(i,j)}_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{\alpha\Delta}{4}\left(-\frac{J^x}{\Delta x} - \frac{J^y}{\Delta y}\right)\Phi_{i+\frac{1}{2},j+\frac{1}{2}}, \qquad & \mathcal{D}^{(i+1,j)}_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{\alpha\Delta}{4}\left(+\frac{J^x}{\Delta x} - \frac{J^y}{\Delta y}\right)\Phi_{i+\frac{1}{2},j+\frac{1}{2}},
\end{aligned} \tag{16}$$

where $\Delta = \sqrt{\Delta x^2 + \Delta y^2}/\sqrt{2}$, J^x and J^y are the Jacobian of the fluxes in x and y directions, respectively, and $\Phi_{i+\frac{1}{2},j+\frac{1}{2}}$ is the dual cell residual $\widetilde{C}_{i+\frac{1}{2},j+\frac{1}{2}} = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$, i.e.,

$$\Phi_{i+\frac{1}{2},j+\frac{1}{2}} := \Phi(\widetilde{\mathscr{F}}_{i+\frac{1}{2},j+\frac{1}{2}}) := \mathscr{F}_{i+1,j+1} - \mathscr{F}_{i,j+1} - \mathscr{F}_{i+1,j} + \mathscr{F}_{i,j} = \int_{\widetilde{C}_{i+\frac{1}{2},j+\frac{1}{2}}} \partial_{xy}\widetilde{\mathscr{F}}_{i+\frac{1}{2},j+\frac{1}{2}} \mathrm{d}x\mathrm{d}y\ . \tag{17}$$

This definition is inspired by the GF-SUPG residual applied on the dual cell of the FV scheme, see [15, 1] for more details.

Conservation is guaranteed by the fact that the diffusion terms sum to zero around one corner.

It is easy to show that, when $\mathscr{F}_{i,j} = \alpha_i + \beta_j$, then $\Phi_{i+\frac{1}{2},j+\frac{1}{2}} = 0$, the diffusion term vanishes. Moreover, if one sums up the 4 corner fluxes related to one cell, also the central part sums up to zero, hence the scheme preserves the stationary solutions [1].

For comparison, we will use also a Rusanov corner flux, which is defined as (14) with the artificial diffusion defined by

$$\begin{aligned}
\mathcal{D}^{(i,j+1)}_{i+\frac{1}{2},j+\frac{1}{2}} &= \alpha\Delta(q_{i,j+1} - \bar{q}_{i+\frac{1}{2},j+\frac{1}{2}}), \qquad & \mathcal{D}^{(i+1,j+1)}_{i+\frac{1}{2},j+\frac{1}{2}} &= \alpha\Delta(q_{i+1,j+1} - \bar{q}_{i+\frac{1}{2},j+\frac{1}{2}}), \\
\mathcal{D}^{(i,j)}_{i+\frac{1}{2},j+\frac{1}{2}} &= \alpha\Delta(q_{i,j} - \bar{q}_{i+\frac{1}{2},j+\frac{1}{2}}), \qquad & \mathcal{D}^{(i+1,j)}_{i+\frac{1}{2},j+\frac{1}{2}} &= \alpha\Delta(q_{i+1,j} - \bar{q}_{i+\frac{1}{2},j+\frac{1}{2}}),
\end{aligned} \tag{18}$$

with $\bar{q}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{4}(q_{i,j} + q_{i+1,j} + q_{i+1,j+1} + q_{i,j+1})$.

This scheme will not be stationarity preserving, as the diffusion does not vanish on the stationary solutions of the acoustic equations with zero divergence. So we will use it as a comparison to show the importance of stationarity preservation for the accuracy of the method.

We want to highlight that the GF-SUPG-FV method is second order accurate at the equilibrium [1] even if the underlying finite volume paradigm is only first order accurate. This is a consequence of the fact that the diffusion term vanishes at the equilibrium, and the method reduces to a consistent central scheme, which is second order accurate. On the other hand, the Rusanov corner flux is only first order accurate at the equilibrium, as its diffusion term does not vanish and dominates the discretization error.

3. Extension to curved domains with ghost point method

Now, we suppose to have a domain Ω with a smooth boundary Γ that is defined as the zero level set of a function ϕ, i.e., $\Gamma = \{\phi = 0\}$. Moreover, the level set function ϕ is a signed distance function, i.e., $\phi(x) = \pm\mathrm{dist}(x, \Gamma)$, where the sign is positive inside the domain and negative outside. We consider a Cartesian grid that covers the domain Ω and we classify the cells as inside or outside the computational domain based on the position of their barycenter with respect to Γ.

To define the GPM [10], we introduce the following notation. We define with $\tilde{\Omega}$ the computational

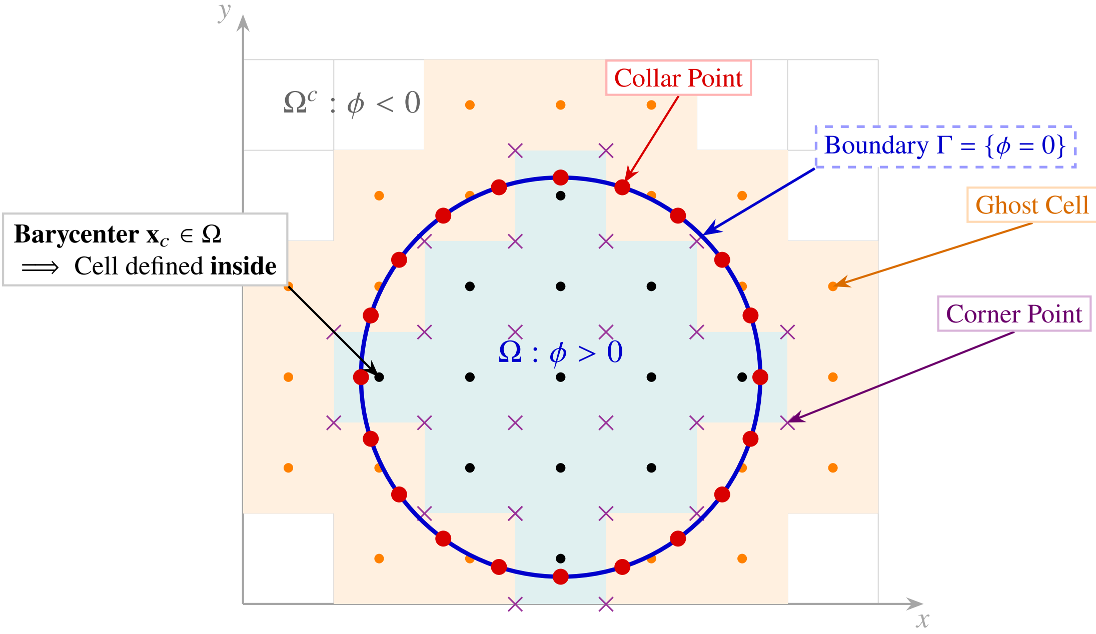

Figure 2: Illustration for ghost point method for a circular domain. In green the cells that are considered inside the computational domain $\tilde{\Omega}$, whose barycenter is inside the physical domain. In violet, the computational corners, neighboring a computational cell. In orange the ghosts cells, neighboring a computational corner. In red, the collar points onto which we impose the boundary conditions

domain, which is the union of the cells whose barycenter is inside Ω. Now, for the update of these cell averages, we will need the corner fluxes at each of the four corners of the cells in $\tilde{\Omega}$. We call all these corners the *computational corners*, see Figure 2. To compute these corner fluxes, we will need to use the 4 values of the solution at the corners, but not all of them are defined as some might be outside $\tilde{\Omega}$. We will define the *ghosts cells* as the cells that are outside $\tilde{\Omega}$ but neighboring a computational corner, see Figure 2. Since the solution is not defined in the ghost cells, we will need to extrapolate it from the solution in the interior cells and the boundary conditions prescribed on the physical boundary Γ.

To do so, we will define the *collar points* as a collection of discrete points lying on the boundary Γ, as a dense enough set. Then, we will impose the boundary conditions on the collar points and we will use them to extrapolate the solution at the ghost cells, which will be used to compute the corner fluxes.

3.1 Extrapolation of the solution at the ghost cells

Various approaches have been proposed in literature to extrapolate the solution at the ghost cells, see for example [10, 8, 16, 17] for similar immersed/unfitted methods. They all include various combination of weak and strong constraints on the boundary conditions at the collar points and on the solution inside the domain. Also the underlying polynomial one wants to interpolate/extrapolate can be of various degree and the stencil can be more or less large.

Let us focus on a corner point whose neighboring cells are not all in the computational domain. To start with a simple approach, let us consider a bi-linear polynomial reconstruction. We will use a square of 3×3 cells to define the polynomial and to impose the conditions. In particular, we will choose the 3×3 cells centering around the cell that, from the corner point, is going to the inside of the domain: take the inward facing normal $(n_x, n_y) = \mathbf{n}(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) := \nabla\phi(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$ at the corner point $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$ and take the cell with barycenter $(x_{i+\frac{1}{2}+\frac{1}{2}\mathrm{sign}(n_x)}, y_{j+\frac{1}{2}+\frac{1}{2}\mathrm{sign}(n_y)})$. The hope with this operation is to have the largest number of internal data to use in the reconstruction procedure.

Now, that we found the center $(x_{\tilde{i}}, y_{\tilde{j}})$ of the 3×3 square, we will look for the constrained least square problem

$$\begin{aligned} &\min_{p(x,y)\in\mathbb{Q}^1} \sum_{\ell=-1}^{1}\sum_{r=-1}^{1} \Big(p(x_{\tilde{i}+\ell}, y_{\tilde{j}+r}) - q(x_{\tilde{i}+\ell}, y_{\tilde{j}+r})\Big)\,\chi(x_{\tilde{i}+\ell}, y_{\tilde{j}+r})_{\tilde{\Omega}} \qquad \text{subject to} \\ &p(x_k^c, y_k^c) = u(x_k^c, y_k^c), \qquad (x_k^c, y_k^c) \in \Gamma \text{ collar points for } k = 1, \dots, N^c_{i+\frac{1}{2},j+\frac{1}{2}}, \end{aligned} \tag{19}$$

where (x_k^c, y_k^c) are the closest $N^c_{i+\frac{1}{2},j+\frac{1}{2}} \in \mathbb{N}$ collar points to the corner point $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$ and $\chi(x_{\tilde{i}+\ell}, y_{\tilde{j}+r})_{\tilde{\Omega}}$ is the indicator function that is 1 if the cell with barycenter $(x_{\tilde{i}+\ell}, y_{\tilde{j}+r})$ is in the computational domain and 0 otherwise.

Let us define the matrix $A_{i+\frac{1}{2},j+\frac{1}{2}}$ as the matrix of the least square problem, i.e.,

$$[A_{i+\frac{1}{2},j+\frac{1}{2}}]_{z,k} = \sum_{\ell=-1}^{1}\sum_{r=-1}^{1} \varphi_z(x_{\tilde{i}+\ell}, y_{\tilde{j}+r})\varphi_k(x_{\tilde{i}+\ell}, y_{\tilde{j}+r})\chi(x_{\tilde{i}+\ell}, y_{\tilde{j}+r})_{\tilde{\Omega}}, \tag{20}$$

where φ_z are the basis functions of the bi-linear polynomial written as $p(x,y) = \sum_z a_z\varphi_z(x,y)$.

The number of collar points to be used in the constrainted least square problem is a depending on the rank of the LSQ matrix: $N^c_{i+\frac{1}{2},j+\frac{1}{2}} = \max(2, N_{\text{basis}} - \text{rank}(A_{i+\frac{1}{2},j+\frac{1}{2}}) + 1)$. This choice is very arbitrary, but we found this as a sweet spot between having the minimum amount of boundary information to impose $N_{\text{basis}} - \text{rank}(A_{i+\frac{1}{2},j+\frac{1}{2}})$ and still introducing some sort of stabilization effect enforcing the boundary conditions. In the literature [10], one can find approaches with various definitions of the computational stencil used to assemble the aforementioned least square system.

This leads to the assembly of the the matrix $D_{i+\frac{1}{2},j+\frac{1}{2}}$ for the Lagrange multipliers of the constraints, which is defined as the basis functions evaluated at the collar points, i.e.,

$$[D_{i+\frac{1}{2},j+\frac{1}{2}}]_{z,k} = \varphi_z(x_k^c, y_k^c), \qquad k = 1, \dots, N^c_{i+\frac{1}{2},j+\frac{1}{2}}. \tag{21}$$

To solve the constrained least square problem, we can use the method of Lagrange multipliers, which leads to the following system of equations:

$$\begin{bmatrix} A_{i+\frac{1}{2},j+\frac{1}{2}} & D_{i+\frac{1}{2},j+\frac{1}{2}} \\ D^T_{i+\frac{1}{2},j+\frac{1}{2}} & 0 \end{bmatrix} \begin{bmatrix} a \\ \lambda \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix}, \tag{22}$$

where a is the vector of coefficients of the polynomial, λ is the vector of Lagrange multipliers, $[b]_k = \sum_{\ell,r=-1}^{1} \varphi_k(x_{\tilde{i}+\ell}, y_{\tilde{j}+r})q(x_{\tilde{i}+\ell}, y_{\tilde{j}+r})\chi(x_{\tilde{i}+\ell}, y_{\tilde{j}+r})_{\tilde{\Omega}}$ is the vector of the right-hand side of the least square problem, and $[c]_k = u(x_k^c, y_k^c)$ is the vector of the right-hand side of the constraints.

When Neumann boundary conditions are imposed, we can use the same approach, but we need to impose the constraints on the normal derivative of the polynomial at the collar points, i.e.,

$$\begin{aligned} &\min_{p(x,y)\in\mathbb{Q}^1} \sum_{\ell=-1}^{1}\sum_{r=-1}^{1} \Big(p(x_{\tilde{i}+\ell}, y_{\tilde{j}+r}) - q(x_{\tilde{i}+\ell}, y_{\tilde{j}+r})\Big)\,\chi(x_{\tilde{i}+\ell}, y_{\tilde{j}+r})_{\tilde{\Omega}} \qquad \text{subject to} \\ &\nabla p(x_k^c, y_k^c)\cdot \mathbf{n}(x_k^c, y_k^c) = g(x_k^c, y_k^c), \qquad (x_k^c, y_k^c) \in \Gamma \text{ collar points for } k = 1, \dots, N^c_{i+\frac{1}{2},j+\frac{1}{2}}, \end{aligned} \tag{23}$$

where g is the function that defines the Neumann boundary condition, i.e., $\nabla u \cdot \mathbf{n} = g$ on Γ.

This changes the definition of the matrix $D_{i+\frac{1}{2},j+\frac{1}{2}}$ in (22), which is now defined as the normal derivative of the basis functions evaluated at the collar points, i.e.,

$$[D_{i+\frac{1}{2},j+\frac{1}{2}}]_{z,k} = \nabla\varphi_z(x_k^c, y_k^c)\cdot \mathbf{n}(x_k^c, y_k^c), \qquad k = 1, \dots, N^c_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{24}$$

and the corresponding right-hand side of the constraints, which is now defined as $[c]_z = g(x_k^c, y_k^c)$.

Clearly, for fixed boundaries, the matrices do not change in time and can be precomputed, inverted and stored for each corner point, which is a very small number of points compared to the total number of cells in the grid. This means that at runtime only few small matrix-vector products are needed to extrapolate the solution at the ghost cells, which is very cheap compared to the rest of the computations performed by the method.

3.2 Vector case

When dealing with vector variables, as the velocity field, one needs to impose different boundary conditions, e.g. imposing wall boundary conditions sums up to having the normal velocity imposed to zero $\mathbf{v}\cdot\mathbf{n} = 0$, the pressure and tangential velocity to satisfy homogeneous Neumann boundary conditions $\partial_{\mathbf{n}}(\mathbf{v}\cdot\tau) = 0 = \partial_{\mathbf{n}} p$. Here, τ is the tangential vector to the boundary, which can be defined as $\tau = (-n_y, n_x)$ in 2D.

This leads to a similar constrained least square problem as the one for the scalar case, but on two variables and with the double of constraints to impose. For this, we need four further matrices of Lagrange multipliers, one for the normal velocity and one for the tangential velocity, which are defined as

$$[D^{n_x}_{i+\frac{1}{2},j+\frac{1}{2}}]_{z,k} = \varphi_z(x_k^c, y_k^c) n_x(x_k^c, y_k^c), \qquad k = 1, \dots, N^c_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{25}$$

$$[D^{n_y}_{i+\frac{1}{2},j+\frac{1}{2}}]_{z,k} = \varphi_z(x_k^c, y_k^c) n_y(x_k^c, y_k^c), \qquad k = 1, \dots, N^c_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{26}$$

$$[D^{\tau_x}_{i+\frac{1}{2},j+\frac{1}{2}}]_{z,k} = \partial_{\mathbf{n}}\varphi_z(x_k^c, y_k^c) \cdot \tau_x(x_k^c, y_k^c), \qquad k = 1, \dots, N^c_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{27}$$

$$[D^{\tau_y}_{i+\frac{1}{2},j+\frac{1}{2}}]_{z,k} = \partial_{\mathbf{n}}\varphi_z(x_k^c, y_k^c) \cdot \tau_y(x_k^c, y_k^c), \qquad k = 1, \dots, N^c_{i+\frac{1}{2},j+\frac{1}{2}}. \tag{28}$$

The final system of equations to solve is then

$$\begin{bmatrix} A_{i+\frac{1}{2},j+\frac{1}{2}} & 0 & D^{n_x}_{i+\frac{1}{2},j+\frac{1}{2}} & D^{\tau_x}_{i+\frac{1}{2},j+\frac{1}{2}} \\ 0 & A_{i+\frac{1}{2},j+\frac{1}{2}} & D^{n_y}_{i+\frac{1}{2},j+\frac{1}{2}} & D^{\tau_y}_{i+\frac{1}{2},j+\frac{1}{2}} \\ (D^{n_x}_{i+\frac{1}{2},j+\frac{1}{2}})^T & (D^{n_y}_{i+\frac{1}{2},j+\frac{1}{2}})^T & 0 & 0 \\ (D^{\tau_x}_{i+\frac{1}{2},j+\frac{1}{2}})^T & (D^{\tau_y}_{i+\frac{1}{2},j+\frac{1}{2}})^T & 0 & 0 \end{bmatrix} \begin{bmatrix} a^u \\ a^v \\ \lambda^n \\ \lambda^\tau \end{bmatrix} = \begin{bmatrix} b^u \\ b^v \\ c^n \\ c^\tau \end{bmatrix}, \tag{29}$$

where a^u and a^v are the coefficients of the polynomial for the two components of the velocity, λ^n and λ^τ are the Lagrange multipliers for the normal and tangential velocity constraints,

$$[b^u]_k = \sum_{\ell,r=-1}^{1} \varphi_k(x_{\tilde{i}+\ell}, y_{\tilde{j}+r}) u(x_{\tilde{i}+\ell}, y_{\tilde{j}+r}) \chi(x_{\tilde{i}+\ell}, y_{\tilde{j}+r})_{\tilde{\Omega}}$$

$$[b^v]_k = \sum_{\ell,r=-1}^{1} \varphi_k(x_{\tilde{i}+\ell}, y_{\tilde{j}+r}) v(x_{\tilde{i}+\ell}, y_{\tilde{j}+r}) \chi(x_{\tilde{i}+\ell}, y_{\tilde{j}+r})_{\tilde{\Omega}}$$

are the right-hand side of the least square problem for the two components of the velocity, and $[c^n]_k = 0$ and $[c^\tau]_k = 0$ are the right-hand side of the constraints for the normal and tangential velocity, respectively.

The same consideration on the precomputation of the matrices applies also in this case, as the matrices do not change in time for fixed geometriesand they are small, so the computational cost of the extrapolation is negligible compared to the rest of the computations of the method.

To extend this technique to higher order of accuracy, one can simply consider higher order reconstructions, e.g. $\mathbb{Q}^p$, and enlarge the stencil from which taking the solution information. Again, the number of constraints has to be chosen in a way to have at least the minimum amount of boundary information to impose, but it can be larger to introduce some sort of stabilization effect. The stability of such methods strongly depends on the size of the used stencil and on the number of constraints, so a careful analysis is needed to find the best combination of these parameters. For instance, in the finite difference context [18], larger stencils composed by only the farthest points are used to stabilize the method.

4. Simulations

We will test the proposed numerical method on various geometries and test cases. We want to assess the accuracy of the scheme and the equilibria of the methods.

All methods will use a second order Runge-Kutta with Butcher tableau

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array}.$$

4.1 Acoustic vortex in ring

We start from vortex simulations. These are divergence-free structures that are preserved by the acoustic equations and they are a perfect test case to assess the stationarity preservation property of the method. In particular, we will consider a squared domain where we remove a circular part in the middle and an the outer part of a larger circle $\Omega = (-1.5, 1.5)^2 \setminus B((0,0),1) \setminus B((0,0),1.384)^c = B((0,0),1.384) \setminus B((0,0),1)$. The initial condition is given by a compact-support vortex centered in the middle of the domain with maximum radius $r_0 = 2.5$, defined as in [19], i.e.,

$$\begin{aligned} p(x,y) &= 1, \\ u(x,y) &= y f(\rho(x,y)), \\ v(x,y) &= -x f(\rho(x,y)), \end{aligned}$$

with $\rho(x,y) = \frac{\sqrt{x^2+y^2}}{r_0}$, where $f(\rho) = \gamma(1+\cos(\pi\rho))^2$ and $\gamma = \frac{12\pi\sqrt{0.981}}{r_0\sqrt{315\pi^2-2048}}$.

Since we know the exact solution, in this section, we carried out two sets of numerical experiments: in the first one, we compare the GF-SUPG-FV approach against the GF-Rusanov-FV one when imposing exact Dirichlet conditions on all conservative variables; in the second one, we compare the two approaches when imposing the more complex wall boundary conditions. In table 1 we can see the convergence results for the GF-Rusanov-FV - GPM and GF-SUPG-FV - GPM methods with wall and Dirichlet boundary conditions. The GF-Rusanov-FV - GPM method shows a slow convergence rate close to order 1 and with large errors, while the GF-SUPG-FV - GPM method shows a better convergence rate, close to order 1.5 for wall BC and at order 2 for Dirichlet BC. It should be noticed that a full second-order accuracy is not reached even for the GF approach, due to the imposition of Neumann boundary conditions on the pressure using the derivative of the bi-linear polynomial, which is formally first-order accurate. Higher order extensions of the GPM also for Neumann BC will be presented in future works.

We can see in figure 3 the velocity vector field of the vortex in a ring problem with GF-SUPG-FV-GPM and GF-Rusanov-FV-GPM schemes on grids with 40×40 and 80×80 cells at time $T = 10$. We can see that the GF-Rusanov-FV - GPM method is not able to preserve the structure of the vortex and diffuses away all the structures leading to zero velocity as the time goes by, in particular for coarse grids, while the GF-SUPG-FV - GPM method is able to preserve it much better, even on the coarser grid. Note that the colorbar of the simulations are not the same.

Looking at the time evolution of the L_2 norm of the residual in figure 4, we can see that the GF-Rusanov-FV - GPM method gets stuck at much higher values as it continue to diffuse away the solution, while the GF-SUPG-FV - GPM method maintains a lower residual being very close to a numerical equilibrium.

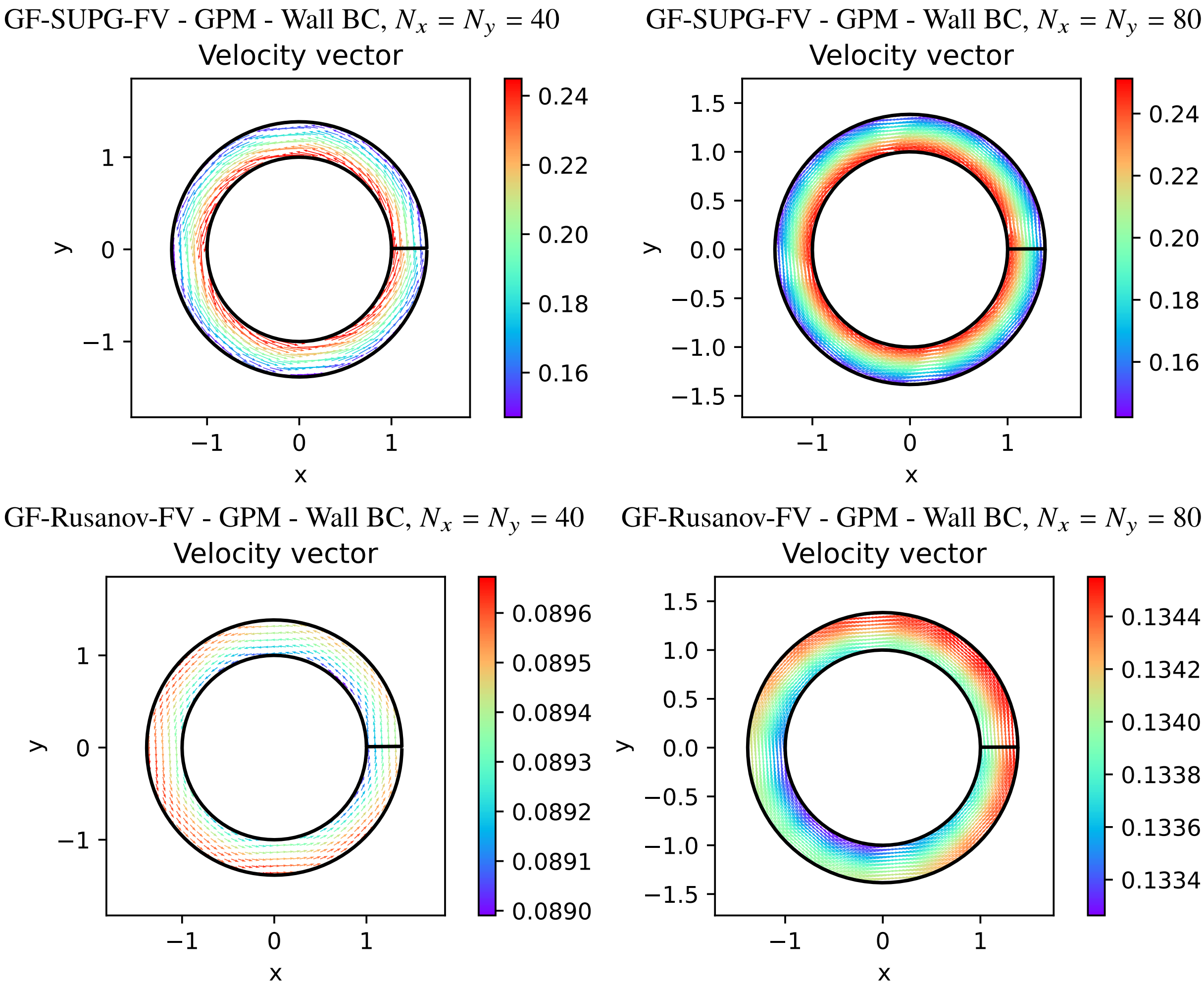

Figure 3: Velocity vector field of the vortex in a ring problem with GF-SUPG-FV-GPM and GF-Rusanov-FV-GPM schemes on grids with 40×40 and 80×80 cells at time $T = 10$.

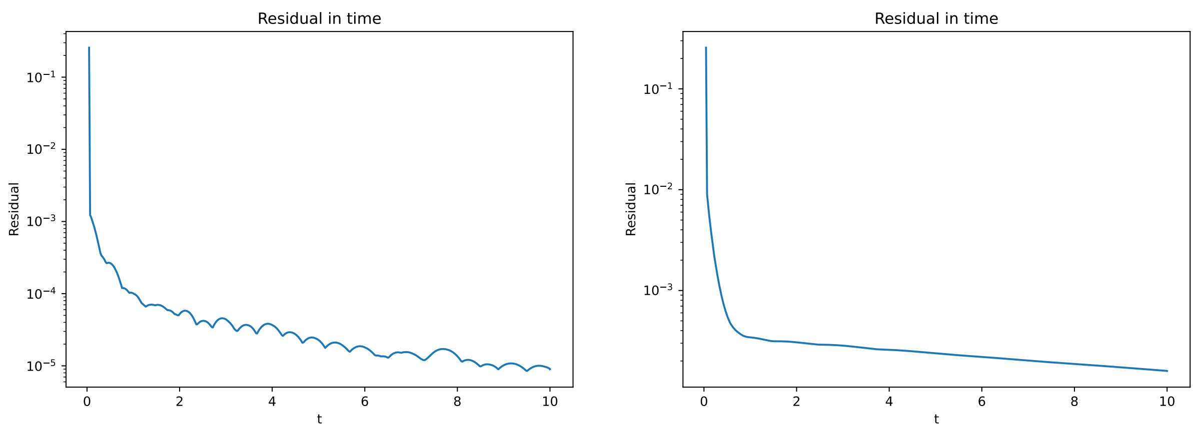

Figure 4: Time evolution of the L_2 norm of the residual for the vortex in a ring problem with GF-SUPG-FV-GPM (left) and GF-Rusanov-FV-GPM (right) schemes on grids with 40×40 cells. Note that the colorbars are different.

Table 1: Linear acoustic system: vortex in a ring ($t_f = 0.1$). L_2 error and order of accuracy $\tilde{n}$ for GF-Rusanov-FV - GPM and GF-SUPG-FV - GPM.

	p		u		v	
N_x, N_y	L_2	$\tilde{n}$	L_2	$\tilde{n}$	L_2	$\tilde{n}$
			GF-SUPG-FV - GPM - Dirichlet BC			
20	9.421e-04	-	1.173e-03	-	1.335e-03	-
40	1.357e-04	2.80	1.573e-04	2.90	1.587e-04	3.07
80	3.257e-05	2.06	3.972e-05	1.99	4.047e-05	1.97
160	8.613e-06	1.92	1.047e-05	1.92	1.055e-05	1.94
320	2.199e-06	1.97	2.413e-06	2.12	2.394e-06	2.14
			GF-SUPG-FV - GPM - Wall BC			
20	2.97e-03	-	3.91e-03	-	3.91e-03	-
40	1.49e-03	0.99	2.33e-03	0.75	2.34e-03	0.74
80	4.47e-04	1.74	1.31e-03	0.83	1.31e-03	0.84
160	9.67e-05	2.21	5.40e-04	1.28	5.40e-04	1.28
320	3.74e-05	1.37	2.05e-04	1.40	2.04e-04	1.40
			GF-Rusanov-FV - GPM - Dirichlet BC			
20	2.39e-04	-	7.24e-03	-	7.38e-03	-
40	7.22e-05	1.73	5.33e-03	0.44	5.35e-03	0.46
80	2.01e-05	1.85	3.25e-03	0.72	3.25e-03	0.72
160	5.54e-06	1.86	1.81e-03	0.84	1.81e-03	0.84
320	1.46e-06	1.92	9.69e-04	0.90	9.69e-04	0.90
			GF-Rusanov-FV - GPM - Wall BC			
20	7.11e-04	-	2.88e-02	-	2.88e-02	-
40	1.02e-03	-0.52	2.39e-02	0.27	2.38e-02	0.28
80	3.90e-04	1.39	1.46e-02	0.71	1.46e-02	0.70
160	1.72e-04	1.18	8.37e-03	0.81	8.33e-03	0.81
320	6.77e-05	1.34	4.77e-03	0.81	4.75e-03	0.81

4.2 Potential flow for acoustics

Next, we will consider the potential flow around a cylinder, which is a stationary solution of the acoustic equations and an exact form is known. We will use this test case again to assess the accuracy of the method and the stationarity preservation property.

We use as a domain a square with a circular hole in the middle $\Omega = (-1.5, 1.5)^2 \setminus B((0,0), 0.5)$, and we set Dirichlet boundary conditions on the square boundaries. The initial condition is given by the potential flow around a cylinder, which is defined as

$$p(x,y) = 1, \qquad u(x,y) = \frac{\partial \psi}{\partial x}, \qquad v(x,y) = \frac{\partial \psi}{\partial y},$$

where ψ is the potential function defined as

$$\psi(x,y) = u_\infty \left(x + \frac{R^2 x}{x^2 + y^2} \right), \tag{30}$$

with $u_\infty = 1$ the velocity at infinity and $R = 0.5$ the radius of the cylinder. Once again, the fact that the exact solution is available allows us to perform two sets of numerical experiments: on one side, by imposing exact Dirichlet conditions on the circular boundary, and on the other, by imposing wall boundary conditions.

In table 2 we can see the convergence results for the GF-Rusanov-FV - GPM and GF-SUPG-FV - GPM methods with wall and Dirichlet boundary conditions. In this case, the Dirichlet boundary conditions push the solution through the simulation in a very accurate way both for the GF-Rusanov-FV - GPM and GF-SUPG-FV - GPM methods, with convergence rates close to order 2, while the wall boundary conditions lead to a much worse convergence rate, close to order 0.75 for GF-Rusanov-FV - GPM and

Table 2: Linear acoustic system: potential flow around a cylinder ($t_f = 1$). L_2 error and order of accuracy $\tilde{n}$ for GF-Rusanov-FV - GPM and GF-SUPG-FV - GPM.

	p		u		v	
N_x, N_y	L_2	$\tilde{n}$	L_2	$\tilde{n}$	L_2	$\tilde{n}$
	GF-SUPG-FV - GPM - Dirichlet BC					
20	1.87e-02	-	4.51e-02	-	5.00e-02	-
40	3.61e-03	2.38	1.42e-02	1.67	1.37e-02	1.87
80	1.15e-03	1.65	3.60e-03	1.98	3.63e-03	1.92
160	2.74e-04	2.06	7.70e-04	2.23	7.85e-04	2.21
320	6.55e-05	2.07	1.80e-04	2.09	1.80e-04	2.12
	GF-SUPG-FV - GPM - Wall BC					
20	1.12e-02	-	6.34e-02	-	5.15e-02	-
40	2.04e-03	2.46	2.22e-02	1.52	1.95e-02	1.40
80	1.74e-03	0.24	9.36e-03	1.24	8.12e-03	1.27
160	4.65e-04	1.90	3.27e-03	1.52	2.75e-03	1.56
320	1.46e-04	1.67	1.23e-03	1.42	9.90e-04	1.47
	GF-Rusanov-FV - GPM - Dirichlet BC					
20	1.39e-02	-	4.90e-02	-	3.41e-02	-
40	3.71e-03	1.90	9.82e-03	2.32	7.53e-03	2.18
80	1.19e-03	1.64	2.85e-03	1.79	1.98e-03	1.93
160	2.79e-04	2.09	5.97e-04	2.25	4.88e-04	2.02
320	5.80e-05	2.27	1.22e-04	2.30	1.16e-04	2.07
	GF-Rusanov-FV - GPM - Wall BC					
20	2.52e-01	-	5.74e-01	-	2.67e-01	-
40	1.86e-01	0.44	4.06e-01	0.50	1.75e-01	0.61
80	1.20e-01	0.62	2.78e-01	0.55	1.22e-01	0.51
160	7.11e-02	0.76	1.80e-01	0.62	8.47e-02	0.53
320	3.92e-02	0.86	1.14e-01	0.66	5.69e-02	0.57

close to order 1.4 for GF-SUPG-FV - GPM. We remark that in this specific case, the exact solution also verify $\Delta u = \Delta v = \Delta p = 0$, so also the classical GF-Rusanov-FV diffusion is zero on the exact solution, hence, the superconvergence of the GF-Rusanov-FV - GPM method with Dirichlet BC.

In the wall boundary condition case, the GF-SUPG-FV - GPM method offers a clear advantage over the GF-Rusanov-FV - GPM method, with much smaller errors and better convergence rates. Note that wall boundary conditions are one that more realistically have to be used as the exact solution is not always known.

Looking at Figure 5, we can see the simulation of the potential flow problem for acoustics with the GF-SUPG-FV scheme on a grid with 80×80 cells at time $T = 10$ with wall BC. The left figure shows the pressure field, the middle figure the norm of the velocity field and the right figure the velocity field represented with arrows. The GF-Rusanov-FV - GPM method is not able to recover constant pressure.

4.3 Potential flow around a flower

In this test, we look for a potential flow around a flower-shaped obstacle, which is a more complex geometry than the cylinder. The domain is defined as $\Omega = (-1.5, 1.5)^2 \setminus \mathcal{F}$, where $\mathcal{F} = \{(x, y) : x^2 + y^2 < R(x, y)\}$ with

$$R(x, y) = R(1 + s \sin(\alpha 2\pi\theta(x, y))), \tag{31}$$

with $s = 3$, $\alpha = 5$, $R = 0.5$ and $\theta(x, y) = \arctan(y/x)$.

We set Dirichlet BC on the square boundaries for $p = 1, u = 1, v = 0$ and wall BC on the flower boundary. The initial condition is given by the potential flow around a cylinder defined as in the previous section.

In this case, no exact solution is known, so we just take a look at a qualitative comparison of the results.

In Figure 6, we can see the simulation of the potential flow around a flower problem for acoustics with the

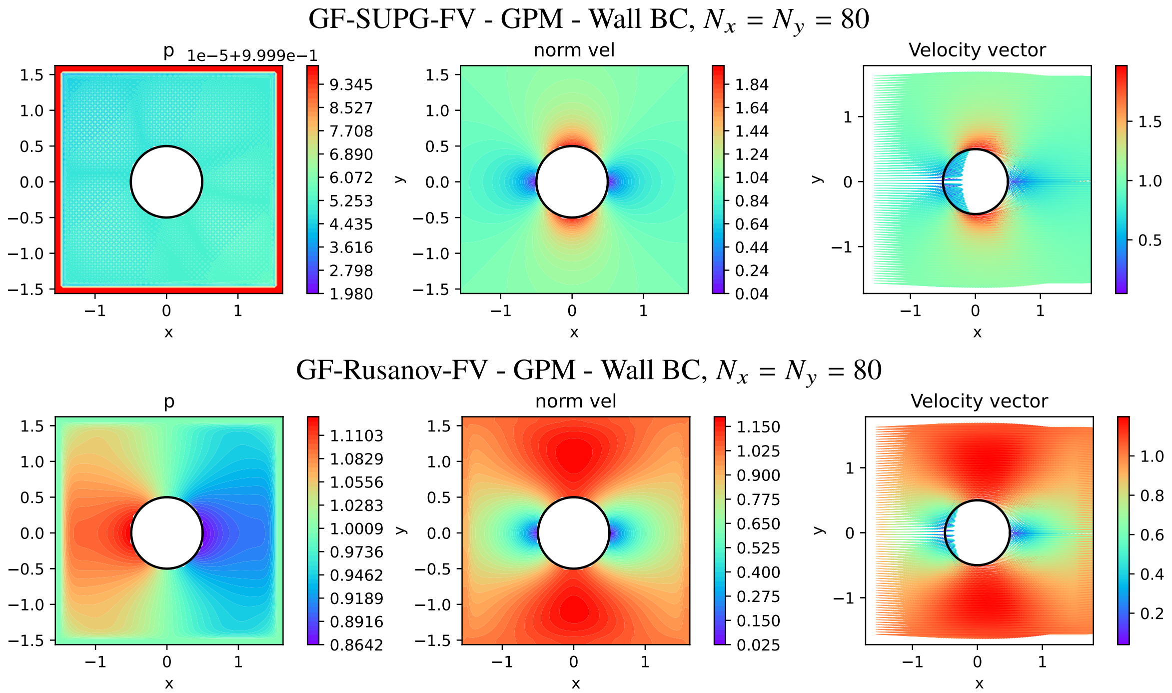

Figure 5: Simulation of the potential flow problem for acoustics with the GF-SUPG-FV scheme on a grid with 80×80 cells at time $T = 1$. The left figure shows the pressure field, the middle figure the norm of the velocity field and the right figure the velocity field represented with arrows.

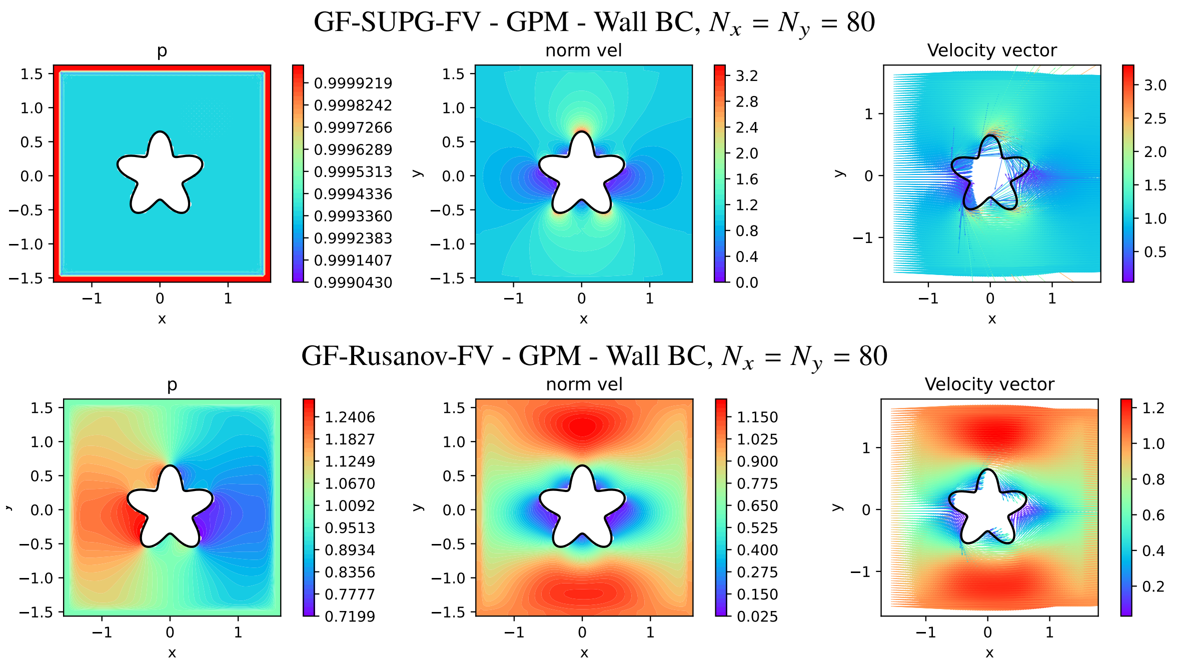

Figure 6: Simulation of the flower problem for acoustics with the GF-SUPG-FV scheme on a grid with 50×50 cells at time $T = 50$. The left figure shows the pressure field, the middle figure the norm of the velocity field and the right figure the velocity field represented with arrows.

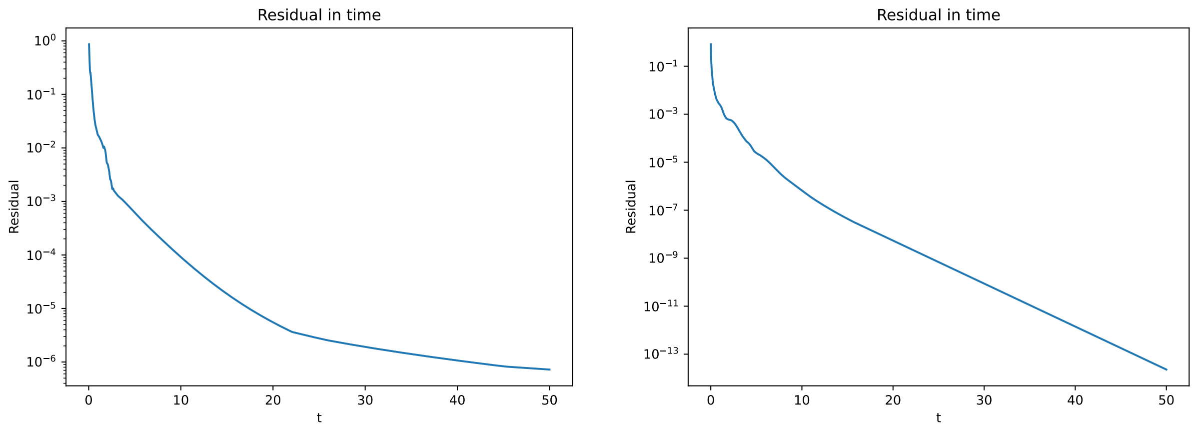

Figure 7: Time evolution of the L_2 norm of the residual for the flower problem for acoustics with GF-SUPG-FV-GPM (left) and GF-Rusanov-FV-GPM (right) schemes on grids with 80×80 cells.

GF-SUPG-FV-GPM and GF-Rusanov-FV-GPM schemes on a grid with 80×80 cells at time $T = 50$. The left figure shows the pressure field, the middle figure the norm of the velocity field and the right figure the velocity field represented with arrows. The first striking result is that the GF-Rusanov-FV - GPM method does no recast a constant pressure, which is quite surprising given the diffusion operator it owns. Secondly, we see that the maximum of the velocity field is essentially at the boundaries of the square for the GF-Rusanov-FV - GPM method, which is quite unrealistic and dictated by the strong diffusion of the method. Note the difference in the scales of the solutions.

On the other hand, the GF-SUPG-FV - GPM method recovers constant pressure up to 10^{-3} perturbations and also provides a more realistic velocity field distribution. The velocity stagnates close to the concavity at the left and at the right of the flower, while it accellerates close to convexities at the top and at the bottom of the flower, which is what we would expect from a potential flow around this obstacle.

The residuals in time in Figure 7 show that both the GF-Rusanov-FV - GPM and the GF-SUPG-FV - GPM methods reach some sort of numerical equilibrium, but clearly the GF-Rusanov-FV - GPM method has chosen a way less accurate equilibrium.

5. Conclusions

In this work, we have demonstrated how to extend the global flux methodology for multi-dimensional equilibria to non rectangular domains using the ghost point method. This requires the cheap solution of constraint least squares problems to compute the ghost point values, extrapolating information from the interior of the domain and from the boundary conditions defined at some collar points. The extension of this work are natural, in particular, we will look at the extension to nonlinear problems as the Euler equations, or Shallow Water equations and to 3D problems. Also, increasing the order of accuracy of the method is a goal of the authors, and this will involve both a high order reconstruction of the FV underlying scheme and a high order extrapolation of the ghost point values [18]. A finite element formulation can also be used to this end as in [7, 12].

References

[1] Wasilij Barsukow, Mirco Ciallella, Mario Ricchiuto, and Davide Torlo. Genuinely multi-dimensional stationarity preserving Finite Volume formulation for nonlinear hyperbolic PDEs. *Journal of Computational Physics*, page 114633, 2026.

[2] Wasilij Barsukow. Stationarity preservation and the low mach number behaviour of the discontinuous

galerkin method on cartesian grids, 2025.

[3] Keith William Morton and Philip L Roe. Vorticity-preserving Lax–Wendroff-type schemes for the system wave equation. *SIAM Journal on Scientific Computing*, 23(1):170–192, 2001.

[4] D. Sidilkover. Factorizable schemes for the equations of fluid flow. *Applied numerical mathematics*, 41(3):423–436, 2002.

[5] R. Jeltsch and M. Torrilhon. On curl-preserving finite volume discretizations for shallow water equations. *BIT Numerical Mathematics*, 46:35–53, 2006.

[6] W. Barsukow. Stationarity preserving schemes for multi-dimensional linear systems. *Mathematics of Computation*, 88(318):1621–1645, 2019.

[7] W. Barsukow, M. Ricchiuto, and D. Torlo. Structure preserving nodal continuous finite elements via global flux quadrature. *Numerical Methods for Partial Differential Equations*, 41(1):e23167, 2025.

[8] Ronald P Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of computational physics*, 152(2):457–492, 1999.

[9] Armando Coco and Giovanni Russo. Finite-difference ghost-point multigrid methods on Cartesian grids for elliptic problems in arbitrary domains. *Journal of Computational Physics*, 241:464–501, 2013.

[10] Armando Coco and Giovanni Russo. Boundary treatment in ghost point finite difference methods for compressible gas dynamics in domain with moving boundaries. *Hyperbolic Problems: Theory, Numerics, Applications*, page 455, 2012.

[11] Wasilij Barsukow, Mario Ricchiuto, and Davide Torlo. Stationarity preserving nodal Finite Element methods for multi-dimensional linear hyperbolic balance laws via a Global Flux quadrature formulation, 2025.

[12] Moussa Ziggaf, Davide Torlo, and Mario Ricchiuto. Arbitrary order stationarity preserving stabilized finite elements for multidimensional nonlinear hyperbolic problems. Application to the Euler equations with gravity. *arXiv preprint arXiv:2603.23185*, 2026.

[13] R. Abgrall and M. Ricchiuto. Hyperbolic balance laws: Residual distribution, local and global fluxes. In Dia Zeidan, Jochen Merker, Eric Goncalves Da Silva, and Lucy T. Zhang, editors, *Numerical Fluid Dynamics: Methods and Computations*, pages 177–222. Springer Nature Singapore, Singapore, 2022.

[14] E. Gaburro, M. Ricchiuto, and M. Dumbser. On general and complete multidimensional Riemann solvers for nonlinear systems of hyperbolic conservation laws, 2025. arXiv eprint, 2506.00207, math.na.

[15] W. Barsukow, Raphael Loubère, and Pierre-Henri Maire. A node-conservative vorticity preserving finite volume method for linear acoustics on unstructured grids. *Math. Comp.*, 94:2299–2343, 2025.

[16] Ricardo Costa, Stéphane Clain, Raphaël Loubère, and Gaspar J Machado. Very high-order accurate finite volume scheme on curved boundaries for the two-dimensional steady-state convection–diffusion equation with Dirichlet condition. *Applied Mathematical Modelling*, 54:752–767, 2018.

[17] Mirco Ciallella, Stephane Clain, Elena Gaburro, and Mario Ricchiuto. Very high order treatment of embedded curved boundaries in compressible flows: ADER discontinuous Galerkin with a space-time Reconstruction for Off-site data. *Computers & Mathematics with Applications*, 175:1–18, 2024.

[18] Alina Chertock, Armando Coco, Alexander Kurganov, and Giovanni Russo. A second-order finite-difference method for compressible fluids in domains with moving boundaries. *Communications in Computational Physics*, 23(1):230–263, 2018.

[19] M. Ricchiuto and D. Torlo. Analytical travelling vortex solutions of hyperbolic equations for validating very high order schemes. *arXiv preprint arXiv:2109.10183*, 2021.