Structure preserving nodal continuous Finite Elements via Global Flux quadrature

Wasilij Barsukow[1], Mario Ricchiuto[2], Davide Torlo[3]

[1] Institut de Mathématiques de Bordeaux (IMB), CNRS UMR 5251, 351 Cours de la Libération, 33405 Talence, France, wasilij.barsukow@math.u-bordeaux.fr

[2] INRIA, Univ. Bordeaux, CNRS, Bordeaux INP, IMB, UMR 5251, 200 Avenue de la Vieille Tour, 33405 Talence cedex, France, mario.ricchiuto@inria.fr

[3] Dipartimento di Matematica G. Castelnuovo, Università di Roma La Sapienza, piazzale Aldo Moro, 5, 00185, Rome, Italy, davide.torlo@uniroma1.it

Abstract

Numerical methods for hyperbolic PDEs require stabilization. For linear acoustics, divergence-free vector fields should remain stationary, but classical Finite Difference methods add incompatible diffusion that dramatically restricts the set of discrete stationary states of the numerical method. Compatible diffusion should vanish on stationary states, e.g. should be a gradient of the divergence. Some Finite Element methods allow to naturally embed this grad-div structure, e.g. the SUPG method or OSS. We prove here that the particular discretization associated to them still fails to be constraint preserving. We then introduce a new framework on Cartesian grids based on surface (volume in 3D) integrated operators inspired by Global Flux quadrature and related to mimetic approaches. We are able to construct constraint-compatible stabilization operators (e.g. of SUPG-type) and show that the resulting methods are vorticity-preserving. We show that the Global Flux approach is even super-convergent on stationary states, we characterize the kernels of the discrete operators and we provide projections onto them.

1 Introduction

1.1 Acoustic equations

This paper focuses on the discretization of hyperbolic PDEs. Although we have in mind applications to hyperbolic conservation laws such as the Euler or shallow water equations with source terms, we will work here in the much simpler setting of the linear wave equations in first-order form in the 2D and general forms:

$$\begin{cases} \partial_t u + \partial_x p = 0, \\ \partial_t v + \partial_y p = 0, \\ \partial_t p + \partial_x u + \partial_y v = 0, \end{cases} \qquad \begin{cases} \partial_t \mathbf{v} + \nabla p = 0, \\ \partial_t p + \nabla \cdot \mathbf{v} = 0, \end{cases} \tag{1}$$

for $u, v, p \colon \Omega \subset \mathbb{R}^2 \to \mathbb{R}$, and with, in 2-d, the notation $\mathbf{v} = (u, v)$. We also introduce the notation

$$\partial_t q + J^x \partial_x q + J^y \partial_y q = 0 \tag{2}$$

with

$$q = \begin{pmatrix} u \\ v \\ p \end{pmatrix}, \qquad J^x = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \qquad J^y = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}. \tag{3}$$

The system of linear acoustics possesses an involution:

$$\partial_t(\nabla \times \mathbf{v}) = 0, \tag{4}$$

which is reminiscent of involutions appearing e.g. in the (vacuum) Maxwell equation. The stationary states of linear acoustics are divergence-free, i.e.

$$\partial_t q \equiv 0 \Longleftrightarrow \nabla \cdot \mathbf{v} \equiv 0 \text{ and } p \equiv p_0 \in \mathbb{R}. \tag{5}$$

Both acoustics and Maxwell equations can be seen as toy-models for the more complex Euler equations and those of magnetohydrodynamics.

1.2 Structure-preserving Finite Difference methods

Numerical methods for hyperbolic problems require appropriate stabilization. It is introduced to obtain L^2 stability (or entropy stability in the non-linear case), or to manage discontinuous solutions. One way to stabilize is to add numerical dissipation. Many numerical methods for multi-dimensional hyperbolic problems contain stabilization initially derived in a one-dimensional setup. For instance, it is customary to compute the normal flux across an edge (or a face) by ignoring any signals propagating in the transverse direction, or emanating from the corners of the cell. This one-dimensional stabilization applied in different directions has a practical impact onto numerical solutions, e.g. on stationary states characterized by a balance of contributions from different directions (see e.g. [1]). The restriction of this datum onto one direction is not stationary, and unsurprisingly the combined numerical diffusion from the one-dimensional problems does not generally cancel out. One observes the datum being diffused away instead of being kept stationary.

Not every kind of multi-dimensional information, however, leads to a method with special properties and improved behavior. For example, in [2] the exact solution of the 4-quadrant Riemann problem has been used to derive a truly multi-dimensional Godunov method. Although it takes into account all the signals from corners, and generally makes no approximation in the evolution step, it fails to be stationarity preserving, or involution preserving.

In the context of linear acoustics on Cartesian grids, a numerical method based on one-dimensional Riemann solvers amounts to the following stabilization

$$\begin{aligned} \partial_t u + \partial_x p &= \frac{1}{2}\Delta x \partial_x^2 u + \text{h.o.t.}, \\ \partial_t v + \partial_y p &= \frac{1}{2}\Delta y \partial_y^2 v + \text{h.o.t.}, \\ \partial_t p + \partial_x u + \partial_y v &= \frac{1}{2}(\Delta x \partial_x^2 p + \Delta y \partial_y^2 p) + \text{h.o.t.}. \end{aligned} \tag{6}$$

One can show that $\partial_x u + \partial_y v = 0$ no longer remains stationary, unless $\partial_x u = 0$, $\partial_y v = 0$ individually. This has a dramatic impact on simulations, as setups that should remain stationary are now diffused, which can be understood as loss of consistency for long-time simulations. Numerical methods whose stationary states are described by a discretization of $\nabla \cdot \mathbf{v} = 0$ (without further constraints) are called *stationarity preserving* [1] and therefore possess a rich set of stationary states. One can show that the low Mach number limit of the Euler equations is related to the long-time limit of linear acoustics [3, 4] and that stationarity-preserving methods are also involution-preserving. Preservation of discrete involutions in the context of Maxwell equations usually is relevant for long-time stability and the correct coupling to matter.

An obvious approach to deriving a stationarity-preserving method is to ensure that the numerical diffusion of $\mathbf{v}$ is a function of the divergence. All the methods suggested in [5, 6, 7,

8, 9, 1] essentially imply

$$\begin{aligned}\partial_t u + \partial_x p &= \frac{1}{2}\Delta x \partial_x(\partial_x u + \partial_y v) + \text{h.o.t.},\\ \partial_t v + \partial_y p &= \frac{1}{2}\Delta y \partial_y(\partial_x u + \partial_y v) + \text{h.o.t.},\\ \partial_t p + \partial_x u + \partial_y v &= \frac{1}{2}(\Delta x \partial_x^2 p + \Delta y \partial_y^2 p) + \text{h.o.t.},\end{aligned} \tag{7}$$

i.e., the choice of $\nabla(\nabla \cdot \mathbf{v})$ as the appropriate diffusion for the evolution of $\mathbf{v}$. Observe that $\nabla p = 0$ and $\partial_x u + \partial_y v = 0$ now are again characterizing the stationary states, i.e. the stationary states of the PDE are exempt from the effect of numerical diffusion.

An interesting dichotomy thus appears to govern the field of truly multi-dimensional methods. While numerical methods that perform well are often derived ad-hoc (e.g. [10]), those with a first-principles derivation do not generally show improved behavior. Another example is [11], where a modified idea of how global conservation is related to local conservation yields a stationarity preserving method, but at one point an explicit choice is made without providing a fundamental reason for it. This points to a general lack of understanding how numerical methods for multi-dimensional problems should be derived and explains the interest in structure preserving numerical methods. Their improved performance in practice is another reason, of course. To provide a more general strategy to achieve stationarity preservation for stabilized continuous Finite Element methods is the aim of this paper.

1.3 Structure-preserving Finite Element methods and failure of SUPG

Finite element methods (FEM) are successful in achieving high order of accuracy and can be used on different types of computational grids. There exists a vast literature on structure preserving FEM, for instance among mixed FEM (e.g. [12, 13, 14]). Hyperbolic systems of conservation laws, however, require stabilization. This paper focuses on continuous Galerkin methods with an artificial diffusion and with the same choice of discretization space for scalars as for vector components. We consider to this end the Streamline-upwind Petrov-Galerkin (SUPG) [15, 16] and the Orthogonal Subscale Stabilization (OSS) [17, 18, 19, 20] approaches. Both methods allow to introduce the proper numerical diffusion structure. To see this for SUPG it is enough to recall that the method is obtained as the weak form of the modified equation

$$\partial_t q + J^x \partial_x q + J^y \partial_y q = \partial_x(\alpha h\, J_x(\partial_t q + J^x \partial_x q + J^y \partial_y q)) + \partial_y(\alpha h\, J_y(\partial_t q + J^x \partial_x q + J^y \partial_y q)) \tag{8}$$

where h is a characteristic mesh size, and α a stabilization parameter/matrix, which we consider constant in the following. For linear acoustics, when expanding the right-hand side we obtain the stabilized equations

$$\begin{cases}\partial_t u + \partial_x p = \partial_x(\alpha h(\partial_t p + \partial_x u + \partial_y v)),\\ \partial_t v + \partial_y p = \partial_y(\alpha h(\partial_t p + \partial_x u + \partial_y v)),\\ \partial_t p + \partial_x u + \partial_y v = \partial_x(\alpha h(\partial_t u + \partial_x p)) + \partial_y(\alpha h(\partial_t v + \partial_y p)).\end{cases} \tag{9}$$

Beside the time derivatives, the stabilization terms are essentially variational approximations of the grad-div operator for the velocity stabilization, and of a standard Laplacian for the pressure equation. This approach thus seems to produce, besides some mixed space-time derivatives, exactly the terms in (7). One might thus expect that this numerical method will be stationarity preserving and therefore vorticity preserving [1]. However, as we will show in Section 3 as well as in the numerical tests, this is not the case. The reason for this is related to the fact that the implication

$$\partial_x u + \partial_y v \equiv 0 \qquad \Rightarrow \qquad \partial_x^2 u + \partial_x \partial_y v \equiv 0 \tag{10}$$

is not true in the discrete:

$$\int \varphi(\partial_x u + \partial_y v)\mathrm{d}x \equiv 0 \ \forall \varphi \in V_h^K \qquad \not\Rightarrow \qquad \int \partial_x \varphi(\partial_x u + \partial_y v)\mathrm{d}x = 0 \ \forall \varphi \in V_h^K. \tag{11}$$

where, on a given Cartesian tessellation of the spatial domain, we denote by V_h^K is the K-th degree finite element approximation space.

1.4 Restoring stationarity preservation via Global Flux quadrature

Flux globalization dates back to the work of [21, 22] in the context of approximations of balance laws

$$\partial_t q + \partial_x F = S(q, x)$$

For this problem a relevant aspect is the super-convergent (or even exact) approximation of non-trivial steady states. To this end, one can write the source term as a flux R which is the primitive of the source term:

$$R = R_0 - \int_{x_0}^{x} S(q(s,t), s))ds.$$

In this setting, discrete steady equilibria verify the relation

$$\partial_x(F + R) = 0 \Rightarrow F + R = G_0 \in \mathbb{R}$$

with $G = F + R$ the so-called global flux.

Following [23], we can now construct a finite element approximation R_h of the flux R which is, just as S, in the space V_h^K of polynomials of degree K (despite being a primitive of S), by using an integral operator I_x which, in FEM, is local for each cell. This leads to

$$R_\alpha = R_0 - \int_{x_0}^{x_\alpha} S_h(x)\mathrm{d}x, \ \forall \alpha \iff R = R_0 - I_x S, \tag{12}$$

with R_α the degrees of freedom of R_h and the right notation is a vectorial version of the left one.

The continuous SEM approximation of the balance law (with periodic BCs or neglecting BCs) can be succinctly written as

$$M_x \frac{dq}{dt} + D_x F - D_x I_x S = 0. \tag{13}$$

where M_x is the mass matrix, D_x is a finite element weak derivative matrix and I_x is the integrator localized in each cell (see Section 4.2 for precise definitions of the notation). In this approach, we have replaced the mass matrix M_x in front of the nodal values of the source, with the product matrix $D_x I_x$. This modification allows to factor the derivative matrix, so that at steady state the scheme reduces to

$$F = F_0 + I_x S\,.$$

The integrator naturally turns out to be the ODE solver associated to the table I_x applied to the flux ODE [23]

$$F' = S(q(F), x)\,.$$

This provides a clear characterization of the discrete steady state and gives a so-called approximate or discrete well balanced principle, in the spirit of e.g. [24, 25]. The integration table I_x defines the properties of the ODE solver. For Lagrangian basis functions on Gauss-Lobatto points, the well known LobattoIIIA methods arise [26, 27]. This method has nodal consistency of order h^{K+2} at internal nodes, and h^{2K} at end-nodes, thus leading to a super-convergent method at steady state.

Due to the fully local structure of the method, and to the fact that it merely involves a particular approximation of the weighted source integral $\int \varphi S$, the authors of [23] proposed to refer to it as *global flux quadrature* (GFq). In this work we propose a genuine generalization of the above idea to multi-dimensional equilibria. When considering the last equation in (1), we observe that it can be written in two ways

$$\partial_t p + \partial_x u + \partial_y v = \partial_t p + \partial_x u + \partial_x \left(\int_{x_0}^{x} \partial_y v \, \mathrm{d}s \right) = \partial_t p + \partial_y v + \partial_y \left(\int_{y_0}^{y} \partial_x u \, \mathrm{d}s \right) = 0.$$

We propose to couple the x and y derivatives by symmetrizing the two directional global flux quadrature formulations as

$$\partial_t p + \partial_x \partial_y \left(\int_{y_0}^{y} \partial_x u \, \mathrm{d}s + \int_{x_0}^{x} \partial_y v \, \mathrm{d}s \right) = 0. \tag{14}$$

In this paper we combine this idea with a high-order grad-div stabilized continuous Finite Element approximations leading to stationarity preserving methods.

1.5 Overview of the paper

The paper is structured as follows: Section 2 introduces a difference/matrix notation for tensor-product FEM spaces on Cartesian grids that is used subsequently. In Section 3, we prove that classical grad-div stabilizations (such as SUPG and OSS) are in general not constraint preserving: the kernels of the stabilizing term and of the Galerkin term do not have a sufficiently large intersection. Global Flux gives rise to a discretization of the divergence different from the one of continuous FEM; it is analyzed in Section 4, where in particular exact projections in the discrete kernel space and nodal consistency estimates and the super-convergent behavior are provided (Section 4.3). The Global Flux approach is applied in Sections 5.1 and 5.2 to construct constraint-compatible SUPG and OSS stabilizations. In Section 5.3, spurious modes in the kernels of the discrete operators are studied. In Section 5.4, curl involutions are characterized using Fourier symbols, and explicit formulas are provided in the $\mathbb{Q}^1$ case. The time discretization is described in Section 6 and numerical results follow in Section 7.

2 Cartesian grids, tensor products, and discrete Fourier transform for Finite Elements

2.1 General definitions

2.1.1 One-dimensional Finite Element spaces

The study shall be restricted to Cartesian grids. We consider two one-dimensional domains $\Omega^x, \Omega^y \subset \mathbb{R}$ and their product $\Omega := \Omega^x \times \Omega^y$. We define a uniform tessellation of each one-dimensional domain $\Omega^x_{\Delta x} = \cup_{i=0}^{N_x-1} E_i^x$, $\Omega^y_{\Delta y} = \cup_{j=0}^{N_y-1} E_j^y$, of elements $E_i^x = [x_i, x_{i+1}], E_j^y = [y_j, y_{j+1}]$ with $|E_i^x| = \Delta x$ and $|E_j^y| = \Delta y$ for all i, j.

To define the continuous Finite Element spaces, we introduce $K+1$ points in each one-dimensional cell

$$x_{i,p} = \hat{x}_p \Delta x + x_i \in E_i^x, \text{ for } p = 0, \dots, K, \text{ and } y_{j,\ell} = \hat{x}_\ell \Delta y + y_j \in E_j^y, \text{ for } \ell = 0, \dots, K$$

that we will use to define the Lagrangian basis functions. In particular, here we will consider points $\{\hat{x}_p\}_{p=0}^K \subset [0,1]$ with $\hat{x}_0 = 0 < \cdots < x_i < \cdots < \hat{x}_K = 1$, e.g. those of Gauss–Lobatto, such that $x_{i,0} = x_{i-1,K}$ for $i = 1, \dots, N_x - 1$ and similarly for y.

We introduce a unique numbering for the point p in cell i with a Greek alphabet index $\alpha := iK + p \in [0, N_x K]$, $p \in [0, K-1]$, so that we will refer uniquely to point $x_\alpha = x_{i,p}$. Let us

define by $M_x + 1 = N_x K + 1$ and $M_y + 1 = N_y K + 1$ the number of points in each direction. There are on average $K - 1$ points per cell, but each cell has access to K points. We will switch between these two notations according to our needs.

We can now introduce the continuous Finite Element spaces of degree K over one/two-dimensional domains as

$$V_{\Delta x} := V_{\Delta x}^K(\Omega_{\Delta_x}^x) := \left\{ q \in \mathcal{C}^0(\Omega_{\Delta x}^x) : q|_E \in \mathbb{P}^K(E), \forall E \in \Omega_{\Delta x}^x \right\}, \tag{15a}$$

$$V_{\Delta y} := V_{\Delta y}^K(\Omega_{\Delta_y}^y) := \left\{ q \in \mathcal{C}^0(\Omega_{\Delta y}^y) : q|_E \in \mathbb{P}^K(E), \forall E \in \Omega_{\Delta y}^y \right\}, \tag{15b}$$

where we denote by $\mathbb{P}^K$ the space of univariate polynomials of degree at most K and by $\mathbb{Q}^K$ the space of multivariate polynomials of degree at most K in each variable.

In particular, we choose as basis of these spaces the high order hat functions that interpolate the points defined above. In each one-dimensional cell E_i^x, we consider $\varphi_{i,p}^x(x) \in V_{\Delta x}$ such that $\varphi_{i,p}^x|_{E_i^x}(x) \in \mathbb{P}^K(E_i^x)$ and $\varphi_{i,p}^x(x_{i,\ell}) = \delta_{j,\ell}$ for all $\ell, p = 0, \dots, K$, with δ the Kronecker delta. Moreover, since φ must be continuous, we have

$$\text{supp}(\varphi_{i,p}^x) = E_i^x \text{ for } p = 1, \dots, K-1, \quad \text{supp}(\varphi_{i,0}^x) = E_{i-1}^x \cup E_i^x \text{ and } \text{supp}(\varphi_{i,K}^x) = E_i^x \cup E_{i+1}^x,$$

recalling that $\varphi_{i-1,K}^x = \varphi_{i,0}^x$. The same holds for the basis functions of the y space. Moreover, $V_{\Delta x}^K(\Omega_{\Delta x}^x) = \text{span}\{\varphi_\alpha^x\}_{\alpha=0}^{M_x}$ with $\varphi_\alpha^x(x) \equiv \varphi_{i,p}^x(x)$ for $\alpha = iK + p$. We use the same spaces to discretize vector components as those we use for scalars.

2.1.2 Tensor-product Finite Element spaces

We define the two dimensional tessellation of Ω as

$$\Omega_h = \bigcup_{i,j=0}^{N_x-1,N_y-1} E_{ij} \tag{16}$$

with $h = \min\{\Delta x, \Delta y\}$ and $E_{ij} := E_i^x \times E_j^y$.

This leads to the definition of V_h as a tensor product of the functional spaces

$$V_h := V_h^K(\Omega_h) := \left\{ q \in \mathcal{C}^0(\Omega_h) : q|_E \in \mathbb{Q}^K(E), \forall E \in \Omega_h \right\}, \tag{17}$$

which is evident in its basis $\{\varphi_{\alpha;\beta}\}_{\alpha,\beta=0}^{M_x,M_y}$ with $\varphi_{\alpha;\beta}(x,y) := \varphi_\alpha^x(x)\varphi_\beta^y(y)$. Finally, we will describe a function $q \in V_{\Delta x}$ as

$$q_h(x) = \sum_{\alpha=0}^{M_x} q_\alpha \varphi_\alpha^x(x) = \sum_{i=0}^{N_x-1} \sum_{p=0}^{K} q_{i,p}\, \varphi_{i,p}|_{E_i^x}(x) \tag{18}$$

and a function $q \in V_h$ as

$$q_h(x,y) = \sum_{\alpha=0;\beta=0}^{M_x;M_y} q_{\alpha;\beta}\varphi_{\alpha;\beta}(x,y) = \sum_{i=0;j=0}^{N_x;N_y} \sum_{p=0;\ell=0}^{K;K} q_{i,p;j,\ell}\, \varphi_{i,p}^x|_{E_i^x}(x)\, \varphi_{j,\ell}^y|_{E_j^y}(y). \tag{19}$$

See Figure 1 for a graphical representation of degrees of freedom (DOFs) $q_{i,p;j,\ell}$ in a cell E_{ij} for $K = 4$.

In the following, we will give a more Finite Difference flavored description of classical FEM operators, in order to introduce Fourier symbols.

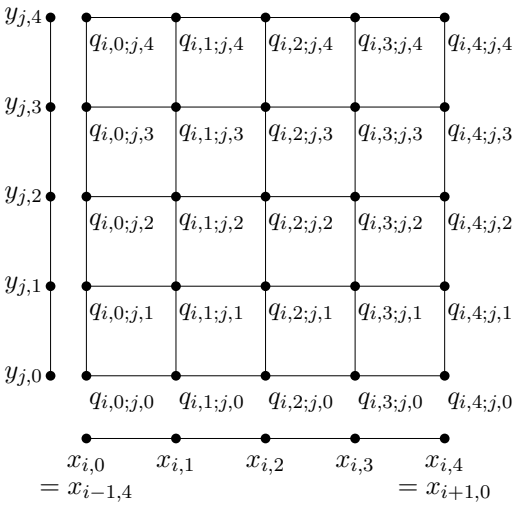

Figure 1: Notation of the degrees of freedom for a function q_h in element E_{ij} for $\mathbb{Q}^4$ elements

2.2 Bridging finite element and uni-directional difference formulae

We start with a definition which can be applied to $\mathbb{P}^1$ FEM and finite differences which can be mapped onto each other. We assume periodic boundary conditions for simplicity.

Definition 2.1 (Finite differences)**.** *Consider a one-dimensional equidistant grid with values $(q_i)_{i\in\mathbb{Z}}$ and a linear unidirectional finite difference formula*

$$(Dq)_i = \sum_{k\in\mathbb{Z}} \alpha_k q_{i+k}. \tag{20}$$

- *Define $k_{max} \in \mathbb{N}^0$ as the smallest value for which the sum can be restricted*

$$(Dq)_i \equiv \sum_{k=-k_{max}}^{k_{max}} \alpha_k q_{i+k}. \tag{21}$$

- *We call $D\colon \mathbb{R}^{\mathbb{Z}} \to \mathbb{R}^{\mathbb{Z}}$ whose action on q at i is defined by* (20) *the* finite difference operator*.*
- *We call a finite difference formula* compact *if $\operatorname{supp}\alpha \subset \mathbb{Z}$ is finite, i.e., if the sum in* (20) *is finite. We shall also call the corresponding finite difference operator* compact *in this case.*
- *The* characteristic polynomial $\mathbb{F}_{t_x}(D)$ *of D is the univariate Laurent polynomial $\sum_{k\in\mathbb{Z}} \alpha_k t_x^k$ in t_x.*

Up to prefactors, the discrete Fourier transform (i.e. inserting $q_i := \exp(\mathbb{i}k_x i\Delta x)$) of $(Dq)_i$ lets appear precisely its characteristic polynomial if one defines $t_x = \exp(\mathbb{i}k_x\Delta x)$ (see e.g. [1]).

Throughout the paper we use arbitrarily high-order methods, discussed now. In the context of Galerkin methods, for $K > 1$, different degrees of freedom are involved.

Example 2.1. *The following derivative operator evaluated in the DoF $\beta = (i, s)$ reads*

$$(D_x q)_{i,s} = \int \varphi^x_{i,s}(x)\partial_x q_h(x)\mathrm{d}x = \sum_{j=0}^{N_x-1} \int_{E^x_j} \varphi^x_{i,s}(x) \sum_{p=0,K} \partial_x \varphi^x_{j,p}(x) q_{j,p} \tag{22}$$

$$\equiv \sum_{k\in\{-1,0,1\}} \sum_{p=0}^{K-1} q_{i+k,p} \int_{E^x_{i+k}} \varphi^x_{i,s}(x)\varphi^x_{i+k,p}(x)\mathrm{d}x, \tag{23}$$

Due to translation invariance the right-hand side integral is going to depend only on k, s and p, not on i.

This motivates the following

Definition 2.2 (High-order differences)**.** *On a one-dimensional equidistant grid, embedding repeated sets of not necessarily equidistant collocation points, consider a linear, high-order unidirectional difference formula*

$$(Dq)_{i,s} = \sum_{k\in\mathbb{Z}} \sum_{p=1}^{K} \alpha_{k,p}^{s} q_{i+k,p} \tag{24}$$

The fact that the collocation points are repeated implies translational invariance, which allows to choose α without a dependence on i. Here, $q_{i+k,p}$ are the same as in (18)*.*

- *We call $D : \mathbb{R}^{\mathbb{Z}} \times [1,K] \to \mathbb{R}^{\mathbb{Z}} \times [1,K]$ the* high-order difference operator*. Observe that the operator has K components (in function of which basis element the expression is tested against).*
- *Define $k_{max} \in \mathbb{N}^0$ to be the smallest integer for which one can restrict the summation:*

$$(Dq)_{i,s} \equiv \sum_{k=-k_{max}}^{k_{max}} \sum_{p=1}^{K} \alpha_{k,p}^{s} q_{i+k,p}. \tag{25}$$

  As is obvious from (23)*, for the usual operators appearing in FEM, $k_{max} = 1$.*

- *The* characteristic polynomial *$\mathbb{F}_{t_x}(D)$ of D is the matrix of univariate Laurent polynomials in t_x*

$$\mathbb{F}_{t_x}(D) := \begin{pmatrix} \sum_{k\in\mathbb{Z}} \alpha_{k,1}^{1} t_x^k & \cdots & \sum_{k\in\mathbb{Z}} \alpha_{k,K}^{1} t_x^k \\ \vdots & \ddots & \vdots \\ \sum_{k\in\mathbb{Z}} \alpha_{k,1}^{K} t_x^k & \cdots & \sum_{k\in\mathbb{Z}} \alpha_{k,K}^{K} t_x^k \end{pmatrix}, \qquad \textit{i.e., } \mathbb{F}_{t_x}(D)_{s,p} = \sum_{k\in\mathbb{Z}} \alpha_{k,p}^{s} t_x^k. \tag{26}$$

  (We still call this matrix a "polynomial" because it can be considered a polynomial in t_x with matrix-valued coefficients.)

Example 2.2. *Consider the mass matrix appearing in*

$$\Delta x (M_x q)_{i,s} = \int \varphi_{i,s}^{x}(x) q_h(x) \mathrm{d}x = \sum_{k=0}^{N_x-1} \sum_{p=1}^{K} \left(\int_{E_k^x} \varphi_{i,s}^{x}(x) \varphi_{k,p}^{x}(x) \mathrm{d}x \right) q_{k,p}. \tag{27}$$

Then, by comparison with (24) *one finds (using the translational invariance)*

$$\alpha_{k,p}^{s} = \int_{\mathbb{R}} \varphi_{0,s} \varphi_{k,p}(x) \mathrm{d}x. \tag{28}$$

Definition 2.3 (Composition in the high-order case)**.** *Given two high-order unidirectional difference formulas on a one-dimensional grid*

$$(Aq)_{i,r} = \sum_{k\in\mathbb{Z}} \sum_{s=1}^{K} \alpha_{k,s}^{r} q_{i+k,s} \qquad\qquad (Bq)_{i,r} = \sum_{k\in\mathbb{Z}} \sum_{s=1}^{K} \beta_{k,s}^{r} q_{i+k,s}, \tag{29}$$

we define the composition *AB of the high-order difference operators A and B by*

$$((AB)q)_{i,r} := \sum_{k\in\mathbb{Z}} \sum_{s=1}^{K} \alpha_{k,s}^{r} (Bq)_{i+k,s} = \sum_{k\in\mathbb{Z}} \sum_{s=1}^{K} \sum_{k'\in\mathbb{Z}} \sum_{s'=1}^{K} \alpha_{k,s}^{r} \beta_{k',s'}^{s} q_{i+k+k',s'}. \tag{30}$$

Proposition 2.1. *The characteristic polynomial* $(\mathbb{F}_{t_x}(RS))_{r,s}$ *of the composition of two high-order difference operators* R *and* S *is the (matrix) product* $\sum_{p=1}^{K}(\mathbb{F}_{t_x}(R))_{r,p}(\mathbb{F}_{t_x}(S))_{p,s}$. *of their characteristic polynomials.*

The proof is given in Appendix A.1. In the high-order case the composition is not commutative, as can easily be seen from the fact that the associated operation on the characteristic polynomials is a matrix product.

2.3 Tensor products and multidirectional difference formulae

Consider now a 2-dimensional Cartesian grid as described in Section 2 with $q_h \in V_h^K$. We consider the following generalized high order finite differences in this context.

Obviously, tensor based FEM allows to factor one dimensional operators. For example for the mass matrix one has

$$\begin{aligned}\iint \varphi^x_{i,s}(x)\varphi^y_{j,p}(y)q_h(x,y)\mathrm{d}x\mathrm{d}y &= \sum_{E_{k\ell}\in\Omega_h}\sum_{r,t=0}^{K}\left(\iint_{E_{k\ell}}\varphi^x_{i,s}(x)\varphi^y_{j,p}(y)\varphi^x_{k,r}(x)\varphi^y_{\ell,t}(y)\mathrm{d}x\mathrm{d}y\right)q_{k,r;\ell,t}\\ &= \sum_{E^x_k\in\Omega^x_{\Delta x}}\sum_{r,t=0}^{K}\left(\int_{E^x_k}\varphi^x_{i,s}(x)\varphi^x_{k,r}(x)\mathrm{d}x\right)\sum_{E^y_\ell\in\Omega^y_{\Delta y}}\left(\int_{E^y_\ell}\varphi^y_{j,p}(y)\varphi^y_{\ell,t}(y)\mathrm{d}y\right)q_{k,r;\ell,t}\\ &= \sum_{k=0}^{N_x-1}\sum_{\ell=0}^{N_y-1}\sum_{s,p=0}^{K}q_{i+k,s;j+\ell,p}\int_{E^x_k}\varphi^x_{i,r}(x)\varphi^x_{i+k,s}(x)\mathrm{d}x\int_{E^y_\ell}\varphi^y_{j,t}(y)\varphi^y_{j+\ell,p}(y)\mathrm{d}y.\end{aligned}$$

This motivates the following definition.

Definition 2.4 (Tensor-product high-order operators)**.** *Consider two high-order unidirectional difference formulas on 1-dimensional Cartesian grids*

$$(Au)_{i,r} = \sum_{k\in\mathbb{Z}}\sum_{s=1}^{K}\alpha^r_{k,s}u_{i+k,s} \qquad\qquad (Bv)_{j,t} = \sum_{\ell\in\mathbb{Z}}\sum_{p=1}^{K}\beta^t_{\ell,p}v_{j+\ell,p}. \tag{31}$$

- *Then, the linear bidirectional high-order difference formula applied on* $q \in V_h^K$

$$((A\otimes B)q)_{i,r;j,t} := \sum_{(k,\ell)\in\mathbb{Z}^2}\sum_{s,p=1}^{K}\alpha^r_{k,s}\beta^t_{\ell,p}q_{i+k,s;j+\ell,p} \tag{32}$$

  is said to be the difference formula associated to the tensor product $A\otimes B$ *of the difference operators* A *and* B.

- *The* characteristic polynomial $\mathbb{F}_{t_x,t_y}(A\otimes B)$ *of a high-order difference operator* $A\otimes B$ *is the following matrix of bivariate Laurent polynomials in* t_x, t_y

$$(\mathbb{F}_{t_x,t_y}(A\otimes B))_{r,t} := \sum_{(k,\ell)\in\mathbb{Z}^2}\sum_{s,p=1}^{K}\alpha^r_{k,s}\beta^t_{\ell,p}t_x^k t_y^\ell. \tag{33}$$

- *The composition of two tensor-product high-order operators* $R := A\otimes B$, $S := C\otimes D$ *is defined as*

$$(RS)q := R(Sq). \tag{34}$$

Proposition 2.2 (Fourier transform of the tensor product in the high-order case)**.** *The characteristic polynomial $\mathbb{F}_{t_x,t_y}(A\otimes B)$ of $A\otimes B$ is the standard Kronecker product of the polynomials of A and B:*

$$\mathbb{F}_{t_x,t_y}(A\otimes B)=\mathbb{F}_{t_x}(A)\otimes\mathbb{F}_{t_y}(B). \tag{35}$$

Proof. Using the definition, we obtain

$$\mathbb{F}_{t_x,t_y}(A\otimes B)_{r,z}=\sum_{(k,\ell)\in\mathbb{Z}^2}\sum_{s,p=1}^{K}\alpha^r_{k,s}\beta^z_{\ell,p}t_x^k t_y^\ell=\sum_{k\in\mathbb{Z}}\sum_{s=1}^{K}\alpha^r_{k,s}t_x^k\sum_{\ell\in\mathbb{Z}}\sum_{p=1}^{K}\beta^z_{\ell,p}t_y^\ell, \tag{36}$$

which is the statement of the theorem. □

Proposition 2.3. *Consider high-order unidirectional difference formulas on a two-dimensional grid*

$$\begin{aligned}(A^x u)_{i,r}&=\sum_{k\in\mathbb{Z}}\sum_{s=1}^{K}(\alpha^x)^r_{k,s}u_{i+k,s},\qquad (A^y v)_{j,t}=\sum_{\ell\in\mathbb{Z}}\sum_{p=1}^{K}(\alpha^y)^t_{\ell,p}v_{j+\ell,p},\\ (B^x u)_{i,r}&=\sum_{k\in\mathbb{Z}}\sum_{s=1}^{K}(\beta^x)^r_{k,s}u_{i+k,s},\qquad (B^y v)_{j,t}=\sum_{\ell\in\mathbb{Z}}\sum_{p=1}^{K}(\beta^y)^t_{\ell,p}q_{j+\ell,p}.\end{aligned} \tag{37}$$

The composition of tensor products is the tensor product of compositions:

$$(A^x\otimes A^y)(B^x\otimes B^y)=(A^xB^x)\otimes(A^yB^y). \tag{38}$$

The proof can be found in the Appendix A.2.

In the paper we use several operators which are listed hereafter for completeness:

$$(M_x)_{\alpha,\beta}:=\int_{\mathbb{R}}\varphi^x_\alpha(x)\varphi^x_\beta(x)\mathrm{d}x,\qquad (\mathbb{1}_x)_{\alpha,\beta}=\delta_{\alpha,\beta}, \tag{39a}$$

$$(D_x)_{\alpha,\beta}:=\int_{\mathbb{R}}\varphi^x_\alpha(x)\partial_x\varphi^x_\beta(x)\mathrm{d}x,\qquad (D^x)_{\alpha,\beta}:=\int_{\mathbb{R}}\partial_x\varphi^x_\alpha(x)\partial_x\varphi^x_\beta(x)\mathrm{d}x, \tag{39b}$$

$$(D^x_x)_{\alpha,\beta}:=\int_{\mathbb{R}}\varphi^x_\alpha(x)\partial_x\varphi^x_\beta(x)\mathrm{d}x. \tag{39c}$$

3 Failure of standard grad-div stabilizations for acoustics

3.1 SUPG stabilization

The stabilized variational form of the SUPG method by Hughes and collaborators [15, 16] can be written as

$$\int\varphi(\partial_t q+J^x\partial_x q+J^x\partial_y q)dx+\int\alpha h(J^x\partial_x\varphi+J^y\partial_y\varphi)(\partial_t q+J^x\partial_x q+J^y\partial_y q)=\mathsf{B.C.s} \tag{40}$$

with α a stabilization constant/matrix and h a reference mesh size. The stability of the method can be shown by replacing the test function φ by $q+\alpha h\,q_t$ for constant α, (neglecting boundary condition terms, see also [28]). The natural energy norm of SUPG given by

$$E_{\mathrm{SUPG}}:=\int\left\{\frac{q^Tq}{2}+(\alpha\,h)^2(J^x\partial_x q+J^y\partial_y q)^T(J^x\partial_x q+J^y\partial_y q)\right\}dx$$

such that

$$\partial_t E_{\mathrm{SUPG}}=-\int\alpha\,h(\partial_t q+J^x\partial_x q+J^x\partial_y q)^T(\partial_t q+J^x\partial_x q+J^y\partial_y q)dx\le 0.$$

As said in Section 1, the method naturally includes a grad-div structure in the stabilization. It thus seems to fit exactly the framework of stationarity preserving methods. However, it actually fails to retain such a property. To show it we exploit the finite element/differences bridge presented in the previous sections, and follow the spectral analysis of [1].

Recall the notation of difference operators in (39) and that $D_x = -D^x$ up to boundary conditions. We can now write (40) as

$$\begin{pmatrix} M_x \otimes M_y & 0 & \alpha\, hD^x \otimes M_y \\ 0 & M_x \otimes M_y & \alpha\, hM_x \otimes D^y \\ \alpha\, hD^x \otimes M_y & \alpha\, hM_x \otimes D^y & M_x \otimes M_y \end{pmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{pmatrix} u \\ v \\ p \end{pmatrix} + \mathcal{E}_{SUPG} \begin{pmatrix} u \\ v \\ p \end{pmatrix} = 0 \tag{41}$$

having introduced the evolution operator

$$\mathcal{E}_{\mathrm{SUPG}} := \begin{pmatrix} \alpha\, hD_x^x \otimes M_y & \alpha\, hD^x \otimes D_y & D_x \otimes M_y \\ \alpha\, hD_x \otimes D^y & \alpha\, hM_x \otimes D_y^y & M_x \otimes D_y \\ D_x \otimes M_y & M_x \otimes D_y & \alpha\, hD_x^x \otimes M_y + \alpha\, hM_x \otimes D_y^y \end{pmatrix}. \tag{42}$$

Let us split the matrices defined above into the central discretization and the stabilization (streamline upwinding) denoting by $\mathcal{A}_{\mathrm{SUPG}} := \mathcal{A}_C + \mathcal{A}_{\mathrm{SU}}$ the matrix in front of the time derivative term, with

$$\mathcal{A}_C := \begin{pmatrix} M_x \otimes M_y & 0 & 0 \\ 0 & M_x \otimes M_y & 0 \\ 0 & 0 & M_x \otimes M_y \end{pmatrix}, \mathcal{A}_{\mathrm{SU}} := \alpha\, h \begin{pmatrix} 0 & 0 & D^x \otimes M_y \\ 0 & 0 & M_x \otimes D^y \\ D^x \otimes M_y & M_x \otimes D^y & 0 \end{pmatrix}, \tag{43a}$$

as well as for the matrix $\mathcal{E}_{\mathrm{SUPG}} = \mathcal{E}_C + \mathcal{E}_{\mathrm{SU}}$ with

$$\mathcal{E}_C := \begin{pmatrix} 0 & 0 & D_x \otimes M_y \\ 0 & 0 & M_x \otimes D_y \\ D_x \otimes M_y & M_x \otimes D_y & 0 \end{pmatrix}, \tag{43b}$$

$$\mathcal{E}_{\mathrm{SU}} := \alpha\, h \begin{pmatrix} D_x^x \otimes M_y & D^x \otimes D_y & 0 \\ D_x \otimes D^y & M_x \otimes D_y^y & 0 \\ 0 & 0 & D_x^x \otimes M_y + M_x \otimes D_y^y \end{pmatrix}. \tag{43c}$$

This way, (41) can be rewritten for $q = (u, v, p)^T$ as

$$0 = \mathcal{A}_{\mathrm{SUPG}} \frac{\mathrm{d}}{\mathrm{d}t} q + \mathcal{E}_{\mathrm{SUPG}} q = (\mathcal{A}_C + \mathcal{A}_{\mathrm{SU}}) \frac{\mathrm{d}}{\mathrm{d}t} q + (\mathcal{E}_C + \mathcal{E}_{\mathrm{SU}}) q. \tag{43d}$$

To be noted that $\mathcal{A}_C$ and $\mathcal{E}_{SU}$ are symmetric positive (semi-)definite, while $\mathcal{A}_{SU}$ and $\mathcal{E}_C$ are anti-symmetric matrices.

Consider now the lowest-order SUPG with $\mathbb{Q}^1$ basis functions. As there is only one degree of freedom per cell, SUPG can be immediately interpreted as a finite difference method, and its properties can be analyzed using techniques from [1]. Assuming for simplicity that $\Delta x = \Delta y = h$ and recalling that the quadrature formula and the Lagrangian basis functions are defined with the same Gauss-Lobatto points, we have

$$\mathbb{F}_{t_x}(M_x) = 1, \qquad \mathbb{F}_{t_x}(D_x) = -\mathbb{F}_{t_x}(D^x) = \frac{t_x^2 - 1}{2t_x\, h}, \qquad \mathbb{F}_{t_x}(D_x^x) = -\frac{(t_x - 1)^2}{t_x\, h^2}. \tag{44}$$

Then,

$$\mathbb{F}_{t_x,t_y}(\mathcal{E}_{\mathrm{SUPG}}) = \begin{pmatrix} -\alpha\frac{(t_x-1)^2}{ht_x} & -\alpha\frac{t_x^2-1}{2t_x}\frac{t_y^2-1}{2ht_y} & \frac{t_x^2-1}{2ht_x} \\ -\alpha\frac{t_x^2-1}{2ht_x}\frac{t_y^2-1}{2t_y} & -\alpha\frac{(t_y-1)^2}{ht_y} & \frac{t_y^2-1}{2ht_y} \\ \frac{t_x^2-1}{2ht_x} & \frac{t_y^2-1}{2ht_y} & -\alpha\frac{(t_x-1)^2}{ht_x} - \alpha\frac{(t_y-1)^2}{ht_y} \end{pmatrix}. \tag{45}$$

As shown in [1], all non-trivial stationary states are given as the right kernel of $\mathcal{E}$, while its left kernel gives the corresponding involutions (if any). However note now that

$$\det \mathbb{F}_{t_x,t_y}(\mathcal{E}_{\mathrm{SUPG}}) = \alpha(t_x-1)^2(t_y-1)^2(\ldots) \neq 0, \tag{46}$$

i.e., its kernel is trivial unless $t_x = 1$ or $t_y = 1$ (functions are constant in x or y) or $\alpha = 0$ (no stabilization). This method does not have non-trivial stationary states, and for the same reason also no discrete involutions. Without proof we note that the same result holds for the $\mathbb{Q}^2$ case.

A more general characterization, still for the $\mathbb{Q}^1$-case, can be obtained observing that

$$\begin{aligned}\det \mathbb{F}_{t_x,t_y}(\mathcal{E}_{\mathrm{SUPG}}) = \alpha^3 h^3 \Big(&\mathbb{F}_{t_y}(D_y^y)\mathbb{F}_{t_x}(M_x) + \mathbb{F}_{t_x}(D_x^x)\mathbb{F}_{t_y}(M_y)\Big)\times \\ &\Big(\mathbb{F}_{t_x}(D_x^x)\mathbb{F}_{t_y}(D_y^y)\mathbb{F}_{t_x}(M_x)\mathbb{F}_{t_y}(M_y) - \mathbb{F}_{t_x}(D_x)^2\mathbb{F}_{t_y}(D_y)^2\Big) \\ &- \alpha h \mathbb{F}_{t_x}(M_x)\mathbb{F}_{t_y}(M_y)\Big(\Big(\mathbb{F}_{t_x}(D_x^x)\mathbb{F}_{t_x}(M_x) + \mathbb{F}_{t_x}(D_x)^2\Big)\mathbb{F}_{t_y}(D_y)^2 + \\ &\Big(\mathbb{F}_{t_y}(D_y^y)\mathbb{F}_{t_y}(M_y) + \mathbb{F}_{t_y}(D_y)^2\Big)\mathbb{F}_{t_x}(D_x)^2\Big),\end{aligned} \tag{47}$$

which vanishes if

$$\mathbb{F}_{t_x}(D_x^x)\mathbb{F}_{t_x}(M_x) = -\mathbb{F}_{t_x}(D_x)^2 \quad \text{and} \quad \mathbb{F}_{t_y}(D_y^y)\mathbb{F}_{t_y}(M_y) = -\mathbb{F}_{t_y}(D_y)^2, \tag{48}$$

i.e.,

$$D_x^x M_x = -D_x^2 \quad \text{and} \quad D_y^y M_y = -D_y^2. \tag{49}$$

Note that for $\mathbb{Q}^1$ FEM discussed here, composition of difference operators is commutative. For general FEM, we have the following result.

Proposition 3.1. *Define the two operators*

$$\partial_x \mathrm{DIV}\mathbf{v} := -(D_x^x \otimes M_y)u - (D^x \otimes D_y)v, \qquad \mathrm{DIV}\mathbf{v} := (D_x \otimes M_y)u + (M_x \otimes D_y)v. \tag{50}$$

Then the following two statements are equivalent:

1.

$$D_x^x = D^x M_x^{-1} D_x \tag{51}$$

2. For all u, v such that $\mathrm{DIV}\mathbf{v} \equiv 0$, $\partial_x \mathrm{DIV}\mathbf{v} = 0$ holds.

Proof. Assume first (51) and $\mathrm{DIV}\mathbf{v} = 0$:

$$\begin{aligned}\partial_x \mathrm{DIV}\mathbf{v} &= -(D_x^x \otimes M_y)u - (D^x \otimes D_y)v \\ &= -(\underbrace{D^x M_x^{-1} D_x}_{D_x^x} \otimes M_y)u - (D^x \underbrace{M_x^{-1} M_x}_{\mathbb{1}} \otimes D_y)v \\ &= -(D^x \otimes M_y)(M_x^{-1} \otimes M_y^{-1})(D_x \otimes M_y)u - (D^x \otimes M_y)(M_x^{-1} \otimes M_y^{-1})(M_x \otimes D_y)v \\ &= -(D^x M_x^{-1} \otimes \mathbb{1}_y)\Big((D_x \otimes M_y)u + (M_x \otimes D_y)v\Big) = 0.\end{aligned}$$

Conversely,

$$\begin{aligned}0 = \partial_x \mathrm{DIV}\mathbf{v} &= -(D_x^x \otimes M_y)u - (D^x \otimes D_y)v && (52)\\ &= -(D_x^x \otimes M_y)u - (D^x \otimes D_y)v + (D^x M_x^{-1} \otimes \mathbb{1}_y)\underbrace{\Big((D_x \otimes M_y)u + (M_x \otimes D_y)v\Big)}_{=\mathrm{DIV}\mathbf{v}=0} && (53)\\ &= -(D_x^x \otimes M_y)u - (D^x \otimes D_y)v + (D^x M_x^{-1} D_x \otimes M_y)u + (D^x \otimes D_y)v && (54)\\ &= -(D_x^x \otimes M_y)u + (D^x M_x^{-1} D_x \otimes M_y)u. && (55)\end{aligned}$$

The set of (u, v) that satisfy $\mathrm{DIV}(u, v) = 0$ is very large, and e.g. u can be considered unconstrained. As (55) shall be true for all those u one concludes $D_x^x = D^x M_x^{-1} D_x$. □

Proposition 3.2. *The stabilization terms of standard SUPG do not vanish when the Galerkin approximation of the divergence vanishes, because for $\mathbb{Q}^k$ FEM (51) is never true.*

Proof. We show the proof for the Gauss-Lobatto basis functions with Gauss-Lobatto quadrature that we will use in the numerical section. In this configuration the mass matrix is diagonal and all diagonal terms are different from zero.

Now, we want to show that at least one element of $D^x M_x^{-1} D_x$ is different from the one of D_x^x. We consider the entry $(i-1,0;i,K)$ for any i and we show that this entry is 0 in D_x^x but not in the other matrix.

$$(D^x(M_x)^{-1}D_x)_{i-1,0;i,K} := \sum_{j\in\mathbb{Z},r\in[0,K-1]} \int \varphi'_{i-1,0}\varphi_{j,r}\mathrm{d}x \frac{1}{(M_x)_{j,r;j,r}} \int \varphi_{j,r}\varphi'_{i,K}\mathrm{d}x \tag{56a}$$

$$= \frac{1}{(M_x)_{i,0;i,0}} \int \varphi'_{i-1,0}\varphi_{i,0}\mathrm{d}x \int \varphi_{i,0}\varphi'_{i,K}\mathrm{d}x \tag{56b}$$

$$= \frac{1}{(M_x)_{i,0;i,0}} \underbrace{\int \varphi'_{i-1,0}\varphi_{i-1,K}dx}_{\neq 0} \underbrace{\int \varphi_{i,0}\varphi'_{i,K}dx}_{\neq 0} \neq 0, \tag{56c}$$

$$(D_x^x)_{i-1,0;i,K} := \int \varphi'_{i-1,0}\varphi'_{i,K}\mathrm{d}x = 0. \tag{56d}$$

In (56b), we have used the fact the only basis function that has support both on the support of $\varphi_{i-1,0}$ and $\varphi_{i,K-1}$ is $\varphi_{i,0} = \varphi_{i-1,K}$. Then, explicitly using the quadrature formula, we observe that $\int_{E_i^x} \varphi'_{i,s}\varphi_{i,r}\mathrm{d}x = \Delta x w_r \varphi'_{i,s}(x_{i,r})$, with $w_r = 1/\Delta x \int_{E_i^x} \varphi_{i,r}$ being the r-th quadrature weight of the Gauss–Lobatto formula. Now, $\varphi'_{i,s}(x_{i,r}) \neq 0$ for $r \neq s$ otherwise $x_{i,r}$ would have been both a zero of $\varphi_{i,s}$ and a local extremum. In this case, this zero of $\varphi_{i,s}$ would have multiplicity higher than one, but, by definition, it has multiplicity one. Hence, it must be different from 0. On the other hand, in (56d) $\varphi'_{i-1,0}$ has support only in E_{i-2}^x and E_{i-1}^x and $\varphi_{i,K}$ has support only in E_i^x and E_{i+1}^x, so the integral is zero. This concludes the proof. □

This Proposition does not allow to conclude that SUPG fails to be stationarity preserving because it might have some other non-trivial discrete stationary states, which are not governed by the Galerkin approximation DIV**v** of the divergence. However, at least for $\mathbb{Q}^1$ (Equation (46)) and $\mathbb{Q}^2$ (without proof) SUPG does not possess nontrivial discrete stationary states. The approach proposed later in the paper allows to side-step the limitations highlighted here without imposing the constraints of Theorem 3.1.

3.2 grad-div Orthogonal Subscale Stabilization (OSS)

The Orthogonal Subscale Stabilization (OSS) is a stabilization technique introduced originally for Stokes equations [17] and then extended for other problems, including convection–diffusion–reaction problems [18, 19]. For hyperbolic equations, it has been studied in [29, 20], in particular its fully discrete Fourier stability when coupled with explicit time integration methods. The OSS stabilization technique allows to use any dissipative operator by introducing a penalization term composed by the variational approximation of the dissipative operator minus the same quantity evaluated using an $L^2(\Omega)$ projection of the appropriate operator. For a Laplacian stabilization, for example, this gives a term of the form $\int_\Omega \varphi(\nabla q - w) = 0$ with w the projection of the gradient on the global approximation space.

For the acoustic system, we aim at constructing a grad-div based operator. We thus propose

to study the following stabilized variational form

$$\begin{aligned}
\int \varphi(\partial_t u + \partial_x p)dx + \int \alpha\, h\partial_x\varphi(\nabla\cdot\mathbf{u} - w^{\nabla\cdot\mathbf{u}})dx = 0\\
\int \varphi(\partial_t v + \partial_y p)dx + \int \alpha\, h\partial_y\varphi(\nabla\cdot\mathbf{u} - w^{\nabla\cdot\mathbf{u}})dx = 0\\
\int \varphi(\partial_t p + \partial_x u + \partial_y v)dx + \int \alpha\, h\partial_x\varphi(\partial_x p - w_x^p)dx + \int \alpha\, h\partial_y\varphi(\partial_y p - w_y^p)dx = 0
\end{aligned} \tag{57}$$

with the projections $w^{\nabla\cdot\mathbf{u}}$, w_x^p and w_y^p defined by

$$\begin{cases}
\int \varphi\left(\nabla\cdot\mathbf{u} - w^{\nabla\cdot\mathbf{u}}\right)dx = 0, & \forall\varphi\in V_h,\\
\int \varphi\left(\partial_x p - w_x^p\right)dx = 0, & \forall\varphi\in V_h,\\
\int \varphi\left(\partial_y p - w_y^p\right)dx = 0, & \forall\varphi\in V_h.
\end{cases} \tag{58}$$

The stability of the method can be shown classically by replacing φ by the velocities and pressure in the main system, summing up the results and removing from it the expression obtained by testing projections with $\alpha\, h(w^{\nabla\cdot\mathbf{u}},\ w_x^p,\ w_y^p)^T$.

After some algebra, one shows that the energy stability of the scheme is characterized by (neglecting boundary conditions, see also [19, 29, 20]):

$$\partial_t \int \frac{q^T q}{2}dx = -\int \alpha\, h(J^x\partial_x q + J^x\partial_y q - \tilde{w})^T(J^x\partial_x q + J^x\partial_y q - \tilde{w})dx \le 0$$

The above stabilized formulation seems a good candidate for being stationary preserving, as it involves the approximation of the proper differential terms, namely the grad-div Laplacian for the velocity equations. Unfortunately, as for SUPG, a standard discretization of the operators involved fails to be stationarity preserving. To show this, we proceed as done in the previous subsection and consider the semi-discrete version of the scheme. We first consider the projection which can be written as

$$\begin{cases}
w^{\nabla\cdot\mathbf{u}} = (M_x\otimes M_y)^{-1}((D_x\otimes M_y)u + (M_x\otimes D_y)v),\\
w_x^p = (M_x\otimes M_y)^{-1}(D_x\otimes M_y)p,\\
w_y^p = (M_x\otimes M_y)^{-1}(M_x\otimes D_y)p,
\end{cases} \tag{59}$$

Then, inserting these definitions into the stabilization terms we can show the following for the horizontal velocity

$$\begin{aligned}
s^u =& \alpha h\left[(D_x^x\otimes M_y)u + (D^x\otimes D_y)v - (D^x\otimes M_y)(M_x\otimes M_y)^{-1}((D_x\otimes M_y)u + (M_x\otimes D_y)v)\right]\\
=& \alpha h\left[(D_x^x\otimes M_y)u + (D^x\otimes D_y)v - (D^x M_x^{-1}D_x\otimes M_y)u - (D^x\otimes D_y)v\right]\\
=& \alpha h\left[(D_x^x\otimes M_y)u - (D^x M_x^{-1}D_x\otimes M_y)u\right] = ((D_x^x - D^x M_x^{-1}D_x)\otimes M_y)u,
\end{aligned} \tag{60}$$

which provides a coupled matrix representation of the stabilization term. Introducing the matrices $Z_x := D_x^x - D^x M_x^{-1}D_x$ and $Z_y := D_y^y - D^y M_y^{-1}D_y$, the OSS stabilization terms can be written in semi-discrete form as

$$s^u = \alpha h M_y\otimes Z_x u, \tag{61a}$$

$$s^v = \alpha h M_x\otimes Z_y v, \tag{61b}$$

$$s^p = \alpha h(M_y\otimes Z_x + M_x\otimes Z_y)p. \tag{61c}$$

The stabilization matrix is

$$\mathcal{E}_{\text{OSS}} := \alpha h\begin{pmatrix} Z_x\otimes M_y & 0 & 0\\ 0 & M_x\otimes Z_y & 0\\ 0 & 0 & Z_x\otimes M_y + M_x\otimes Z_y\end{pmatrix} \tag{61d}$$

and the OSS formulation can be succinctly written as

$$\mathcal{A}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q} + \mathcal{E}\mathbf{q} = 0, \qquad \text{with } \mathcal{A} = \mathcal{A}_C,\ \mathcal{E} = \mathcal{E}_C + \mathcal{E}_{\mathrm{OSS}}. \tag{62}$$

Similarly to SUPG, we observe that

$$\mathbb{F}_{t_x,t_y}(\mathcal{E}) = \begin{pmatrix} \alpha h F_{t_x}(Z_x) & 0 & \frac{t_x^2-1}{2ht_x} \\ 0 & \alpha h F_{t_y}(Z_y) & \frac{t_y^2-1}{2ht_y} \\ \frac{t_x^2-1}{2ht_x} & \frac{t_y^2-1}{2ht_y} & F_{t_x}(Z_x) + F_{t_y}(Z_y) \end{pmatrix}. \tag{63}$$

As $F_{t_x}(Z_x) = -\frac{(t_x-1)^2}{t_x h^2} + \frac{(t_x^2-1)^2}{4t_x^2 h^2}$ factors out a $(t_x-1)^2$ term and $F_{t_y}(Z_y)$ factors out a $(t_y-1)^2$ term,

$$\det \mathbb{F}_{t_x,t_y}(\mathcal{E}) = \alpha(t_x-1)^2(t_y-1)^2(\dots). \tag{64}$$

This means that the kernel is only non-trivial ($\det \mathbb{F}_{t_x,t_y}(\mathcal{E}) = 0$) when $t_x = 1$ or $t_y = 1$ (functions constant in x or y), i.e. the method is not stationarity preserving.

4 Global Flux quadrature and continuous Finite Elements

4.1 Global flux quadrature in multi-D: the GFq divergence operator

The classical Galerkin approximation of the divergence $\int \varphi(\partial_x u_h + \partial_y u_h)\mathrm{d}x\mathrm{d}y$ gives

$$\mathrm{DIV}\mathbf{v} = D_x \otimes M_y u + M_x \otimes D_y v\,. \tag{65}$$

with the discrete operators defined in (39). To generalize the 1D Global Flux idea to multiple dimensions, we use the symmetric approximation (14), introducing the new notion of divergence

$$\partial_x u + \partial_y v \equiv \partial_{xy}(U+V), \tag{66}$$

where

$$U := U_0 + \int_{y_0}^{y} u\mathrm{d}y \Longleftrightarrow \partial_y U = u, \qquad V := V_0 + \int_{x_0}^{x} v\mathrm{d}x \Longleftrightarrow \partial_x V = v. \tag{67}$$

As in one dimension, we construct discrete approximations of U_h and V_h in the same polynomial space of u_h and v_h. To achieve this, we provide a line-by-line definition of U_h and V_h whose nodal values can be constructed as

$$U = \mathbb{1}_x \otimes I_y u\ , \quad V = I_x \otimes \mathbb{1}_y v, \tag{68}$$

with integration operators defined for 1D FEM as

$$\begin{aligned} (I_x)_{i,s;k,p} &:= \int_{x_{i,0}}^{x_{i,s}} \varphi_{k,p}^x(x)\mathrm{d}x \quad \text{for } s = 1,\dots,K \quad \text{and} \\ (I_y)_{i,s;k,p} &:= \int_{y_{i,0}}^{y_{i,s}} \varphi_{k,p}^y(y)\mathrm{d}y \quad \text{for } s = 1,\dots,K. \end{aligned} \tag{69}$$

The modified weak divergence $\int \varphi\partial_{xy}(U_h + V_h)\mathrm{d}x$ then reads

$$\mathrm{DIV}\mathbf{v} = D_x \otimes D_y(U+V) = D_x \otimes D_y I_y u + D_x I_x \otimes D_y v = D_x \otimes D_y\left(\mathbb{1}_x \otimes I_y u + I_x \otimes \mathbb{1}_y v\right). \tag{70}$$

As in 1D, compared to (65) the latter formula essentially involves modifications of the mass matrices: they are replaced by the operators $D_x I_x$ and $D_y I_y$. The GFq divergence operator (70) obtained with this modification has a clear characterization of its kernel.

Proposition 4.1 (Physically relevant part of the kernel of the GFq divergence)**.** *The global flux quadrature divergence operator* (70) *vanishes identically for*

$$U_{i,k;j,s} + V_{i,k;j,s} = f(i,k) + g(j,s). \tag{71}$$

Proof. By construction $D_x \otimes D_y(f+g) = 0$ since $D_y f = 0$ and $D_x g = 0$ which immediately yields the result. □

A neat way of writing the above property is obtained by using the local assembly of (70), which reads using a FEM notation

$$[\mathrm{DIV}\mathbf{v}]_{\alpha;\beta} = \sum_{E \ni (\alpha;\beta)} [D_x^E \otimes D_y^E (\mathbb{1}_x^E \otimes I_y^E \, u^E)]_{\alpha;\beta} + \sum_{E \ni (\alpha;\beta)} [D_x^E \otimes D_y^E (I_x^E \otimes \mathbb{1}_y^E \, v^E)]_{\alpha;\beta} \, .$$

where the superscript ^{E} denotes the local entries of operators and arrays inside the element E. On the element $E = E_{ij}$, consider now the local arrays

$$[u_0^E]_{i,s;j,p} := u_{i,0;j,p} \; \forall s = 0, \ldots, K \, , \quad [v_0^E]_{i,s;j,p} := v_{i,s;j,0} \; \forall p = 0, \ldots, K \, .$$

Define now the elemental array of integrated divergences on each element $E = E_{ij}$

$$\begin{aligned} \Phi^E :=& (\mathbb{1}_x^E \otimes I_y^E)(u^E - u_0^E) + (I_x^E \otimes \mathbb{1}_y^E)(v^E - v_0^E), \\ \Phi^E_{i,s;j,p} =& \int_{y_{j,0}}^{y_{j,p}} (u_h(x_{i,s}, y) - u_h(x_{i,0}, y)) \mathrm{d}y + \int_{x_{i,0}}^{x_{i,s}} (v_h(x, y_{j,p}) - v_h(x, y_{j,0})) \mathrm{d}x. \end{aligned} \tag{72}$$

Using the fact that $D_x^E \otimes D_y^E u_0^E = D_x^E \otimes D_y^E v_0^E = 0$, we can readily see that

$$[\mathrm{DIV}\mathbf{v}]_{\alpha;\beta} = \sum_{E \ni (\alpha;\beta)} [(D_x^E \otimes D_y^E)\Phi^E]_{\alpha;\beta}. \tag{73}$$

Proposition 4.2 (GFq divergence and vanishing subcell integrals)**.** *The global flux quadrature divergence operator* (70) *vanishes identically whenever* $\forall E_{ij}$ *and* $\forall \, s, p \in E_{ij}$ *the integrated divergence on the subcell* $[x_{i,0}, x_{i,s}] \times [y_{j,0}, y_{j,p}]$ *vanishes:*

$$\Phi^E_{i,s;j,p} = 0 \; \forall s, p \text{ and } \forall E_{i,j} \Rightarrow \mathrm{DIV}\mathbf{v} = 0.$$

The proof follows directly the previous computations.
The last proposition shows two important properties:

1. *the approach introduced allows to define, at steady state, as many linearly independent zero divergences as the number of nodes in the mesh;*

2. *the GFq approach allows to naturally pass from nodal to face integrated quantities.* Indeed, U and V contain integrated values of the velocities in the directions normal to the faces of the element sub-cells. These are natural objects to express the integrals

$$\iint (\partial_x u + \partial_y v) \mathrm{d}x\mathrm{d}y = \int [u]_x \mathrm{d}y + \int [v]_y \mathrm{d}x = \left[\int u \mathrm{d}y \right]_x + \left[\int v \mathrm{d}x \right]_y . \tag{74}$$

This establishes a loose link to mimetic schemes using face averages of normal components.

4.2 Construction of the integrators in multi-D

Finally, we give a general recipe how to construct integrators I_x, I_y that allow to discretize

$$\partial_y U = u, \qquad \partial_x V = v \tag{75}$$

in such a way that all the discrete spatial derivatives are compact differences, once the method is expressed in terms of u and v.

To ensure a local nature of the method, a condition on I_x, I_y is that $\mathcal{E}$, and in particular $D_x I_x, D_x^x I_x$, etc. have to be compact difference operators. For $\mathbb{Q}^1$ FEM this is ensured by choosing

$$I_x = \Delta x \frac{t_x + 1}{2(t_x - 1)} \tag{76}$$

because the characteristic polynomial of any finite difference formula that discretizes a derivative can always be divided by $t_x - 1$ (see [30]). Observe the identities

$$D_x I_x = \frac{t_x^2 + 2t_x + 1}{4t_x}, \qquad D_x^x I_x = -D_x, \tag{77}$$

which follow from

$$\mathbb{F}_{t_x}(D_x I_x) = \frac{(t_x+1)^2}{4t_x}, \qquad \mathbb{F}_{t_x}(D_x^x I_x) = -\frac{(t_x-1)(t_x+1)}{2t_x \Delta x} = -\mathbb{F}_{t_x}(D_x). \tag{78}$$

The idea in the context of FEM is to take $u, v \in V_h^K$ and integrate them in y and x, respectively, inside each cell E_{ij}. This would give piecewise $U(x,y) = \int^y u(x,s)\mathrm{d}s \in \mathbb{Q}^K(E_i^x) \times \mathbb{Q}^{K+1}(E_j^y)$ and $V(x,y) = \int^x v(s,y)\mathrm{d}s \in \mathbb{Q}^{K+1}(E_i^x) \times \mathbb{Q}^K(E_j^y)$. We then choose to project U, V pointwise back onto $\mathbb{Q}^K(E_{ij})$. In particular, we write

$$u(x,y) = \sum_{(i,j)\in\mathbb{Z}^2} \sum_{z,w=1}^{K} u_{i,z;j,w} \varphi_{i,z}^x(x) \varphi_{j,w}^y(y) = \sum_{(i,j)\in\mathbb{Z}^2} \sum_{z,w=0}^{K} u_{i,z;j,w} \varphi_{i,z}(x)\Big|_{E_i^x} \varphi_{j,w}(y)\Big|_{E_j^y}, \tag{79a}$$

$$v(x,y) = \sum_{(i,j)\in\mathbb{Z}^2} \sum_{z,w=1}^{K} v_{i,z;j,w} \varphi_{i,z}^x(x) \varphi_{j,w}^y(y) = \sum_{(i,j)\in\mathbb{Z}^2} \sum_{z,w=0}^{K} v_{i,z;j,w} \varphi_{i,z}^x(x)\Big|_{E_i^x} \varphi_{j,w}^y(y)\Big|_{E_j^y}, \tag{79b}$$

see Figure 1. Integrating u (or v) with respect to y (or x), we get in the cell $E_{i,j}$:

$$\begin{aligned} \hat{U}(x,y) &:= \int_{y_0}^{y} u(x,y')\mathrm{d}y' = \hat{U}(x,y_j) + \int_{y_j}^{y} u(x,y')\mathrm{d}y' = \\ &\hat{U}(x,y_j) + \sum_{z',w'=0}^{K} u_{i,z';j,w'} \int_{y_j}^{y} \varphi_{i,z'}^x(x)\varphi_{j,w'}^y(y')dy', \end{aligned} \tag{80a}$$

$$\begin{aligned} \hat{V}(x,y) &:= \int_{x_0}^{x} v(x',y)\mathrm{d}x' = \hat{V}(x_i,y) + \int_{x_i}^{x} v(x',y)\mathrm{d}x' = \\ &\hat{V}(x_i,y) + \sum_{z',w'=0}^{K} v_{i,z';j,w'} \int_{x_i}^{x} \varphi_{i,z'}^x(x')\varphi_{j,w'}^y(y)dx'. \end{aligned} \tag{80b}$$

We will show below that the step-functions $\hat{U}(x,y_j), \hat{V}(x_i,y)$ are of no importance for the final form of the method, which will allow us to eventually drop them. We also want to highlight that the restriction of u on cell E_{ij}

$$U\Big|_{E_{ij}} = \sum_{z',w'=0}^{K} u_{i,z';j,w'} \int_{y_j}^{y} \varphi_{i,z'}^x(x)\varphi_{j,w'}^y(y')dy' \tag{81}$$

includes the degrees of freedom associated to $z' = 0$ and $w' = 0$, which belong also to the previous cell. Recall their definitions:

$$u_{i,0;j,w'} := u_{i-1,K;j,w'}, \qquad \forall w' = 1, \dots, K, \tag{82a}$$

$$u_{i,z';j,0} := u_{i,z';j-1,K}, \qquad \forall w' = 1, \dots, K, \tag{82b}$$

$$u_{i,0;j,0} := u_{i-1,K;j-1,K}. \tag{82c}$$

Proposition 4.3 (Differentiation of integrals is independent on the starting value)**.** *Consider $U(x,y) := \sum_{z,w} \varphi^x_{i,z}(x)\varphi^y_{j,w}(y)U_{i,z;j}$ in the cell E_{ij} with some $U_{i,z;j}$ only depending on the cell j, not on w, the DoF in y. Then, $\partial_y U(x,y) = 0$ in the cell E_{ij}.*

Proof. Let us compute $\partial_y U(x,y)$ in the cell E_{ij}. To this end we need to include the degrees of freedom associated to $z, w = 0$:

$$\partial_y U(x,y)\Big|_{E_{ij}} = \partial_y \left(\sum_{z,w=0}^{K} \varphi^x_{i,z}(x)\varphi^y_{j,w}(y)U_{i,z;j} \right) = \sum_{z=0}^{K} \varphi^x_{i,z}(x)U_{i,z;j}\partial_y \underbrace{\left(\sum_{w=0}^{K} \varphi^y_{j,w}(y) \right)}_{\equiv 1} = 0. \tag{83}$$

□

It this does not matter which constant we choose to define U in each cell. As in the construction of the method only $\partial_y U$ appears, the term $\hat{U}(x, y_j)$ in (80a) can be dropped straight away. We thus define

$$U_{i,z;j,w} := \sum_{w'=0}^{K} u_{i,z;j,w'} \int_{y_j}^{y_{j,w}} \varphi_{j,w'}(y')dy', \tag{84a}$$

$$V_{i,z;j,w} := \sum_{z'=0}^{K} v_{i,z';j,w} \int_{x_i}^{x_{i,z}} \varphi_{i,z'}(x')dx', \qquad z, w = 1, \dots, K, \tag{84b}$$

and pass from U and V to u and v with the matrix multiplications

$$U = \mathbb{1}_x \otimes I_y u \qquad\qquad V = I_x \otimes \mathbb{1}_y v \tag{85}$$

with the integrator I_x defined by

$$(I_x \otimes \mathbb{1}_y v)_{i,z;j,w} = \sum_{z'=0}^{K} v_{i,z';j,w} \int_{x_i}^{x_{i,z}} \varphi_{i,z'}(x')dx'. \tag{86}$$

Notice that doing so, we are implicitly defining $U(x,y) := \sum_{\alpha,\beta} \varphi_{\alpha;\beta}(x,y)U_{\alpha;\beta} \in V_h^K$, while formally $\hat{U}$ in (80a) was belonging to a different functional space with higher degree of polynomials in y direction. This step is a projection onto the space V_h. The two polynomials, nevertheless, coincide on the degrees of freedom, i.e.,

$$\hat{U}(x_\alpha, y_\beta) = U_{\alpha;\beta} = U(x_\alpha; y_\beta). \tag{87}$$

Example 4.1. *In the case of $\mathbb{Q}^1$ FEM ($K = 1$) one finds*

$$(I_x v)_{i,1;j,w} = v_{i,0;j,w} \int_{x_{i,0}}^{x_{i+1,0}} \varphi_{i,0}(x')dx' + v_{i,1;j,w} \int_{x_{i,0}}^{x_{i,1}} \varphi_{i,1}(x')dx' = \Delta x \frac{v_{i,0;j,w} + v_{i+1,0;j,w}}{2} \tag{88a}$$

i.e.

$$\mathbb{F}_{t_x}(I_x) = \Delta x \frac{t_x + 1}{2t_x} = \Delta x \left(\frac{t_x + 1}{2(t_x - 1)} - \frac{t_x + 1}{2t_x(t_x - 1)} \right), \tag{88b}$$

which is similar to, but not exactly, the factor $\frac{t_x+1}{2(t_x-1)}$ used in Equation (76).

4.3 Nodal projection, nodal consistency, and super-convergence

We have now a new definition of the discrete divergence, and we have shown that the desired physical equilibria are part of its kernel. We now set out to study the following two questions:

- Given an element of the space of discrete divergence-free solutions of Propositions 4.1 and 4.2, what is its formal consistency with respect to exact analytical solutions?
- Given a divergence free vector field, how to devise a projection onto the space of discrete divergence free solutions?

The first question is covered by the following

Proposition 4.4 (GFq divergence: consistency estimate)**.** *Consider a $C^P(\Omega)$ solenoidal vector field (u_e, v_e) with $P \geq 1$, such that the solenoidal condition $\partial_x u_e(x,y) = -\partial_y v_e(x,y)$ is true in every point and in particular at all collocation points. Given an ODE $U'(t) = F(U,t)$, let the integrators*

$$U_p - U_0 = (I_x F)_p \ , \quad U_p - U_0 = (I_y F)_p \ ,$$

be exact when F is a polynomial of degree M. Then, for $P \geq M$, the global flux divergence (70) *admits exact discrete kernels verifying Propositions 4.1 or 4.2, and such that $u = u_e$ and $v = v_e$ on $\partial\Omega_h$ which also verify the consistency estimates $u = u_e + \mathcal{O}(h^M)$, and $v = v_e + \mathcal{O}(h^M)$.*

Proof. Since $\partial_x u_e + \partial_y v_e = 0$ is true pointwise and in particular at all collocation points (x_α, y_β), we can remove from the expression of DIV operators applied to pointwise values of the derivatives $\partial_x u_e(x_\alpha, y_\beta)$ and $\partial_y v_e(x_\alpha, y_\beta)$. In particular note that

$$\partial_x u_e(x_\alpha, y_\beta) + \partial_y v_e(x_\alpha, y_\beta) = 0 \Rightarrow (\partial_x u_e)_h + (\partial_y v_e)_h = 0$$

We thus start from (70) and add or remove interpolated values of the above zero divergence tested against all $\varphi^x \varphi^y \in V_h$

$$\begin{aligned} \mathrm{DIV}\mathbf{v} =& (D_x \otimes D_y I_y)(u_h + (I_x \otimes \mathbb{1}_y)((\partial_x u_e)_h + (\partial_y v_e)_h)) + \\ & (D_x I_x \otimes D_y)(v_h + (\mathbb{1}_x \otimes I_y)((\partial_x u_e)_h + (\partial_y v_e)_h)). \end{aligned} \tag{89}$$

Simple manipulations show that the previous expression is equivalent to

$$\begin{aligned} \mathrm{DIV}\mathbf{v} =& (D_x \otimes D_y I_y)(u_h + (I_x \otimes \mathbb{1}_y)(\partial_y v_e)_h) + \\ & (D_x I_x \otimes D_y)(v_h + (\mathbb{1}_x \otimes I_y)(\partial_x u_e)_h) + (D_x I_x \otimes D_y I_y)(\cancelto{0}{(\partial_x u_e)_h + (\partial_y v_e)_h}). \end{aligned} \tag{90}$$

Since $u = u_e$ and $v = v_e$ on $\partial\Omega$, we consider discrete states defined by marching along gridlines using the ODE integrator defined by I_x and I_y by integrating for every x_α and every y_β the ODEs implicitly appearing in (90), i.e.,

$$\frac{du(x, y_\beta)}{dx} = -\partial_y v_e(x, y_\beta) \ , \quad \frac{dv(x_\alpha, y)}{dy} = -\partial_x u_e(x_\alpha, y).$$

By construction the resulting nodal values $u(x_\alpha, y_\beta)$ and $v(x_\alpha, y_\beta)$ verify

$$u_h + (I_x \otimes \mathbb{1}_y)(\partial_y v_e)_h = \mathrm{c}_x(y) \ , \quad v_h + (\mathbb{1}_x \otimes I_y)(\partial_x u_e)_h = \mathrm{c}_y(x) \tag{91}$$

and by virtue of (90) are thus exact discrete solutions of $\mathrm{DIV}\mathbf{v} = 0$. Moreover, if integration is started from the boundary points (x_0, y_β) and (x_α, y_0), the hypotheses on the exactness of integration tables and on the regularity of (u_e, v_e) lead to the local nodal consistency estimates

$$\begin{aligned} u(x_\alpha, y_\beta) &= u_e(x_0, y_\beta) - \int_{x_0}^{x_s} \partial_y v_e(x, y_\beta) dx + \mathcal{O}(h^{M+1}) = u_e(x_\alpha, y_\beta) + \mathcal{O}(h^{M+1}) \ , \\ v(x_\alpha, y_\beta) &= v_e(x_\alpha, y_0) - \int_{y_0}^{y_\beta} \partial_x u_e(x_\alpha, y) dy + \mathcal{O}(h^{M+1}) = v_e(x_\alpha, y_\beta) + \mathcal{O}(h^{M+1}). \end{aligned}$$

The global consistency is obtained classically by considering the space marching on the whole domain, on a number of cells of order h^{-1} which leads to the sough h^M estimate. □

Concerning the initialization, the proof of Theorem 4.2, and in particular (72), provides an idea how to construct discrete projections on the kernel of (70). In particular, we define hereafter the following quadrature-based projection.

Definition 4.1 (Line-by-line quadrature projection)**.** *Let (u_e, v_e) be a smooth enough vector field. Let $u(0, y_\beta) = u_e(0, y_\beta)$ and $v(x_\alpha, 0) = v_e(x_\alpha, 0)$ on the bottom and left of the domain $x = 0$ and $y = 0$. Given these values, we define recursively over line/row elements the* $\mathbf{v}$ *fulfilling*

$$[I_y^{E_j^y} u^{E_j^y}(x_{i,s})]_p := \int_{y_{j,0}}^{y_{j,p}} u_e(x_{i,s}, y)dy\ , \quad [I_x^{E_i^x} v^{E_i^x}(y_{j,p})]_s := \int_{x_{i,0}}^{x_{i,s}} v_e(x, y_{j,p})dx \tag{92}$$

with $I_x^{E_i^x}$ and $I_y^{E_j^y}$ the local restriction of the integration tables, and with local initial conditions on each element $u_h(x_s, y_{j,0}) = u_h(x_s, y_{j-1,K})$ and $v_h(x_{i,0}, y_p) = v_h(x_{i-1,K}, y_p)$.

We can immediately prove the following

Proposition 4.5 (Line-by-line quadrature projection of solenoidal data)**.** *Let (u_e, v_e) be a given smooth enough solenoidal field, if the quadrature of the components of (u_e, v_e) in* (92) *is of order M_q then the line by line/row by row quadrature projection is equivalent to* (91) *within $\max(h^{M_q}, h^M)$, and it is in the kernel of* (70) *for exact integration of the right hand sides in* (92)*. Moreover, the projected data has a pointwise consistency w.r.t. (u_e, v_e) of order $\max(h^{M_q}, h^M)$.*

Proof. We prove the result for the u component, the proof for the v component is similar. We can write that

$$\begin{aligned}
\int_{y_{j,0}}^{y_{j,p}} u_h(x_{i,s}, y)dy &= \int_{y_{j,0}}^{y_{j,p}} u_e(x_{i,s}, y)dy + \mathcal{O}(h^{M_q+1}) \\
&= \int_{y_{j,0}}^{y_{j,p}} u_e(x_{i,0}, y)dy - \int_{y_{j,0}}^{y_{j,p}} \int_{x_{i,0}}^{x_{i,s}} \partial_y u_e(x, y)dx\, dy + \mathcal{O}(h^{M_q+1}) \\
&= \int_{y_{j,0}}^{y_{j,p}} u_e(x_{i,0}, y)dy - \int_{y_{j,0}}^{y_{j,p}} [I_x^{E_i^x}(\partial_y u_e)_h^{E_i^x}(y)]_s\, dy + \mathcal{O}(h^{M_q+1}) + \mathcal{O}(h^{M+1})
\end{aligned}$$

having used the hypotheses of the local initial conditions. The latter is equivalent to

$$u_h(x_{i,s}, y_{j,p}) + (I_x \otimes \mathbb{1}_y)(\partial_y v_e)_h = u_h(x_{i,0}, y_{j,p}) + \mathcal{O}(h^{M_q+1}) + \mathcal{O}(h^{M+1})$$

which shows that the equivalence with (92) within $\max(h^{M_q}, h^M)$. Using the solenoidal condition on (u_e, v_e), one can also check now that by definition

$$\begin{aligned}
[\Phi^E]_{i,s;j,p} &= \int_{y_0}^{y_p} (u_h(x_s, y) - u_h(x_0, y))dy + \int_{x_0}^{x_s} (v_h(x, y_p) - v_h(x, y_0))dx \\
&= \int_{y_0}^{y_p} (u_e(x_s, y) - u_e(x_0, y))dy + \int_{x_0}^{x_s} (v_e(x, y_p) - v_e(x, y_0))dx + \mathcal{O}(h^{M_q+1}) = \mathcal{O}(h^{M_q+1})
\end{aligned}$$

From Proposition 4.2 for exact integration the projected data is in the kernel of (70). □

The above directional initialization introduces some apparent dependence on the initial integration point and direction of marching. However, the scheme is in reality symmetric with respect to the above choices. This can be seen from the following property.

Proposition 4.6 (Reversibility of global flux quadrature SEM)**.** *The projection operator $D_x I_x$ obtained with definitions* (69) *is independent on the orientation of integration.*

Proof. The reversed table $\tilde{I}_x$ verifies

$$(I_x)_{i,j;i,l} = \int_{x_{i,0}}^{x_{i,l}} \varphi_{i,l}(x)dx = \int_{x_{i,0}}^{x_{i,K}} \varphi_{i,l}(x)dx + \int_{x_{i,K}}^{x_{i,j}} \varphi_{i,l}(x)dx = \int_{x_{i,0}}^{x_{i,K}} \varphi_l(x)dx + (\tilde{I}_x)_{i,j;i,l},$$

with $(\tilde{I}_x)_{i,j;i,l} := -\int_{x_{i,j}}^{x_{i,K}} \varphi_{i,l}(x)dx$. This implies $I_x S = \overline{S} + \tilde{I}_x S$, with $\overline{S}$ containing constant entries given by the average of the source. As a consequence $D_x I_x S = D_x \tilde{I}_x S$. □

The above property shows that locally both directions of integration provide the same global flux quadrature scheme at the end. In practice, it is the boundary conditions that will define the actual steady solution of the scheme. The line by line/row by row quadrature projection is in practice simple to implement but unless corrected with several sweeps in different vertical/horizontal directions, or of some combinations of the projection in different directions, it is affected by the accumulation of the error along the mesh, as any ODE integrator. To avoid this drawback we have considered the direct optimization based projection defined below.

Definition 4.2 (Optimization based projection)**.** *Consider initial data obtained by nodally sampling a given solenoidal vector field: $(u_e(x_s, y_p), v_e(x_s, y_p))$. The optimization based projection consists in looking for perturbed nodal data $(\tilde{u}(x_s, y_p), \tilde{v}(x_s, y_p))$ whose error w.r.t. the initial sample data is minimized, under the constraint that the discrete divergence should vanish:*

$$(\tilde{u}, \tilde{v}) = \underset{u,v \in V_h \,:\, \mathrm{DIV}\mathbf{v}=0}{\arg\min} ||u - u_e||_2^2 + ||v - v_e||_2^2. \tag{93}$$

The above definition requires solving a linearly constraint optimization problem with a very simple quadratic functional to be minimized. This solution can be obtained e.g. using a Trust-Region Constrained Algorithm[1]. For the above method we cannot prove any consistency estimate, but in practice we obtain data with the nodal consistency of Theorem 4.4 and Proposition 4.5 within 2 iterations of the algorithm.

When initializing the solution with a given solenoidal vector field, we thus have three possibilities: sampling at collocation points; line by line/row by row quadrature projection; optimization based projection. These will be evaluated and compared in detail in the results section.

5 GFq based grad-div compatible stabilization

A high-order approximation of the divergence is not enough, because the system (1) under consideration is hyperbolic and stabilization is required. It was shown in Section 1.2 that appropriate stabilization must be used in order to preserve the stationary states. We integrate in this section the global Flux technique into the SUPG and OSS stabilizations.

5.1 SUPG stabilization with GFq

We construct the GFq variant of the SUPG method by evaluating the integrals

$$\begin{aligned}
\int \varphi_h(\partial_t u_h + \partial_x p_h)dx + \int \alpha\, h \partial_x \varphi(\partial_t p_h + \partial_{xy}(U_h + V_h))dx = 0 \\
\int \varphi_h(\partial_t v_h + \partial_y p_h)dx + \int \alpha\, h \partial_y \varphi(\partial_t p_h + \partial_{xy}(U_h + V_h))dx = 0 \\
\int \varphi_h(\partial_t p_h + \partial_{xy}(U_h + V_h)dx + \int \alpha\, h \partial_x \varphi(\partial_t u_h + \partial_x p_h)dx + \int \alpha\, h \partial_y \varphi_h(\partial_t v_h + \partial_y p_h)dx = 0
\end{aligned} \tag{94}$$

[1] an open source implementation can be found in `scipy.optimize`

We then use the definitions (84) of the nodal values of U_h and V_h and end up with the modified version of the SUPG scheme:

$$(\mathcal{A}_C + \mathcal{A}_{\text{SU}})\frac{d\mathbf{q}}{dt} + \mathcal{E}_{\text{C-GFq}}\mathbf{q} + \mathcal{E}_{\text{SU-GFq}}\mathbf{q} = 0 \tag{95}$$

with $\mathcal{A}_C$ and $\mathcal{A}_{\text{SU}}$ the same as in Section 3.1 and $\mathcal{E}_{\text{SUPG-GFq}} = \mathcal{E}_{\text{C-GFq}} + \mathcal{E}_{\text{SU-GFq}}$ with

$$\mathcal{E}_{\text{C-GFq}} = \begin{pmatrix} 0 & 0 & D_x \otimes M_y \\ 0 & 0 & M_x \otimes D_y \\ D_x \otimes (D_y I_y) & (D_x I_x) \otimes D_y & 0 \end{pmatrix} \tag{96a}$$

and

$$\mathcal{E}_{\text{SU-GFq}} = \alpha\, h \begin{pmatrix} D_x^x \otimes (D_y I_y) & (D_x^x I_x) \otimes D_y & 0 \\ D_x \otimes (D_y^y I_y) & (D_x I_x) \otimes D_y^y & 0 \\ 0 & 0 & D_x^x \otimes M_y + M_x \otimes D_y^y \end{pmatrix} \tag{96b}$$

The form of the above operators allows to prove the following result.

Proposition 5.1 (Equilibria of SUPG-GFq)**.** *Any state with constant pressure p and velocities in the kernel of the global flux divergence operator, as characterized by Proposition 4.1 or equivalently Proposition 4.2, is a steady equilibrium of the GFq based SUPG.*

Proof. If (71) is true, then both the divergence and the stabilization terms in the u and v equations vanish:

$$D_x \otimes (D_y I_y)u + (D_x I_x) \otimes D_y v = (D_x \otimes D_y)(\mathbb{1}_x \otimes I_y u + I_x \otimes \mathbb{1}_y v) = (D_x \otimes D_y)(U + V) = 0$$

and also

$$\begin{aligned} \alpha\, h\Big(D_x^x \otimes (D_y I_y)u + (D_x^x I_x) \otimes D_y v\Big) &= \alpha\, h(D_x^x \otimes D_y)\Big(\mathbb{1}_x \otimes I_y u + I_x \otimes \mathbb{1}_y v\Big) \\ &= \alpha\, h D_x^x \otimes D_y(U + V) = 0 \end{aligned}$$

and similarly for the v equation. □

Using the characteristic polynomial / Fourier representation of the scheme, as done in Section 3.1, one immediately confirms for the case of $\mathbb{Q}^1$ FEM that, independently on further choices

$$\det \mathbb{F}_{t_x,t_y}(\mathcal{E}) = 0. \tag{97}$$

and that the right kernel of $\mathbb{F}_{t_x,t_y}(\mathcal{E})$ (related stationary states) is parallel to

$$(-\mathbb{F}_{t_x}(I_x), \mathbb{F}_{t_y}(I_y), 0)^{\mathrm{T}}. \tag{98}$$

The above property shows one of the key differences with respect to the scheme of Section 3.1: the kernel of the stabilization includes that of the divergence, in particular it contains equilibria which have physical meaning, and are characterized by Theorem 4.1 and Proposition 4.2. The kernel of the divergence, however, may also contain non-physical modes. This aspect is discussed in more detail in Section 5.3.

5.2 GFq stabilization operators: OSS

Proceeding in a similar manner we construct a GFq version of the OSS stabilization, whose discrete equations are obtained from, $\forall \varphi_h \in V_h^K$,

$$\begin{aligned}
\int \varphi_h(\partial_t u_h + \partial_x p_h)dx + \int \alpha\, h\partial_x\varphi_h(\partial_{xy}(U_h+V_h) - w_h^{\nabla\cdot\mathbf{u}})dx = 0,\\
\int \varphi_h(\partial_t v_h + \partial_y p_h)dx + \int \alpha\, h\partial_y\varphi_h(\partial_{xy}(U_h+V_h) - w_h^{\nabla\cdot\mathbf{u}})dx = 0,\\
\int \varphi_h(\partial_t p_h + \partial_{xy}(U_h+V_h))dx + \int \alpha\, h\partial_x\varphi_h(\partial_x p_h - w_h^{p_x})dx + \int \alpha\, h\partial_y\varphi_h(\partial_y p - w^{p_y})dx = 0,
\end{aligned} \tag{99}$$

with the projections $w_h^{\nabla\cdot\mathbf{u}}$, w_x^p and w_y^p defined by

$$\begin{cases}
\int \varphi_h \left(\partial_{xy}(U_h+V_h) - w_h^{\nabla\cdot\mathbf{u}}\right) dx = 0,\\
\int \varphi_h \left(\partial_x p_h - w_h^{p_x}\right) dx = 0,\\
\int \varphi_h \left(\partial_y p_h - w_h^{p_y}\right) dx = 0,
\end{cases} \tag{100}$$

With the notation of Sections 3.2 and 5.1, we can show that the evaluation of the above integrals leads to the following stabilization terms for the velocity equations (compare with the the OSS ones in (61))

$$\begin{aligned}
s^u =& \alpha h \left\{(Z_x \otimes D_y I_y)u + (Z_x I_x \otimes D_y)v\right\},\\
s^v =& \alpha h \left\{(D_x \otimes Z_y I_y)u + (D_x I_x \otimes Z_y)v\right\}.
\end{aligned} \tag{101}$$

The pressure stabilization is identical to the standard case. This leads to the final GFq form of the OSS stabilized scheme:

$$\mathcal{A}_C \frac{d\mathbf{q}}{dt} + \mathcal{E}_{\text{C-GFq}}\mathbf{q} + \mathcal{E}_{\text{OSS-GFq}}\mathbf{q} = 0 \tag{102a}$$

where

$$\mathcal{E}_{\text{OSS-GFq}} := \alpha h \begin{pmatrix} Z_x \otimes D_y I_y & Z_x I_x \otimes D_y & 0 \\ D_x \otimes Z_y I_y & D_x I_x \otimes Z_y & 0 \\ 0 & 0 & Z_x \otimes M_y + M_x \otimes Z_y \end{pmatrix}. \tag{102b}$$

As for SUPG with GFq, we can readily characterize the steady states of this method.

Proposition 5.2 (Known equilibria of OSS-GFq)**.** *Any state with constant pressure p and velocities in the kernel of the global flux divergence operator, as characterized by Theorem 4.1 or equivalently Proposition 4.2, is a steady equilibrium of the GFq based OSS method.*

Proof. The only thing we need to check is that the stabilization terms (101) vanish. This is a trivial consequence of the hypotheses made, which imply that not only all the terms in the pressure stabilization vanish, but also $(D_x \otimes D_y)(U_h + V_h)$ and $w_h^{\nabla\cdot\mathbf{u}}$. □

For $\mathbb{Q}^1$, the right kernel is the same as for the SUPG (98).

Again, the above property is a major difference with respect to the scheme of Section 3.2: the stabilization shares with the non-stabilized scheme its kernel of equilibria, characterized by Theorem 4.1 and Proposition 4.2. As already remarked for the SUPG, these may not be just the physically relevant ones, but the kernel might contain non-physical modes. This aspect is discussed in more detail in Section 5.3.

5.3 Kernel of derivative operators

Elements of the kernel of the GFq divergence operator are also contained in the kernel of the full discretization schemes proposed, including the stabilization, and that among them one can identify representatives of continuous stationary states. However, in general, other steady states, i.e. spurious modes, may exist in the kernel of the divergence operator. In this case, it is essential that such unphysical modes are not in the kernel of the stabilization, so that they will be dissipated, if present.

One of the interesting features of the tensor based GFq method is that everything boils down to studying the $(U + V)$ variable instead of the two variables u and v independently, and information on one-dimensional kernels can be very easily applied to the multi-dimensional case. This section is devoted to the study of the impact of the SUPG and OSS stabilization on spurious modes contained in the kernel of the discrete divergence.

5.3.1 One dimensional kernels

Before dealing with two dimensions consider one-dimensional operators. To do so, let us introduce another simplified notation for the following FEM spaces

$$V^K_{\Delta x}(\Omega^x_{\Delta x}) = \left\{q \in \mathcal{C}^0(\Omega^x_{\Delta x}) : q|_E \in \mathbb{P}^K(E),\, \forall E \in \Omega^x_{\Delta x}\right\}, \tag{103a}$$

$$V^K_{\Delta x,0}(\Omega^x_{\Delta x}) = \left\{q \in \mathcal{C}^0(\Omega^x_{\Delta x}) : q|_E \in \mathbb{P}^K(E),\, \forall E \in \Omega^x_{\Delta x} \text{ and } q(x) = 0\, \forall x \in \partial\Omega^x_{\Delta x}\right\}, \tag{103b}$$

$$V^{K-1}_{\Delta x,b}(\Omega^x_{\Delta x}) = \left\{q \in L^2(\Omega^x_{\Delta x}) : q|_E \in \mathbb{P}^{K-1}(E),\, \forall E \in \Omega^x_{\Delta x}\right\}. \tag{103c}$$

In order to study the derivative operators, we do not impose any boundary conditions that could introduce further constraints. We are looking for the kernel of the following operators defined on $V^K_{\Delta x}(\Omega^x_{\Delta x}) \ni u_h$

$$D_x u := \int_{\Omega^x_{\Delta x}} \varphi^x(x)\partial_x u_h(x), \qquad \forall \varphi^x \in V^K_0(\Omega^x_{\Delta x}), \tag{104}$$

$$D^x_x u := \int_{\Omega^x_{\Delta x}} \partial_x\varphi^x(x)\partial_x u_h(x), \qquad \forall \varphi^x \in V^K_0(\Omega^x_{\Delta x}). \tag{105}$$

We observe that $V^K_{\Delta x}(\Omega^x_{\Delta x}) \sim \mathbb{R}^{N_x\times K+1}$, $V^K_{\Delta x,0}(\Omega^x_{\Delta x}) \sim \mathbb{R}^{N_x\times K-1}$, so the linear operators can be equivalently seen as $D_x, D^x_x : \mathbb{R}^{N_x\times K+1} \to \mathbb{R}^{N_x\times K-1}$. Then it is clear that there is a non-empty kernel of dimension at least 2 for these operators. The trivial constant state is of course part of both kernels. This gives us a hint to study a simplified form of these operators. Observe that $z_h := \partial_x u_h \in V^{K-1}_{\Delta x,b}(\Omega^x_{\Delta x})$. Clearly, all the constant states $u_h \equiv u_0 \in \mathbb{R}$ vanish upon differentiation, as $\partial_x : V^K_h(\Omega^x_{\Delta x}) \sim \mathbb{R}^{N_x\times K+1} \to V^{K-1}_{\Delta x,b}(\Omega^x_{\Delta x}) \sim \mathbb{R}^{N_x\times K}$ has a one-dimensional kernel generated by $u_h \equiv 1$. This can be easily seen looking at each cell for functions that are $\partial_x u_h = 0$. Indeed, being 0 polynomials in each cell, it must be that u_h is constant in every cell, and, being continuous, it must be a constant over the whole domain.

Consider therefore new operators $\tilde{D}_x, \tilde{D}^x_x : V^{K-1}_{\Delta x,b}(\Omega^x_{\Delta x}) \sim \mathbb{R}^{N_x\times K} \to V^K_{\Delta x,0}(\Omega^x_{\Delta x}) \sim \mathbb{R}^{N_x\times K-1}$ defined as

$$\tilde{D}_x z := \int_{\Omega^x_{\Delta x}} \varphi(x) z_h(x), \qquad \forall \varphi \in V^K_{\Delta x,0}(\Omega^x_{\Delta x}), \tag{106}$$

$$\tilde{D}^x_x z := \int_{\Omega^x_{\Delta x}} \partial_x\varphi(x) z_h(x), \qquad \forall \varphi \in V^K_{\Delta x,0}(\Omega^x_{\Delta x}). \tag{107}$$

Proposition 5.3 (Kernel characterization)**.** *$\tilde{D}_x, \tilde{D}^x_x : \mathbb{R}^{N_x\times K} \to \mathbb{R}^{N_x\times K-1}$ have kernels of dimension one. The kernel of $\tilde{D}_x$ is generated by a function that is discontinuous at each cell interface, while the kernel of $\tilde{D}^x_x$ is generated by the constant function* 1.

The proof can be found in Appendix D.

Corollary 5.1 (Kernel characterization of D_x^x, Z_x and D_x)**.** *Consider $D_x, D_x^x, Z_x : \mathbb{R}^{N_x \times K+1} \to \mathbb{R}^{N_x \times K-1}$ defined with test functions in $V_{\Delta x,0}^K$ and trial functions in $V_{\Delta x}^K$. The kernel of D_x^x is $\langle 1, x \rangle$, the kernel of Z_x contains $\langle 1, x \rangle$, while the kernel of $D_x = \langle 1, w \rangle$ with w a non-constant function with discontinuities in the first derivative at each cell interface. Moreover, the kernel of Z_x does not contain w.*

The proof is trivial for D_x^x and D_x, while the proof for Z_x can be found in Appendix D.

Example 5.1 (Analysis of the one-dimensional operator kernels for $\mathbb{P}^2$)**.** *In the $\mathbb{P}^1$ case, i.e. for Finite Differences, one usually associates spurious modes with the checkerboard mode $t_x = -1$. One can show that D_x for $\mathbb{P}^2$ Finite Elements has a non-trivial kernel iff $t_x = 1$. This does not mean, though, that it is checkerboard-free, because*

$$\ker \mathbb{F}_{t_x}(D_x) = \operatorname{span}\left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) \tag{108}$$

and thus the values $q_{i,0}$ and $q_{i,1}$ can be specified independently. The opposite case is exemplified by D_x^x whose kernel is also nontrivial iff $t_x = 1$, but

$$\ker \mathbb{F}_{t_x}(D_x^x) = \operatorname{span}\left(\begin{pmatrix} 1 \\ 1 \end{pmatrix} \right), \tag{109}$$

such that it contains only uniform constants, no checkerboards.

5.3.2 Global flux operator kernels in two dimensions

When we consider the two-dimensional extension of these operators, we can still focus on similar function spaces, namely $V_h^K(\Omega_h), V_{h,0}^K(\Omega_h), V_{h,b}^{K-1}(\Omega_h)$, obtaining for $(U+V)_h \in V_h^K(\Omega_h)$

$$D_x \otimes D_y(U+V) = \int_{\Omega_h} \varphi(x,y)\partial_x\partial_y(U+V)_h(x,y)\mathrm{d}x\mathrm{d}y, \qquad \forall \varphi \in V_{h,0}^K(\Omega_h), \tag{110}$$

$$D_x^x \otimes D_y(U+V) = \int_{\Omega_h} \partial_x\varphi(x,y)\partial_x\partial_y(U+V)_h(x,y)\mathrm{d}x\mathrm{d}y, \qquad \forall \varphi \in V_{h,0}^K(\Omega_h), \tag{111}$$

$$D_x \otimes D_y^y(U+V) = \int_{\Omega_h} \partial_y\varphi(x,y)\partial_x\partial_y(U+V)_h(x,y)\mathrm{d}x\mathrm{d}y, \qquad \forall \varphi \in V_{h,0}^K(\Omega_h). \tag{112}$$

Again, we notice that $V_h^K(\Omega_h) \sim \mathbb{R}^{(N_xK+1)\times(N_yK+1)}$, $V_{h,0}^K(\Omega_h) \sim \mathbb{R}^{(N_xK-1)\times(N_yK-1)}$ and $V_{h,0}^{K-1}(\Omega_h) \sim \mathbb{R}^{(N_xK)\times(N_yK)}$, and that the operator $\partial_x\partial_y : V_h^K(\Omega_h) \to V_{h,0}^{K-1}(\Omega_h)$ has a kernel of dimension $N_xK + N_yK + 1$. Kernel bases trivially include $\{(0,\dots,0,1,0,\dots,0) \otimes (1,\dots,1)\}$ (these are N_xK+1), and $\{(1,\dots,1) \otimes (0,\dots,0,1,0,\dots,0)\}$ (these are N_yK+1). The spaces generated by these two sets have an intersection which is of dimension one and generated by $(1,\dots,1)\otimes(1,\dots,1)$. Hence, we have a clear description of the kernel of $\partial_x\partial_y$ which consists of the constant-by-line solutions: the interesting steady states that are automatically in the kernel of all GF operators. We can now focus on the spurious modes.

Define $\widetilde{D_x \otimes D_y}, \widetilde{D_x^x \otimes D_y}, \widetilde{D_x \otimes D_y^y} : V_{h,b}^{K-1}(\Omega_h) \to V_{h,0}^K(\Omega_h)$ by their actions on $z_h \in V_{h,b}^{K-1}(\Omega_h)$ as

$$\widetilde{D_x \otimes D_y} z_h = \int_{\Omega_h} \varphi(x,y) z_h(x,y)\mathrm{d}x\mathrm{d}y, \qquad \forall \varphi \in V_{h,0}^K(\Omega_h), \tag{113a}$$

$$\widetilde{D_x^x \otimes D_y} z_h = \int_{\Omega_h} \partial_x\varphi(x,y) z_h(x,y)\mathrm{d}x\mathrm{d}y, \qquad \forall \varphi \in V_{h,0}^K(\Omega_h), \tag{113b}$$

$$\widetilde{D_x \otimes D_y^y} z_h = \int_{\Omega_h} \partial_y\varphi(x,y) z_h(x,y)\mathrm{d}x\mathrm{d}y, \qquad \forall \varphi \in V_{h,0}^K(\Omega_h). \tag{113c}$$

Note that $\widetilde{D_x \otimes D_y} = \tilde{D}_x \otimes \tilde{D}_y$, $\widetilde{D_x^x \otimes D_y} = \tilde{D}_x^x \otimes \tilde{D}_y$ and $\widetilde{D_x \otimes D_y^y} = \tilde{D}_x \otimes \tilde{D}_y^y$, and that the kernel of the Kronecker product of two such operators is given by the space

$$\ker(A_x \otimes B_y) = \ker(A_x) \otimes V_{\Delta y}^K(\Omega_{\Delta y}^y) + V_{\Delta x}^K(\Omega_{\Delta x}^x) \otimes \ker(B_y). \tag{114}$$

Then, using Theorem 5.3, we can infer many things on the kernel of the two-dimensional operators. First of all we notice that

$$D_x \otimes D_y \Phi = 0 \Longleftrightarrow \Phi = \Phi_x + \Phi_y$$

with $\Phi_x \in \ker(D_x) \otimes \mathbb{R}^{N_y K+1}$ and $\Phi_y \in \mathbb{R}^{N_x K+1} \otimes \ker(D_y)$. This means that

$$\begin{aligned} \Phi_x &= 1 \otimes g_1 + w \otimes g_2 \text{ with } g_1, g_2 \in \mathbb{R}^{N_y K+1}, \\ \Phi_y &= f_1 \otimes 1 + f_2 \otimes w \text{ with } f_1, f_2 \in \mathbb{R}^{N_x K+1}. \end{aligned} \tag{115}$$

Let us focus on Φ_y, as the same holds for the x component. We can distinguish between the desired equilibria $f_1 \otimes 1$ and the checkerboard spurious modes $f_2 \otimes w$. Clearly, $f_1 \otimes 1$ belongs also to the kernel of the stabilization $D_x \otimes D_y^y$ and $D_x^x \otimes D_y$ as $1 \in \ker(D_y)$ and $1 \in \ker(D_y^y)$; so the desired equilibria are preserved. At the same time,

$$(D_x \otimes D_y^y)(f_2 \otimes w) \neq 0 \Longleftrightarrow f_2 \notin \ker(D_x).$$

So, the spurious modes of $D_x \otimes D_y$ are diffused away by the stabilization operators, except for those generated by $1 \otimes w, w \otimes 1$, and $w \otimes w$. In principle, these modes could be filtered out by boundary conditions or initial conditions.

Example 5.2 (Fourier analysis of the operator kernels for $\mathbb{Q}^1$)**.** *Recall that for $\mathbb{Q}^1$ Finite Elements, the characteristic polynomials of the relevant operators are*

$$\ker(D_x \otimes D_y) = \left\{ \Phi : \frac{(t_x^2-1)}{2t_x\Delta x} \frac{(t_y^2-1)}{2t_y\Delta y} \Phi = 0 \right\}, \tag{116a}$$

$$\ker(D_x^x \otimes D_y) = \left\{ \Phi : \frac{(t_x-1)^2}{t_x\Delta x} \frac{(t_y^2-1)}{2t_y\Delta y} \Phi = 0 \right\}, \tag{116b}$$

$$\ker(D_x \otimes D_y^y) = \left\{ \Phi : \frac{(t_x^2-1)}{2t_x\Delta x} \frac{(t_y-1)^2}{t_y\Delta y} \Phi = 0 \right\}. \tag{116c}$$

The kernel of $D_x \otimes D_y$ contains functions that are either constant in one of the directions, or are checkerboards $\mathfrak{c} \in \mathfrak{C}$ in one of the directions:

$$\mathfrak{C}_x = \{\varphi \neq 0 : (t_x+1)\varphi = 0\}, \qquad \mathfrak{C}_y = \{\varphi \neq 0 : (t_y+1)\varphi = 0\}, \qquad \mathfrak{C} := \mathfrak{C}_x \cup \mathfrak{C}_y. \tag{117}$$

One easily can verify the inclusions

$$\ker(D_x^x \otimes D_y) \subset \ker(D_x \otimes D_y), \qquad \ker(D_x \otimes D_y^y) \subset \ker(D_x \otimes D_y). \tag{118}$$

However, not all the checkerboards are damped by the numerical diffusion:

$$\mathfrak{a} \in \mathfrak{C}_x \Rightarrow \begin{cases} a \notin \ker(D_x^x \otimes D_y) & \text{if } t_y^2 \neq 1, \\ a \in \ker(D_x^x \otimes D_y) & \text{if } t_y^2 = 1. \end{cases} \tag{119}$$

The checkerboards not dissipated are

$$\{\varphi \neq 0 : (t_x+1)(t_y^2-1)\varphi = 0\} \cup \{\varphi \neq 0 : (t_y+1)(t_x^2-1)\varphi = 0\}. \tag{120}$$

5.4 Discrete involutions

Involutions of the SUPG-GFq matrix $\mathcal{E}_{\text{SUPG-GFq}}$ (96) can be found by using the characteristic polynomial / Fourier representation of the scheme, following [1] as done in Section 3.1.

The left kernel (the kernel of the transpose, related to the stationary involution) of $\mathcal{E}_{\text{SUPG-GFq}}$ in the $\mathbb{Q}^1$ case is parallel to

$$\begin{pmatrix} \mathbb{F}_{t_x}(D_x)\Big(\mathbb{F}_{t_y}(D_y)^2\mathbb{F}_{t_x}(M_x) - \alpha^2h^2\mathbb{F}_{t_y}(D_y^y)\Big(\mathbb{F}_{t_y}(D_y^y)\mathbb{F}_{t_x}(M_x) + \mathbb{F}_{t_x}(D_x^x)\mathbb{F}_{t_y}(M_y)\Big)\Big) \\ \mathbb{F}_{t_y}(D_y)\Big(- \mathbb{F}_{t_x}(D_x)^2\mathbb{F}_{t_y}(M_y) + \alpha^2h^2\mathbb{F}_{t_x}(D_x^x)\Big(\mathbb{F}_{t_y}(D_y^y)\mathbb{F}_{t_x}(M_x) + \mathbb{F}_{t_x}(D_x^x)\mathbb{F}_{t_y}(M_y)\Big)\Big) \\ \alpha h\left(-\mathbb{F}_{t_x}(D_x^x)\mathbb{F}_{t_y}(D_y)^2\mathbb{F}_{t_x}(M_x) + \mathbb{F}_{t_x}(D_x)^2\mathbb{F}_{t_y}(D_y^y)\mathbb{F}_{t_y}(M_y)\right) \end{pmatrix}^T \tag{121}$$

Observe that it does not depend on I_x, I_y.

Proposition 5.4 (Involutions of SUPG-GFq)**.** *There exist K^2 discrete involutions remaining stationary for any initial data subject to evolution according to SUPG-GFq with $\mathbb{Q}^K$ FEM.*

Proof. We use the fact that the SUPG-GFq method for linear acoustics can be written as a function of $U+V$ and p, instead of u, v, p individually. This is obvious from Equation (94). In particular the 3×3 (block) matrix $\mathcal{E}_{\text{SUPG}-\text{GFq}}$ from (96b) can be written as

$$\mathcal{E}_{\text{SUPG}-\text{GFq}}\begin{pmatrix} u \\ v \\ p \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha h D_x^x \otimes D_y & D_x \otimes M_y \\ \alpha h D_x \otimes D_y^y & M_x \otimes D_y \\ D_x \otimes D_y & \alpha h D_x^x \otimes M_y + \alpha h M_x \otimes D_y^y \end{pmatrix}}_{=:\hat{\mathcal{E}}_{\text{SUPG}-\text{GFq}}} \begin{pmatrix} U+V \\ p \end{pmatrix} \tag{122}$$

Matrix $\hat{\mathcal{E}}_{\text{SUPG}-\text{GFq}}$, for $\mathbb{Q}^k$ FEM, has $3K^2$ rows and $2K^2$ columns. Its left kernel therefore is at least K^2-dimensional. □

However, in practice it is rather difficult to explicitly express the preserved involution. In particular, one cannot simply extend the result of $\mathbb{Q}^1$. Defining

$$\mathcal{K} := D_x^x \otimes M_y + M_x \otimes D_y^y, \quad \omega := \begin{pmatrix} D_x\Big(M_x \otimes D_y^2 - \alpha^2h^2 D_y^y\mathcal{K}\Big) \\ D_y\Big(- D_x^2 \otimes M_y + \alpha^2h^2 D_x^x\mathcal{K}\Big) \\ \alpha h\left(-M_xD_x^x \otimes D_y^2 + D_x^2 \otimes M_yD_y^y\right) \end{pmatrix}^{\mathrm{T}} \tag{123}$$

we find

$$(\omega\hat{\mathcal{E}}) = (v_1 \;\; v_2)^T$$

with

$$v_1 = \alpha h\Big\{[D_x, M_xD_x^x] \otimes D_y^3 + D_x^3 \otimes [M_yD_y^y, D_y] + \alpha^2h^2\left((D_x^x \otimes D_y)[\mathcal{K}, D_x \otimes D_y^y] + [D_x^x \otimes D_y, (D_x \otimes D_y^y)\mathcal{K}]\right)\Big\},$$

$$v_2 = D_x[M_x, D_x] \otimes D_y[D_y, M_y] + \alpha^2h^2[(D_x^x \otimes D_y)\mathcal{K}, M_x \otimes D_y] + \alpha^2h^2[D_x \otimes M_y, (D_x \otimes D_y^y)\mathcal{K}].$$

The appearance of commutators in each term in general prevent ω from being an involution. This is due to the fact that one cannot perform the same computations with block matrices as with usual matrices if the blocks do not commute. It is only for $K=1$ that the matrices reduce to scalars and (123) is indeed the involution.

The left kernel of $\mathcal{E}_{\text{OSS-GFq}}$ in (102b) is, for $\mathbb{Q}^1$,

$$\begin{pmatrix} \mathbb{F}_{t_x,t_y}(D_y)\mathbb{F}_{t_x,t_y}(M_x) + \alpha^2h^2\mathcal{K}_u \\ -\mathbb{F}_{t_x,t_y}(D_x)\mathbb{F}_{t_x,t_y}(M_y) + \alpha^2h^2\mathcal{K}_v \\ \alpha h\left(-\frac{\mathbb{F}_{t_x,t_y}(D_x^x)\mathbb{F}_{t_x,t_y}(D_y)\mathbb{F}_{t_x,t_y}(M_x)}{\mathbb{F}_{t_x,t_y}(D_x)} + \frac{\mathbb{F}_{t_x,t_y}(D_x)\mathbb{F}_{t_x,t_y}(D_y^y)\mathbb{F}_{t_x,t_y}(M_y)}{\mathbb{F}_{t_x,t_y}(D_y)}\right) \end{pmatrix}, \tag{124}$$

where $\mathcal{K}_u$ and $\mathcal{K}_v$ are reported in Appendix C. It is a consistent discretization of $(\partial_y, -\partial_x, 0)^{\mathrm{T}}$.

6 Time stepping via Deferred Correction

In Equations (41), we have described the classical SUPG discretization of the linear acoustic system within a time-continuous framework, while in (95) we have modified the spatial discretization to obtain a GF formulation that is vorticity preserving. Similarly, in Section 5.2 we have introduced the spatial discretization of the OSS and its GF version. The goal of this section is to introduce an arbitrarily high-order time discretization.

The deferred correction is a class of arbitrarily high-order time integrations that are based on a space–time residual formulation [31, 32, 33]. This residual is then approximated with a certain level of accuracy that matches the order of the space–time discretization. We start by introducing Lagrangian basis functions in time (we will use Gauss-Lobatto Lagrangian basis functions). Then, we define 2 time operators, following [34]: a high-order one that corresponds to an implicit collocation RK, in this case to the Lobatto IIIA, and a low-order one that corresponds to an explicit RK, in our case the explicit Euler method. The implicit high-order operator will be based on the residual that we want to minimize, while the explicit one is a simple aid to set up an iterative process.

The implicit operator can be seen as a high-order FEM discretization in space and time. We focus, for simplicity, on the ODE

$$u' + F(u) = 0 \tag{125}$$

inside one timestep $[t_n, t_{n+1}]$, as for every one-step method. In this timestep, we define M sub-timesteps through $M+1$ sub-timenodes $t_n = t^0 < \cdots < t^m < \cdots < t^M = t_{n+1}$ and the Lagrangian interpolating polynomials $\gamma_r(t)$ for $r = 0, \ldots, M$. We then denote with the superscript r the approximation of the solution q at the sub-timenode t^r, i.e., $q^r \approx q(t^r)$. Now, for each sub-timenode $m = 1, \ldots, M$, the high-order time discretization operator reads

$$T^{2,m}(\underline{q}) = \frac{q^m - q^0}{\Delta t} + \frac{1}{\Delta t}\int_{t^0}^{t^m} F\left(\sum_{r=0}^{M} \gamma_r(t) q^r\right) dt. \tag{126}$$

Then, using as quadrature points the same Lagrangian subtimenodes, we have

$$T^{2,m}(\underline{q}) = \frac{q^m - q^0}{\Delta t} + \sum_{r=0}^{M} \frac{1}{\Delta t}\int_{t^0}^{t^m} \gamma_r(t)\, dt\, F\left(q^r\right) = \frac{q^m - q^0}{\Delta t} + \sum_{r=0}^{M} \vartheta_r^m F\left(q^r\right), \tag{127}$$

with $\vartheta_r^m = \frac{1}{\Delta t}\int_{t^0}^{t^m} \gamma_r(t) dt$, which are independent on Δt. The simple explicit Euler operator, instead, will read

$$T^{1,m}(\underline{q}) = \frac{q^m - q^0}{\Delta t} + \sum_{r=0}^{M} \frac{1}{\Delta t}\int_{t^0}^{t^m} \gamma_r(t)\, dt\, F\left(q^0\right) = \frac{q^m - q^0}{\Delta t} + \beta^m F\left(q^0\right), \tag{128}$$

with $\beta^m = \frac{t^m - t^0}{\Delta t}$. Moreover, as in our case, mass matrices or more complex spatial discretizations can be included in both operators. One can further simplify the T^1 operator by using a lumped first-order approximation version of the mass matrix in front of the term $\frac{q^m - q^0}{\Delta t}$, leading to an explicit matrix-free solver for $T^1(\underline{q}) = \underline{r}$.

The DeC iterative method is then defined as

$$\begin{cases} \underline{q}^{(0)} = q(t^0) = q_n, \\ T^1(\underline{q}^{(p)}) = T^1(\underline{q}^{(p-1)}) - T^2(\underline{q}^{(p-1)}), \qquad \text{for } p = 1, \ldots, P, \\ q_{n+1} = q^{(P)}(t^M), \end{cases} \tag{129}$$

with P being the order of accuracy of the scheme. Notice that each iteration p implies the solution of M systems that are explicit and matrix–free. Hence, for each time step, in total, we need to compute around MP equivalent RK stages, more precisely $M(P-1)+1$.

Proposition 6.1 (Order of accuracy of DeC time integrator [35, 36])**.** *The order of accuracy of the DeC is the minimum between the number of iterations P and the order of the time discretization Q. For Gauss–Lobatto nodes it is* $\min(P, 2M)$*, with M the number of the subtimesteps.*

Now, for clarity, we discuss the SUPG/OSS spatial discretization in the matrix formulation introduced in (43d) and in (62), respectively. We will give in Appendix B the expansion of the space-time discretization in each equation, for ease in reproducibility (only for the SUPG case). The T^2 operator encompasses the SUPG/OSS residual, so, differently from the ODE case, we have to insert also the mass matrix term. It is defined for $m = 1, \dots, M$ as

$$T^{2,m}(\underline{q}) = \mathcal{A}\frac{q^m - q^0}{\Delta t} + \sum_{r=0}^{M} \vartheta_r^m \mathcal{E} q^r. \tag{130}$$

On the other side, the T^1 low order operator is a simplified version of T^2, in particular, the mass matrix is a simple lumped version of the mass matrix, i.e., $(L_x)_{\alpha,\beta} = \delta_{\alpha,\beta} \int_{\Omega^x_{\Delta x}} \varphi^x_\alpha(x)\mathrm{d}x$, which we define as

$$\mathcal{L} := \begin{pmatrix} L_x \otimes L_y & 0 & 0 \\ 0 & L_x \otimes L_y & 0 \\ 0 & 0 & L_x \otimes L_y \end{pmatrix}. \tag{131}$$

Hence, the T^1 operator can be defined for $m = 1, \dots, M$ as

$$T^{1,m}(\underline{q}) = \mathcal{L}\frac{q^m - q^0}{\Delta t} + \beta^m \mathcal{E} q^0. \tag{132}$$

The update formula in (129), after the simplification of the terms in the T^1 operators reads for every $p = 1, \dots, P$ and every $m = 1, \dots, M$

$$0 = \mathcal{L}\frac{q^{(p),m} - q^{(p-1),m}}{\Delta t} + \mathcal{A}\frac{q^{(p-1),m} - q^0}{\Delta t} + \sum_{r=0}^{M} \vartheta_r^m \mathcal{E} q^{(p-1),r}, \tag{133}$$

where $q^{(p),m}$ is the only unknown term and $\mathcal{L}$ is a diagonal matrix.

Proposition 6.2 (Order of accuracy of the space–time DeC)**.** *The SUPG–DeC/OSS–DeC space time discretization presented above is of order* $\min(K + 1, 2M, P)$*.*

Proof. The proof follows the ones of [34, 35] with the details on the SUPG/OSS discretization defined in [29, 20]. □

Proposition 6.3 (Steady states of the DeC)**.** *If $\bar{q}$ is such that $\mathcal{E}\bar{q} = 0$, then $\bar{q}$ is a steady state of the DeC, i.e., if $q_n = \bar{q}$ then $q_{n+1} = \bar{q}$.*

Proof. We proceed by induction on p, for all m, showing that $q^{(p),m} = q_n = \bar{q}$. By definition of the DeC, for $p = 0$ we set all $q^{(0),m} = \bar{q}$ and for $m = 0$ we set all $q^{(p),0} = \bar{q}$. Then, for $p = 1, \dots, P$ and $m = 1, \dots, M$ we have that

$$q^{(p),m} = q^{(p-1),m} - \mathcal{L}^{-1}\mathcal{A}(q^{(p-1),m} - q^0) - \Delta t \mathcal{L}^{-1} \sum_{r=0}^{M} \vartheta_r^m \mathcal{E} q^{(p-1),r} = q^{(p-1),m} = \bar{q}, \tag{134}$$

as for $p - 1$ we have that $\mathcal{E} q^{(p-1),r} = 0$ for all $r = 0, \dots, M$ and that $q^{(p-1),m} = q^0 = \bar{q}$ for all $m = 1, \dots, M$. Hence, also $q_{n+1} = q^{(P),M} = \bar{q}$. □

The consequence of this theorem is that for the SUPG-GFq and OSS-GFq discretizations $\mathcal{E}$, we know a class of steady states, hence, the DeC time integration method will preserve them. This allows us to easily converge towards the steady states, or to set up initial conditions that verify the condition $\mathcal{E}q = 0$ and to preserve them.

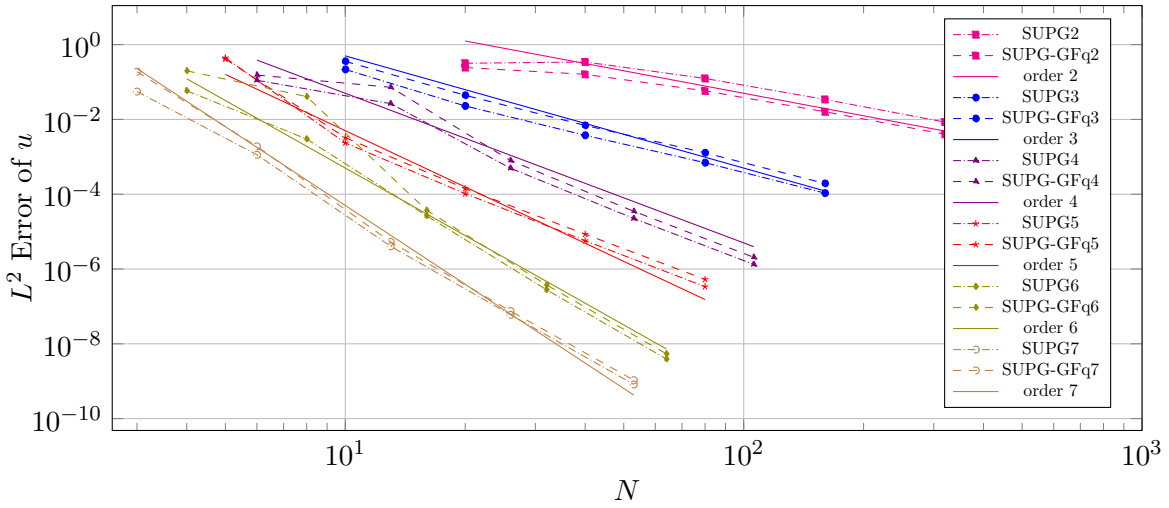

Figure 2: Oblique flow: convergence of L^2 error in u w.r.t. the number of elements in x

7 Numerical results

In this section, we show the benefits of the proposed formulation through various numerical tests. In all computations we have used Gauss–Lobatto points both for interpolation and quadrature.

7.1 Convergence analysis on a smooth oblique flow

To show the arbitrarily high-order property of the SUPG and SUPG-GFq DeC-FEM methods of Section 4, we use a smooth two-dimensional problem of an oblique wave on the square $[0,1]^2$ with periodic boundary conditions. Its analytical solution for the linear acoustic equations (1) is

$$\begin{cases} u(x,y,t) = -\frac{1}{2c}\left(\cos(\alpha\xi(x,y)+ct) - \cos(\alpha\xi(x,y)-ct)\right)\cos(\vartheta), \\ v(x,y,t) = -\frac{1}{2c}\left(\cos(\alpha\xi(x,y)+ct) - \cos(\alpha\xi(x,y)-ct)\right)\sin(\vartheta), \\ p(x,y,t) = \frac{1}{2}\left(\cos(\alpha\xi(x,y)+ct) + \cos(\alpha\xi(x,y)-ct)\right), \end{cases} \tag{135}$$

with $\alpha = \frac{2\pi}{\lambda\cos(\vartheta)}$ with $\vartheta = \frac{\pi}{4}$ and $\lambda = \frac{1}{4}$. We run the simulations up to $T = 1$.

The errors are compared for the two methods in Figure 2 (for u). The SUPG and SUPG–GF methods provide very similar errors and both of them converge with the expected order of accuracy.

7.2 Divergence-free solutions

In this section, we will consider two analytical solutions that are divergence-free. Both have the velocity field of a vortex and a constant pressure. Both are be defined on the unit square with Dirichlet boundary conditions and centered in $(x_0, y_0) = (0.5, 0.5)$. The first one is a compactly supported solution in $\mathcal{C}^6$, while the second one is used for convergence purposes and is a $\mathcal{C}^\infty$ compactly supported function.

Both can be written as

$$\begin{cases} u(x,y) = f(\rho(x,y)\cdot(y-y_0) \\ v(x,y) = -f(\rho(x,y))\cdot(x-x_0) \\ p(x,y) = 1 \end{cases} \tag{136}$$

with $\rho(x,y) = \frac{\sqrt{(x-x_0)^2+(y-y_0)^2}}{r_0}$ with $r_0 = 0.45$ the radius of the support. The first test case is defined by

$$f(\rho) = \gamma(1+\cos(\pi\rho))^2, \tag{137}$$

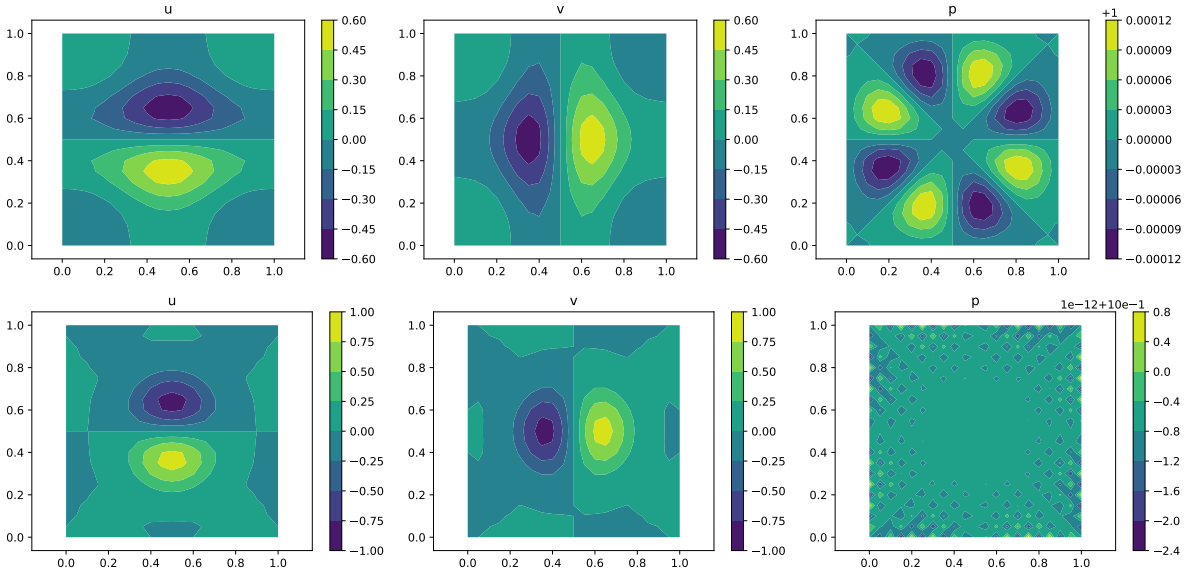

Figure 3: Simulation at time $T = 100$ of the vortex (137) with 20×20 cells and $\mathbb{P}^1$ elements for the SUPG (top) and SUPG–GF (bottom) schemes

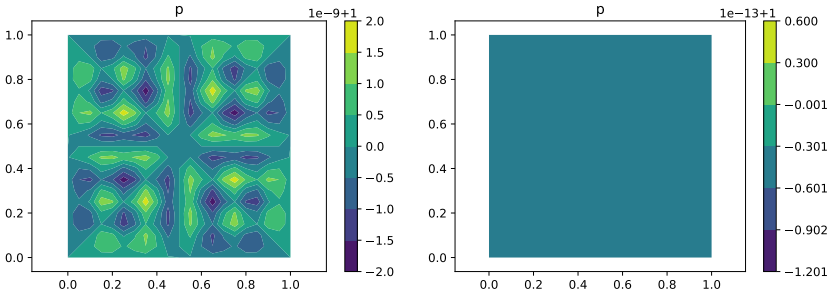

Figure 4: Simulation at time $T = 100$ of the vortex (137) (only p) with 10×10 cells and $\mathbb{P}^2$ elements for the SUPG (left) and SUPG–GF (right) schemes

with $\gamma = \frac{12\pi\sqrt{0.981}}{r_0\sqrt{315\pi^2 - 2048}}$, see [37] for the origin of these solutions. The second is defined by

$$f(\rho) = 2\gamma e^{-\frac{1}{2(1-\rho)^2}} \sqrt{\frac{g}{r_0(1-\rho)^3}} \tag{138}$$

with $g = 9.81$, $\gamma = 0.2$ if $\rho < 1$, else 0.

For the first vortex (137), let us consider some qualitative results on a coarse mesh. In Figure 3, we compare the solution at time $T = 100$ for $\mathbb{Q}^1$ SUPG and SUPG-GFq methods on a grid of 20×20 cells. On the one hand, for long simulation times the SUPG scheme leads to vertical/horizontal artifacts that are not physical and the pressure does not converge to a constant state. On the other hand, the SUPG-GFq does not show this behavior and converges to the divergence-free state with constant pressure. In Figure 4, we compare the pressures for $\mathbb{Q}^2$ schemes and observe the same behavior on a smaller scale. SUPG–GFq obtains a constant pressure up to machine precision, while SUPG has oscillations of the order of 10^{-9}.

In Figure 5, we show the norm of the discrete divergence in time. For the SUPG-GFq schemes it decays exponentially until reaching machine precision (higher orders decay faster than low order schemes). For the SUPG schemes the decay is much slower and it depends heavily on the order of accuracy. In particular, the second order method is very inaccurate, while the fourth order scheme reaches small values of divergence at the final time.

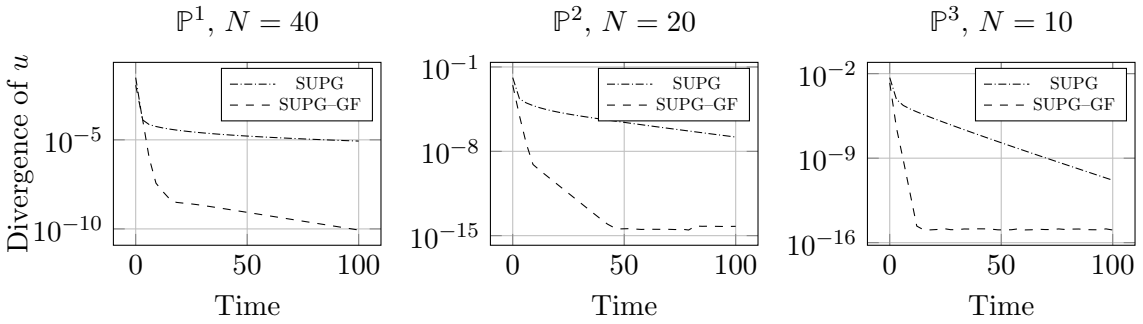

Figure 5: Norm of the discrete divergence of $\mathbf{v}$ for SUPG ($D_x \otimes M_y u + M_x \otimes D_y v$) and SUPG-GFq ($D_x \otimes (D_y I_y)u + (D_x I_x) \otimes D_y v$) as a function of time for different orders of accuracy

Table 1: Smooth vortex convergence results $\mathbb{P}^1$: without GF (left) and with GF (right)

N	err u	err v	err p	ord u	ord v	ord p	N	err u	err v	err p	ord u	ord v	ord p
20	4.36e-03	4.36e-03	2.06e-03	0.00	0.00	0.00	20	1.60e-03	1.60e-03	8.16e-04	0.00	0.00	0.00
40	1.17e-03	1.17e-03	4.19e-04	1.90	1.90	2.30	40	3.13e-04	3.13e-04	1.75e-04	2.35	2.35	2.22
80	2.44e-04	2.44e-04	9.47e-05	2.26	2.26	2.15	80	7.39e-05	7.39e-05	4.62e-05	2.08	2.08	1.92
160	4.79e-05	4.79e-05	2.35e-05	2.35	2.35	2.01	160	1.84e-05	1.84e-05	1.17e-05	2.00	2.00	1.98
320	1.01e-05	1.01e-05	5.78e-06	2.24	2.24	2.02	320	4.60e-06	4.60e-06	2.89e-06	2.00	2.00	2.02

Table 2: Smooth vortex convergence results $\mathbb{P}^2$: without GF (left) and with GF (right)

N	err u	err v	err p	ord u	ord v	ord p	N	err u	err v	err p	ord u	ord v	ord p
10	5.06e-03	5.06e-03	1.87e-03	0.00	0.00	0.00	10	1.91e-03	1.91e-03	9.07e-04	0.00	0.00	0.00
20	1.61e-03	1.61e-03	4.45e-04	1.65	1.65	2.07	20	2.12e-04	2.12e-04	8.69e-05	3.17	3.17	3.38
40	4.16e-04	4.16e-04	6.95e-05	1.95	1.95	2.68	40	1.76e-05	1.76e-05	5.92e-06	3.59	3.59	3.87
80	1.03e-04	1.03e-04	1.22e-05	2.01	2.01	2.51	80	1.14e-06	1.14e-06	4.57e-07	3.95	3.95	3.70
160	2.32e-05	2.32e-05	2.43e-06	2.15	2.15	2.33	160	7.14e-08	7.14e-08	3.21e-08	4.00	4.00	3.83

Table 3: Smooth vortex convergence results $\mathbb{P}^3$: without GF (left) and with GF (right)

N	err u	err v	err p	ord u	ord v	ord p	N	err u	err v	err p	ord u	ord v	ord p
6	7.32e-03	7.32e-03	2.61e-03	0.00	0.00	0.00	6	1.57e-03	1.57e-03	6.72e-04	0.00	0.00	0.00
13	1.21e-03	1.21e-03	3.25e-04	2.33	2.33	2.69	13	1.90e-04	1.90e-04	6.68e-05	2.73	2.73	2.99
26	1.84e-04	1.84e-04	1.63e-05	2.71	2.71	4.32	26	1.13e-05	1.13e-05	3.61e-06	4.07	4.07	4.21
53	1.69e-05	1.69e-05	1.04e-06	3.35	3.35	3.86	53	4.06e-07	4.06e-07	1.10e-07	4.67	4.67	4.90
106	1.29e-06	1.29e-06	6.62e-08	3.71	3.71	3.98	106	1.34e-08	1.34e-08	1.88e-09	4.92	4.92	5.87

Table 4: Smooth vortex convergence results $\mathbb{P}^4$: without GF (left) and with GF (right)

N	err u	err v	err p	ord u	ord v	ord p	N	err u	err v	err p	ord u	ord v	ord p
5	4.30e-03	4.30e-03	1.44e-03	0.00	0.00	0.00	5	1.60e-03	1.60e-03	7.29e-04	0.00	0.00	0.00
10	8.86e-04	8.86e-04	1.09e-04	2.28	2.28	3.73	10	1.95e-04	1.95e-04	5.55e-05	3.04	3.04	3.72
20	8.87e-05	8.87e-05	6.96e-06	3.32	3.32	3.96	20	7.59e-06	7.59e-06	1.81e-06	4.68	4.68	4.94
40	5.47e-06	5.47e-06	2.59e-07	4.02	4.02	4.75	40	1.86e-07	1.86e-07	3.80e-08	5.35	5.35	5.57
80	3.35e-07	3.35e-07	1.48e-08	4.03	4.03	4.13	80	3.29e-09	3.29e-09	6.75e-10	5.82	5.82	5.82

Table 5: Smooth vortex convergence results $\mathbb{P}^5$: without GF (left) and with GF (right)

N	err u	err v	err p	ord u	ord v	ord p	N	err u	err v	err p	ord u	ord v	ord p
4	4.25e-03	4.25e-03	1.16e-03	0.00	0.00	0.00	4	2.03e-03	2.03e-03	9.83e-04	0.00	0.00	0.00
8	6.68e-04	6.68e-04	8.80e-05	2.67	2.67	3.72	8	1.46e-04	1.46e-04	3.38e-05	3.80	3.80	4.86
16	5.35e-05	5.35e-05	2.93e-06	3.64	3.64	4.91	16	6.25e-06	6.25e-06	1.32e-06	4.55	4.55	4.68
32	2.87e-06	2.87e-06	1.22e-07	4.22	4.22	4.59	32	1.08e-07	1.08e-07	2.19e-08	5.85	5.85	5.91
64	8.52e-08	8.52e-08	2.48e-09	5.08	5.08	5.62	64	1.11e-09	1.11e-09	1.13e-10	6.61	6.61	7.60

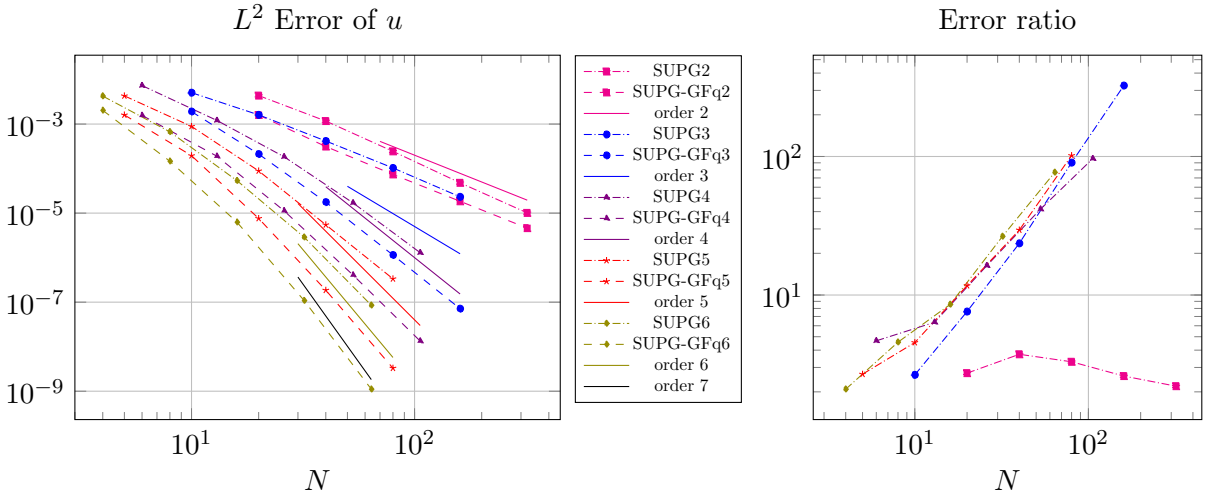

Figure 6: Smooth vortex: convergence of L^2 error of u with respect to the number of elements in x (left) and error ratios between SUPG and SUPG-GFq (right)

For the smooth vortex (138), we test the convergence for arbitrarily high order at final time $T = 1$. Although the solution is $\mathcal{C}^\infty$, the spatial derivatives of the solution are quite steep [37]. Hence, it is not trivial to observe the right convergence rate for coarse meshes. Nevertheless, in Tables 1-5 we see that all SUPG-GFq solutions reach the expected order, while SUPG seems to struggle at this objective. Moreover, the errors magnitude for the GF formulation are significantly smaller than for the classical one, as much as by two orders of magnitude. In Figure 6, we depict the errors for u and the ratio of the SUPG errors and the SUPG-GFq ones. SUPG-GFq methods of lower order can outperform the SUPG method, see for instance how close SUPG-GFq-4 and SUPG-6 are.

7.3 Perturbation of divergence–free solutions

In this section, we study the behavior of the schemes when a perturbation is applied to an equilibrium state. Typically, one is not aware of the steady state before running the simulation, moreover, the class of steady states of (1) is quite rich and it is not fully determined by an external datum, like for 1D or 1D-like source-driven equilibria [38].

Several options are available to run such setups.

1. The first obvious possibility is to just use **analytical initial conditions** to start the simulations. This, however, might lead to a loss in accuracy in the early stages of the simulations.

2. A second approach consists in running **long time simulations**, reaching the equilibrium solution and to add the perturbation afterwards. This approach guarantees that a discrete equilibrium is used as initial condition. This strategy is natural for simulations in which the equilibrium is not known *a priori*. However, it can be expensive to run a long time simulation, just to have a short simulation of the perturbation.

3. Finally, **discretely well-prepared initial data** can be first obtained and the perturbation added to those. We propose two approaches to reach such a goal: an optimization of the analytical initial condition constrained to the discrete div-free property and a line-by-line reconstruction of the initial conditions. The description of the two strategies is given in Section 4.3.

To start the discussion with the analytical initial condition, we want to first check the amount of discrete divergence carried by the analytical IC. We display in Figure 7 the norm of the discrete

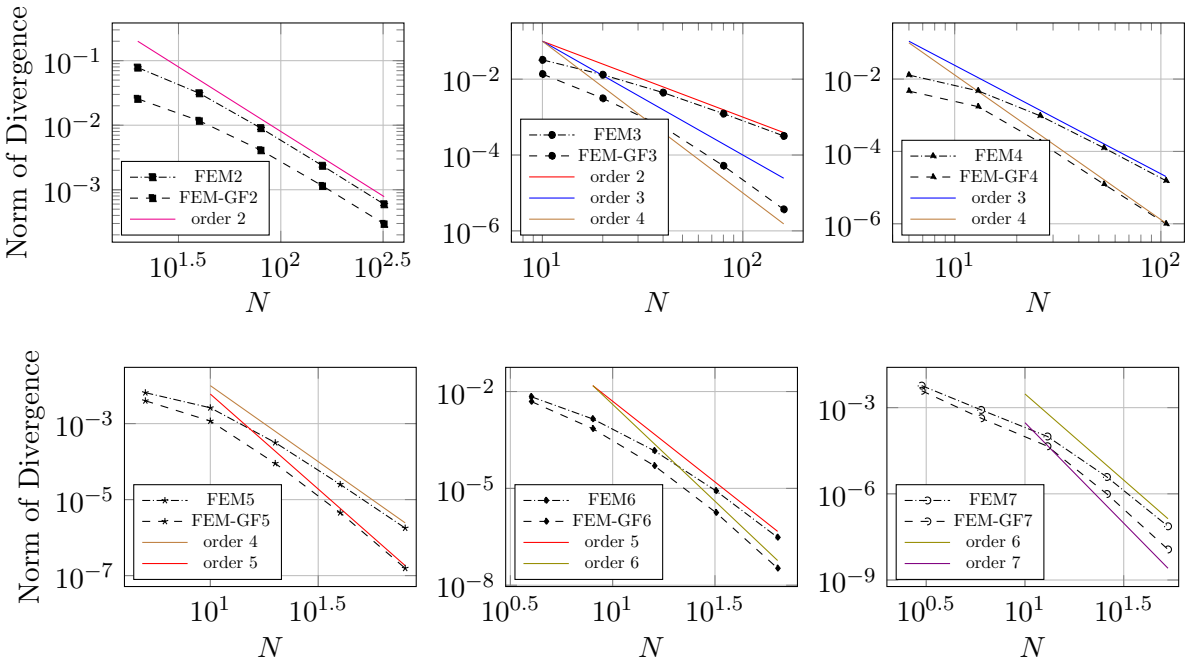

Figure 7: Smooth vortex: convergence of divergence operator on exact IC with respect to the number of elements in x

divergence ($\tilde{D}_x u + \tilde{D}_y v$) of the analytical initial conditions of the $\mathcal{C}^\infty$ vortex (138). We have used Gauss–Legendre polynomial $\mathbb{P}^p$ and clearly see some super-convergence pattern. The divergence should decay with order p, but we observe in most of the cases $p+1$ convergence and for $\mathbb{P}^2$ we observe order 4. Comparing it with the classical FEM divergence, in Figure 7, we observe that the GF divergence is much more accurate and gains one extra order of accuracy with respect to the classical method (except for $\mathbb{P}^1$ where also FEM gains it and for $\mathbb{P}^3$ where GF–FEM gains 2 order of accuracy). Nevertheless, such an error is still too high to preserve a perturbation of an equilibrium.

Let us add a perturbation to the pressure in the form of a Gaussian centered in $x_p = (0.4, 0.43)$ with scaling coefficient $r_0 = 0.1$ and radius defined as $\rho(x) = \sqrt{\|x - x_p\|}/r_0$

$$\delta_p(x) = \varepsilon e^{-\frac{1}{2(1-\rho(x))^2}+\frac{1}{2}}, \tag{139}$$

until a final time $T = 0.35$, obtaining the solution $(u_p(x,t), p_p(x,t))$.

In Figure 8, we do not apply any preprocessing, but we just run the analytical perturbed solution as initial condition. We compare different meshes and orders to understand how the schemes behave. The plot for order 4 and 26×26 cells with SUPG-GFq shows the most accurate scheme we tested. For lower resolutions, we observe a clear advantage of the SUPG-GFq scheme over the SUPG only scheme, even if, it is easy to observe numerical noise also for SUPG-GFq in the slightly coarser mesh configurations.

In the other setups studied, we can reduce the size of the perturbation and still be able to see its evolution. In Figure 9, we use the integral procedure to preprocess the data and find the equilibrium, see Section 4.3, while in Figure 10, we use the optimization procedure as described above. Once the equilibrium is obtained, we add the perturbation δ_p and let the simulation run. We observe that the results, even with a much smaller perturbation, are very accurate for SUPG-GFq even for very coarse grids. To achieve comparable results, the standard SUPG scheme requires a very fine mesh and high resolution to capture the motion of the perturbation.

In Figure 11, we start from the optimal equilibrium obtained according to the procedure outlines in Definition 4.2, but we use the OSS stabilization technique. There are less precise

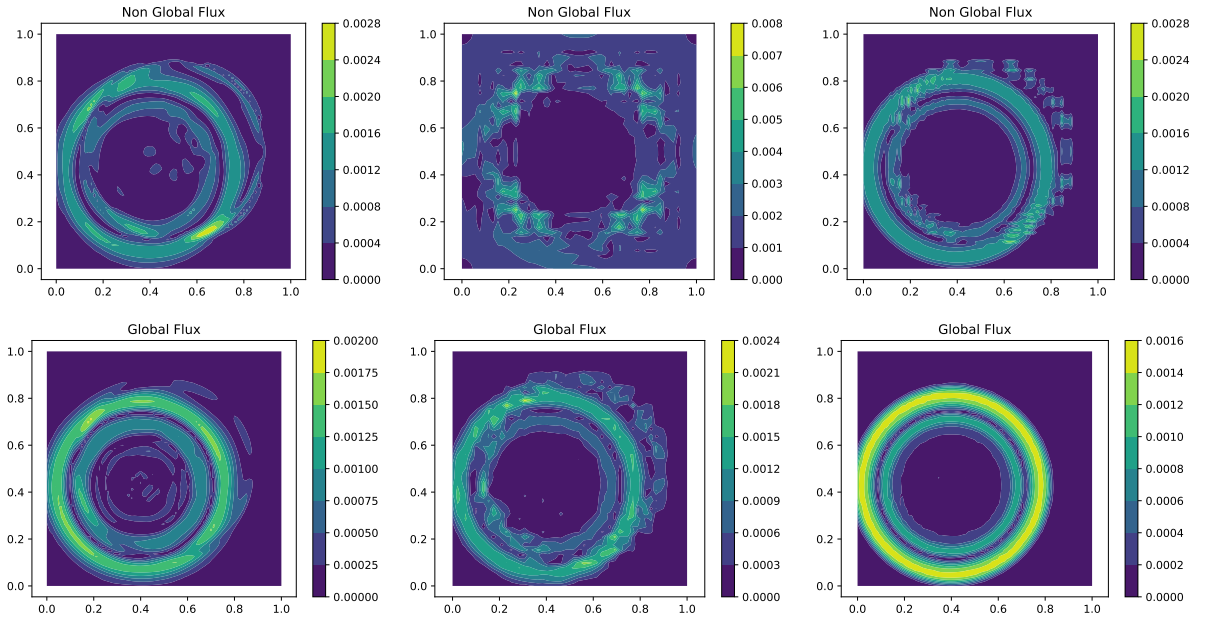

Figure 8: Perturbation($\varepsilon = 10^{-2}$) test: analytical solution. Plot of $\|\mathbf{u}_{eq} - \mathbf{u}_p\|$, with $\mathbf{u}_{eq}$ the analytical equilibrium (138). SUPG (top), SUPG-GFq (bottom). Left $\mathbb{P}^1$ with 80×80 cells, center $\mathbb{P}^3$ with 13 cells, right $\mathbb{P}^3$ with 26 cells

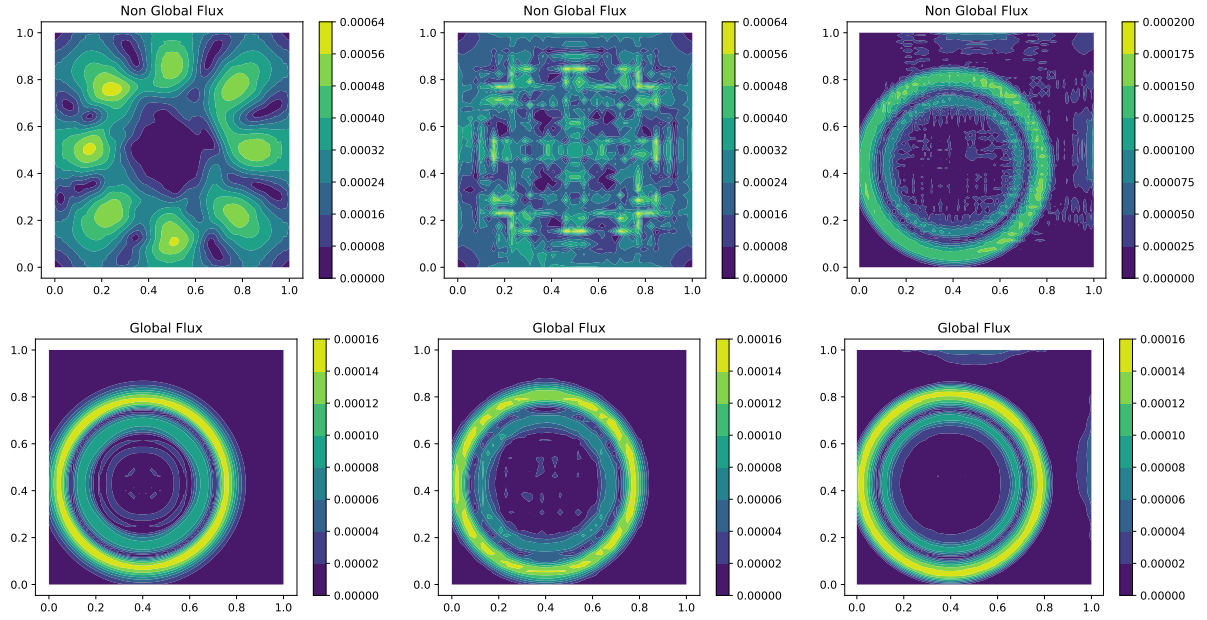

Figure 9: Perturbation($\varepsilon = 10^{-3}$) test: line-by-line equilibrium solution, see Section 4.3. Plot of $\|\mathbf{u}_{eq} - \mathbf{u}_p\|$, with $\mathbf{u}_{eq}$ the analytical equilibrium (138). SUPG (top), SUPG-GFq (bottom). Left $\mathbb{P}^1$ with 80×80 cells, center $\mathbb{P}^3$ with 13 cells, right $\mathbb{P}^3$ with 26 cells

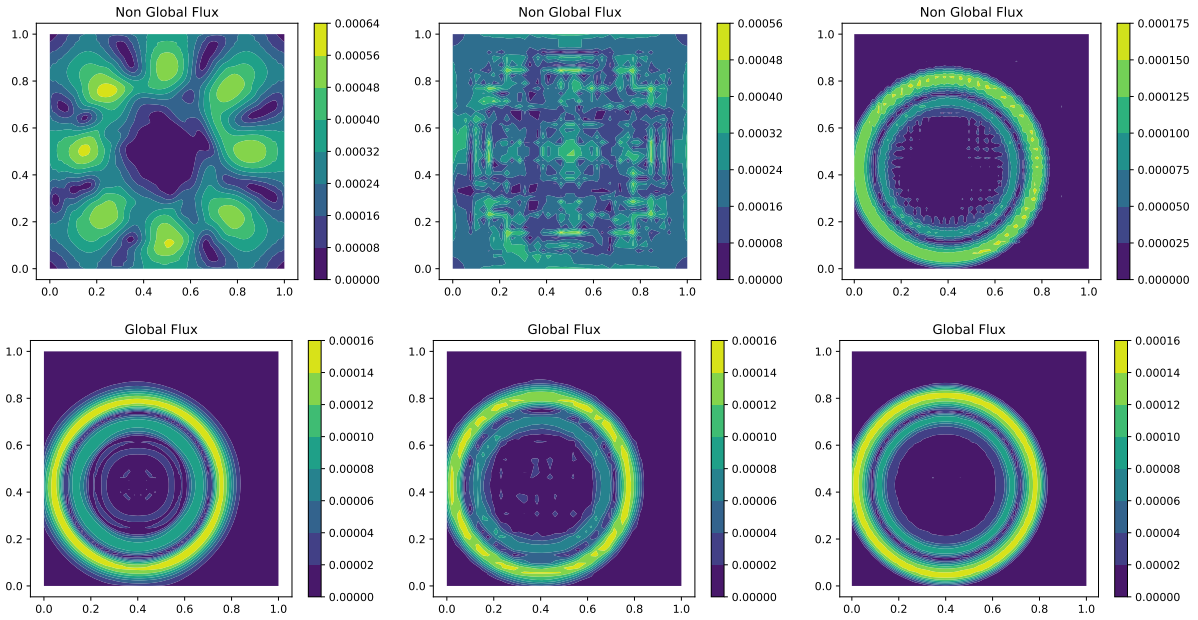

Figure 10: Perturbation($\varepsilon = 10^{-3}$) test: optimal equilibrium solution, see Section 4.3. Plot of $\|\mathbf{u}_{eq} - \mathbf{u}_p\|$, with $\mathbf{u}_{eq}$ the analytical equilibrium (138). SUPG (top), SUPG-GFq (bottom). Left $\mathbb{P}^1$ with 80×80 cells, center $\mathbb{P}^3$ with 13 cells, right $\mathbb{P}^3$ with 26 cells

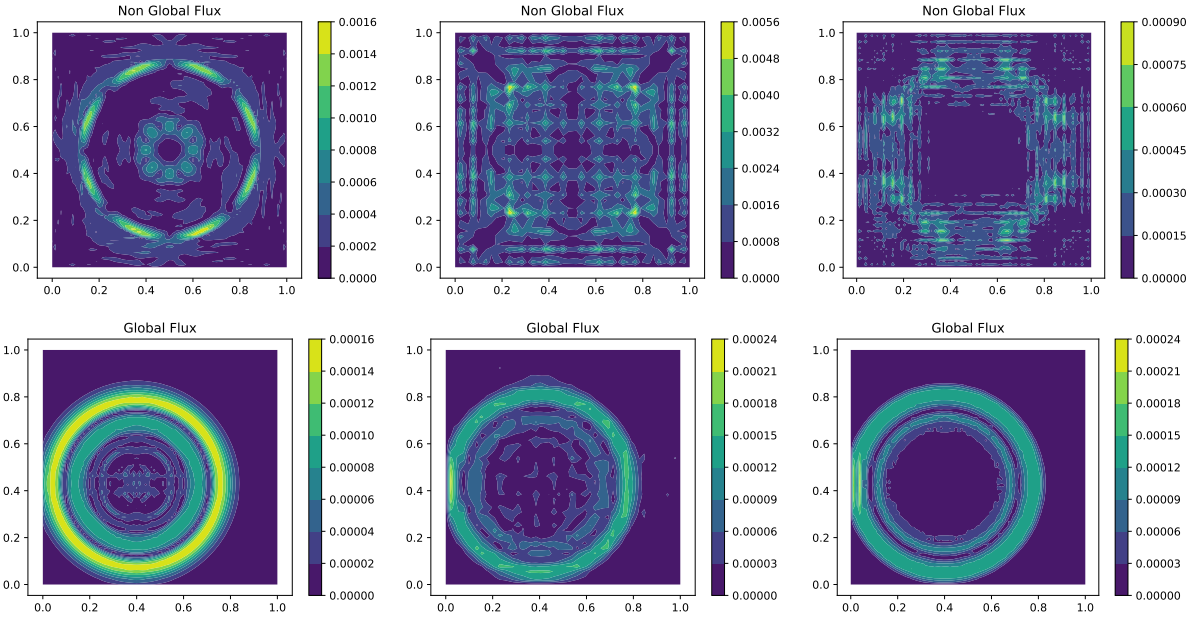

Figure 11: Perturbation($\varepsilon = 10^{-3}$) test: optimal equilibrium solution, see Section 4.3. Plot of $\|\mathbf{u}_{eq} - \mathbf{u}_p\|$, with $\mathbf{u}_{eq}$ the analytical equilibrium (138). OSS (top), OSS-GFq (bottom). Left $\mathbb{Q}^1$ with 80×80 cells, center $\mathbb{Q}^3$ with 13 cells, right $\mathbb{Q}^3$ with 26 cell

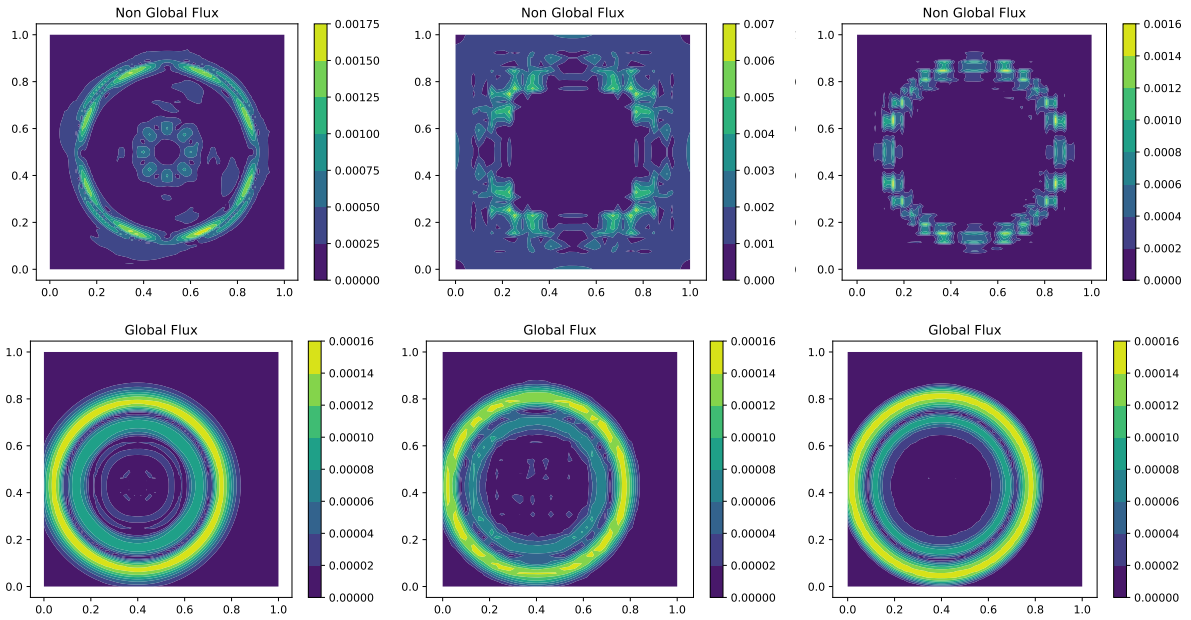

Figure 12: Perturbation test ($\varepsilon = 10^{-3}$): long time equilibrium solution. Plot of $\|u_{eq} - u_p\|$, with u_{eq} the analytical equilibrium (138). SUPG (top), SUPG-GFq (bottom). Left $\mathbb{Q}^1$ with 80×80 cells, center $\mathbb{Q}^3$ with 13 cells, right $\mathbb{Q}^3$ with 26 cells

results, with respect to SUPG, as boundaries in this case influence more the solutions, but the GF version is still very accurate for all the presented tests. We only use OSS in this test for brevity.

Finally, in Figure 12, we use as initial solution the long-time simulation of the previous test ($T = 100$) with the SUPG-GFq scheme that reaches convergence. We observe that adding a perturbation to such an initial condition leads to very clear results for the SUPG-GFq scheme.

7.4 Riemann Problem

We present in this section a two-dimensional Riemann problem (RP) centered in $x_0 = (0.5, 0.5)$ on the unit square $\Omega = [0, 1]^2$ [39]. The initial conditions are

$$u(x) = \begin{cases} 1, & \text{if } x > 0.5 \text{ and } y > 0.5, \\ 0, & \text{else,} \end{cases} \qquad v(x) = 0, \qquad p(x) = 0. \tag{140}$$

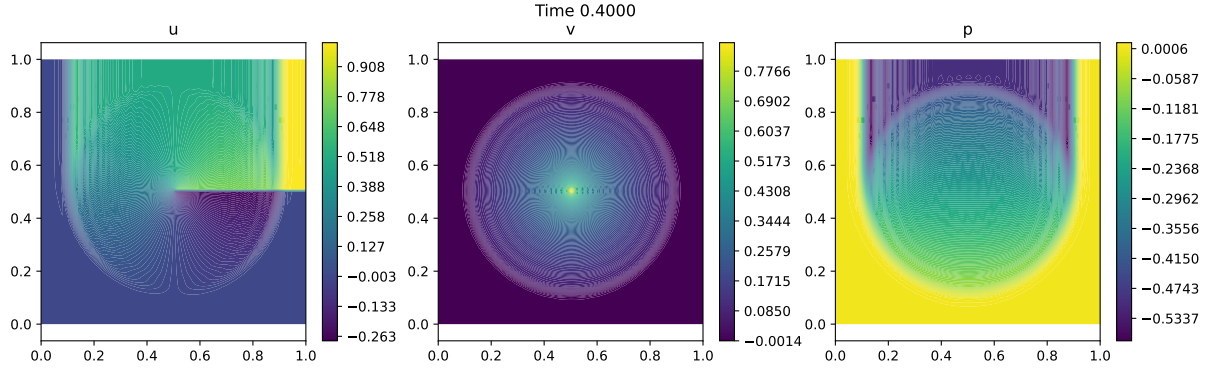

Figure 13: Riemann Problem. Simulation at time $T = 0.4$ with $\mathbb{P}^2$ elements and 50×50 cells with SUPG-GFq scheme

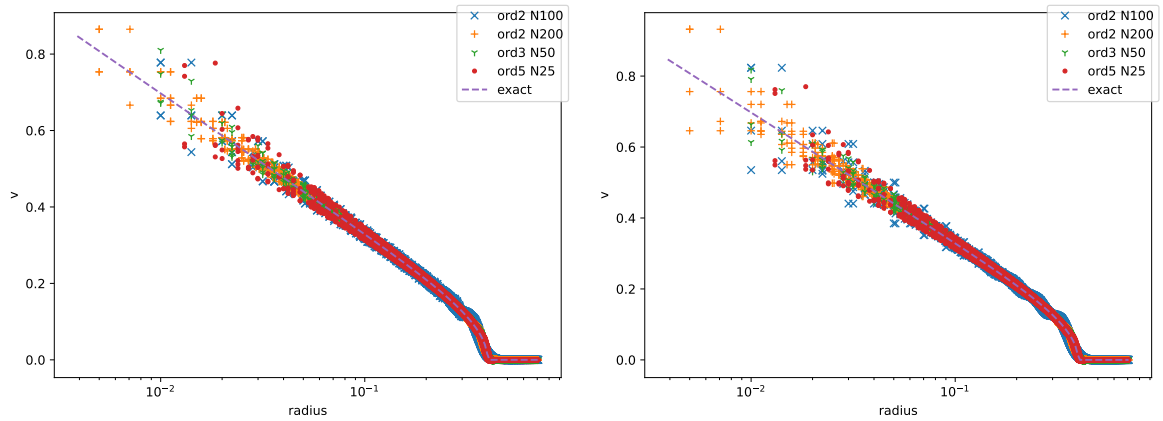

Figure 14: Riemann Problem. Distribution of the solution v for different elements and meshes. Left SUPG scheme, right SUPG-GFq scheme

It has been shown [39] that the perpendicular component v has a logarithmic singularity in the center of the RP for $t > 0$:

$$\begin{aligned} v(x,y,t) &= \frac{1}{2\pi}\mathcal{L}\left(\frac{\sqrt{(x-x_0)^2+(y-y_0)^2}}{ct}\right), \\ \mathcal{L}(s) &:= \log\left(\frac{1+\sqrt{1-s^2}}{s}\right) = -\log\left(\frac{s}{2}\right) - \frac{s^2}{4} + \mathcal{O}(s^4). \end{aligned} \tag{141}$$

In Figure 13, we plot the solution for $\mathbb{Q}^2$ elements with the SUPG-GFq method. Other resolutions and orders give qualitatively similar results. We can observe that some oscillations appear due to the Gibbs phenomenon at the discontinuities of the solution.

This can also be seen in Figure 14, where the solution v is plotted against the radius. We do not observe many differences between SUPG and SUPG-GFq.

8 Conclusions and perspectives

In this paper, we have studied the preservation of steady states for acoustics when using stabilized continuous finite elements. We have shown that, despite their structure, classical grad-div stabilizations as SUPG and OSS are in general not stationarity preserving due to an incompatibility between the stabilizing term and of the Galerkin one. Borrowing ideas from the so-called Global Flux quadrature, we have proposed a new framework allowing to construct constraint-compatible stabilization operators, which additionally are non-vanishing for at least some of the unwanted spurious modes. We have characterized rigorously the discrete kernel of the schemes obtained in terms of stability and consistency, and characterized the corresponding curl involutions using Fourier symbols. Numerical results confirm the theoretical developments. Ongoing work involves the extension to non-homogeneous systems, e.g. the linearized shallow water equations with Coriolis, friction, mass, and other sources, as well as the extension to discontinuous polynomial approximations. The application to non-linear systems (shallow water and Euler equations with gravity) is also under development.

Acknowledgments

D.T. was supported by the Ateneo Sapienza projects 2022 “Approssimazione numerica di modelli differenziali e applicazioni” and 2023 “Modeling, numerical treatment of hyperbolic equations

and optimal control problems". DT is member of the INdAM Research National Group of Scientific Computing (INdAM-GNCS). M.R. is a member of the Cardamom team, Inria at University of Bordeaux.

A Additional definitions and proofs

A.1 Proof of Theorem 2.1

The characteristic polynomials of A and B are

$$(\mathbb{F}_{t_x}(A))_{r,s} = \sum_{k\in\mathbb{Z}} \alpha^r_{k,s} t_x^k \qquad\qquad (\mathbb{F}_{t_x}(B))_{r,s} = \sum_{k\in\mathbb{Z}} \beta^r_{k,s} t_x^k \tag{142}$$

while that of AB is

$$(\mathbb{F}_{t_x}(AB))_{r,s'} = \sum_{k\in\mathbb{Z}}\sum_{s=1}^{K}\sum_{k'\in\mathbb{Z}} \alpha^r_{k,s}\beta^s_{k',s'} t_x^{k+k'} = \sum_{s=1}^{K}\sum_{k\in\mathbb{Z}} \alpha^r_{k,s} t_x^k \sum_{k'\in\mathbb{Z}} \beta^s_{k',s'} t_x^{k'} = \sum_{s=1}^{K} (\mathbb{F}_{t_x}(A))_{r,s}(\mathbb{F}_{t_x}(B))_{s,s'}. \tag{143}$$

A.2 Proof of Theorem 2.3

Simply inserting the definitions:

$$\begin{aligned}
((A^x \otimes A^y)(B^x \otimes B^y)q)^{rt}_{ij} &= \left(\sum_{(k,\ell)\in\mathbb{Z}^2}\sum_{s,p=1}^{K} (\alpha^x)^r_{k,s}(\alpha^y)^t_{\ell,p} q_{i+k,s;j+\ell,p}\right) \\
&\qquad \left(\sum_{(k,\ell)\in\mathbb{Z}^2}\sum_{s,p=1}^{K} (\beta^x)^{r,t}_{k,s;\ell,p}(\beta^y)^t_{\ell,p} q_{i+k,s;j+\ell,p}\right) \\
&= \sum_{(k,\ell)\in\mathbb{Z}^2}\sum_{(k',\ell')\in\mathbb{Z}^2}\sum_{s,p=1}^{K}\sum_{s',p'=1}^{K} (\alpha^x)^r_{k,s}(\alpha^y)^t_{\ell,p}(\beta^x)^r_{k',s'}(\beta^y)^t_{\ell',p'} q_{i+k+k',s';j+\ell+\ell',p'} \\
&= \sum_{k\in\mathbb{Z}}\sum_{k'\in\mathbb{Z}}\sum_{s=1}^{K}\sum_{s'=1}^{K} (\alpha^x)^r_{k,s}(\beta^x)^r_{k',s'} \sum_{\ell\in\mathbb{Z}}\sum_{\ell'\in\mathbb{Z}}\sum_{p=1}^{K}\sum_{p'=1}^{K} (\alpha^y)^t_{\ell,p}(\beta^y)^t_{\ell',p'} q_{i+k+k',s';j+\ell+\ell',p'} \\
&= ((A^xB^x)\otimes(A^yB^y)q)^{rt}_{ij}.
\end{aligned}$$

B Fully discrete T^2 SUPG operators

As an example, we explicitly list here the T^2 operator for the SUPG stabilization. We recall that indexes m, r are devoted to subtimenodes, indexes i, j are devoted to degrees of freedom, index d is for dimension and s, w for variable index. We start first with the classical SUPG

operator, corresponding to the matrix formulation (130):

$$T_u^{2,m}(\underline{q}) = M_x \otimes M_y \frac{u^m - u^0}{\Delta t} + D_x \otimes M_y \sum_r \vartheta_r^m p^r + \alpha h \left(D^x \otimes M_y \frac{p^m - p^0}{\Delta t} + D_x^x \otimes M_y \sum_r \vartheta_r^m u^r + D^x \otimes D_y \sum_r \vartheta_r^m v^r \right), \tag{144a}$$

$$T_v^{2,m}(\underline{q}) = M_x \otimes M_y \frac{v^m - v^0}{\Delta t} + M_x \otimes D_y \sum_r \vartheta_r^m p^r + \alpha h \left(M_x \otimes D^y \frac{p^m - p^0}{\Delta t} + D_x \otimes D^y \sum_r \vartheta_r^m u^r + M_x \otimes D_y^y \sum_r \vartheta_r^m v^r \right), \tag{144b}$$

$$T_p^{2,m}(\underline{q}) = M_x \otimes M_y \frac{p^m - p^0}{\Delta t} + D_x \otimes M_y \sum_r \vartheta_r^m u^r + M_x \otimes D_y \sum_r \vartheta_r^m v^r + \alpha h \left(D^x \otimes M_y \frac{u^m - u^0}{\Delta t} + M_x \otimes D^y \frac{v^m - v^0}{\Delta t} + (D_x^x \otimes M_y + M_x \otimes D_y^y) \sum_r \vartheta_r^m p^r \right). \tag{144c}$$

Now, we describe the SUPG-GFq T^2 operators, corresponding to the semidiscrete (95)

$$T_u^{2,m}(\underline{q}) = M_x \otimes M_y \frac{u^m - u^0}{\Delta t} + D_x \otimes M_y \sum_r \vartheta_r^m p^r + \alpha h \left(D^x \otimes M_y \frac{p^m - p^0}{\Delta t} + D_x^x \otimes D_y I_y \sum_r \vartheta_r^m u^r + D_x^x I_x \otimes D_y \sum_r \vartheta_r^m v^r \right), \tag{145a}$$

$$T_v^{2,m}(\underline{q}) = M_x \otimes M_y \frac{v^m - v^0}{\Delta t} + M_x \otimes D_y \sum_r \vartheta_r^m p^r + \alpha h \left(M_x \otimes D^y \frac{p^m - p^0}{\Delta t} + D_x \otimes D_y^y I_y \sum_r \vartheta_r^m u^r + D_x I_x \otimes D_y^y \sum_r \vartheta_r^m v^r \right), \tag{145b}$$

$$T_p^{2,m}(\underline{q}) = M_x \otimes M_y \frac{p^m - p^0}{\Delta t} + D_x \otimes D_y I_y \sum_r \vartheta_r^m u^r + D_x I_x \otimes D_y \sum_r \vartheta_r^m v^r + \alpha h \left(D^x \otimes M_y \frac{u^m - u^0}{\Delta t} + M_x \otimes D^y \frac{v^m - v^0}{\Delta t} + (D_x^x \otimes M + M \otimes D_y^y) \sum_r \vartheta_r^m p^r \right). \tag{145c}$$

C Curl involution for OSS: definitions

Here, we give implicitly the definition of $\mathcal{K}_u$ and $\mathcal{K}_v$.

$$(\mathbb{F}_{t_y}(D_y M_y^2)\mathbb{F}_{t_x}(M_x))\mathcal{K}_u = -\Big\{ \left(\mathbb{F}_{t_y}(D_y)^2 + \mathbb{F}_{t_y}(D_y^y M_y)\right) \Big(\mathbb{F}_{t_y}(D_y)^2 \mathbb{F}_{t_x}(M_x)^2 - \mathbb{F}_{t_y}(M_y) \left(-\mathbb{F}_{t_y}(D_y^y)\mathbb{F}_{t_x}(M_x)^2 - \mathbb{F}_{t_x}(D_x)^2 \mathbb{F}_{t_y}(M_y) - \mathbb{F}_{t_x}(D_x^x M_x)\mathbb{F}_{t_x,t_y}(M_y)\right) \Big) \Big\},$$

$$(\mathbb{F}_{t_x}(D_x M_x^2)\mathbb{F}_{t_y}(M_y))\mathcal{K}_v = \Big\{ \left(\mathbb{F}_{t_x}(D_x)^2 + \mathbb{F}_{t_x}(D_x^x M_x)\right) \Big(\mathbb{F}_{t_y}(D_y)^2 \mathbb{F}_{t_x}(M_x)^2 - \mathbb{F}_{t_y}(M_y) \Big(-\mathbb{F}_{t_y}(D_y^y)\mathbb{F}_{t_x}(M_x)^2 - \mathbb{F}_{t_x}(D_x)^2 \mathbb{F}_{t_y}(M_y) - \mathbb{F}_{t_x}(D_x^x M_x)\mathbb{F}_{t_y}(M_y) \Big) \Big) \Big\}.$$

D One–dimensional Kernel characterization

Lemma D.1 (Invertibility of mixed mass matrix)**.** *Consider $\{\hat{\varphi}_i\}_{i=0}^K$ the set of Lagrangian polynomials generated by $K+1$ Gauss Lobatto points in $\mathbb{P}^K([0,1])$ and $\{\hat{\psi}_i\}_{i=0}^{K-1}$ the set of Lagrangian polynomials generated by K Gauss Lobatto points in $\mathbb{P}^{K-1}([0,1])$. Consider the matrix $A_{ij} := \int_0^1 \hat{\varphi}_i\hat{\psi}_j \mathrm{d}x$ for $i = 0,\dots,K$ and $j = 0,\dots,K-1$. Now, consider the square matrices obtained as the restriction of A that we denote by $B_{ij} = A_{ij}$ for $i = 0,\dots,K-1$, $j = 0,\dots,K-1$. Then, B is invertible.*

Proof. Let us denote with (x_i, w_i) for $i = 0,\dots,K$ the $K+1$ Gauss–Lobatto quadrature nodes and weights, respectively. This quadrature formula is exact for polynomials of degree $2K-1$, hence,

$$B_{ij} = \int_0^1 \hat{\varphi}_i(x)\hat{\psi}_j(x)\mathrm{d}x = w_i\hat{\psi}_j(x_i). \tag{146}$$

We want to show that the system $\sum_{j=1}^{K-1} B_{ij}q_j = r_i$ for $i = 1,\dots,K-1$ is invertible. Using (146), the system becomes

$$\sum_{j=1}^{K-1} B_{ij}q_j = r_i \iff \sum_{j=1}^{K-1} \hat{\psi}_j(x_i)q_j = \frac{r_i}{w_i}. \tag{147}$$

Now, the system (147) for q_j with $j = 0,\dots,K-1$ is equivalent to finding a polynomial $q_h \in \mathbb{P}^{K-1}$ that interpolates the K distinct point $\{(0,0)\}\cup\{(x_i, r_i/w_i)\}_{i=1}^{K-1}$, which has one and only one solution. Hence, the matrix B is invertible. □

Proposition D.1 (Kernel characterization of $\tilde{D}_x$)**.** *$\tilde{D}_x : \mathbb{R}^{N_x\times K} \to \mathbb{R}^{N_x\times K-1}$ has kernel of dimension one and it is generated by a function that is discontinuous at each cell interface, hence, not the constant function.*

Proof. Let us quickly recall the definition and notation of the matrix. Let us order the indexes of the degrees of freedom of $V^K_{\Delta x}(\Omega^x_{\Delta x})$ with a unique index. Recall that for $i = 0,\dots,N_x - 1$ $\varphi_{i,k}|_{E_i^x} \in \mathbb{P}^K(E_i^x)$ are the Lagrangian basis functions defined through the Gauss–Lobatto quadrature points. We also recall that for the degrees of freedom $r = 0,\dots,K$ of the cell E_i^x we assign a unique index the $\alpha := iK + r$ with the equivalence $\varphi_{i,K} = \varphi_{i+1,0}$ for $i = 0,\dots,N_x-2$. For $V^K_{\Delta x,0}(\Omega^x_{\Delta x})$ the first and the last degrees of freedom are neglected, and we can define its basis as $\{\varphi_\alpha\}_{\alpha=1}^{N_xK-1}$. For the broken polynomial space $V^{K-1}_{\Delta x,b}(\Omega^x_{\Delta x})$, instead, for each Lagrangian basis function $\psi_{j,\ell}|_{E_j^x} \in \mathbb{P}^{K-1}(E_j^x)$ for $j = 0,\dots,N_x-1$ and $k = 1,\dots,K$, we define another unique index $\beta = jK + k$ with $\beta = 1,\dots,N_xK$.

Then, the matrix is defined for $\alpha = 1,\dots,N_xK-1$ and $\beta = 1,\dots,N_xK$ as

$$(\tilde{D}_x)_{\alpha;\beta} := \int_{\Omega^x_{\Delta x}} \varphi_\alpha(x)\psi_\beta(x)\mathrm{d}x. \tag{148}$$

Now, we want to compute the kernel of these operators, so we are solving for each of them N_xK-1 equations for $\alpha = 1,\dots,N_xK-1$

$$\sum_{\beta=1}^{N_xK} (\tilde{D}_x)_{\alpha,\beta}q_\beta = 0. \tag{149}$$

In this first part of the proof, we show that any basis the kernel of $\tilde{D}_x$ must have discontinuities across cells if $q_{i,0} \neq 0$ for all $i = 1,\dots,N_x-1$. Later, we will show that $\tilde{D}_x$ has full rank N_x-1 and hence that basis generates the whole kernel.

Each term of the matrices can be computed as sum of integrals of polynomials of degree $2K-1$ at most, hence, it will be exactly computed by the Gauss–Lobatto quadrature formula

with K nodes which defines also the polynomials of degree K. For the matrix $\tilde{D}_x$, if we focus on the degrees of freedom $\alpha = (i, 0)$ for $i = 1, \dots, N_x - 1$, and using the Lobatto quadrature formula with weights $\{w_r\}_{r=0}^K$, we obtain that (149) becomes: find $q_h \in V_{\Delta x,b}^{K-1}(\Omega_{\Delta x}^x)$ such that

$$0 = \int_{E_{x,i-1}} \varphi_{i-1,K}(x) q_h(x) \mathrm{d}x + \int_{E_{x,i}} \varphi_{i,0}(x) q_h(x) \mathrm{d}x = \Delta x \, w_0 (q_{i-1,K-1} + q_{i,0}). \tag{150}$$

Here, clearly we have that $q_{i,0} = -q_{i-1,K-1}$ for $i = 1, \dots, N_x - 1$. Hence, the constant 1 cannot be in the kernel of $\tilde{D}_x$.

Now, to show the matrix is full ranked, we study its structure in (151), recalling that $\tilde{D}_x \in \mathbb{R}^{(N_x K-1)\times(N_x K)}$.

$$\begin{array}{r}
K-1 \left\{ \right. \\
K \left\{ \right. \\
K \left\{ \right.
\end{array}
\left[\begin{array}{ccc|ccc|ccc}
 & Z & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
* & * & * & & & & 0 & 0 & 0 \\
0 & 0 & 0 & & B & & 0 & 0 & 0 \\
0 & 0 & 0 & & & & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & * & * & * & & & \\
0 & 0 & 0 & 0 & 0 & 0 & & B & \\
0 & 0 & 0 & 0 & 0 & 0 & & &
\end{array}\right] \tag{151}$$

We observe that it contains matrices $B \in \mathbb{R}^{(K-1)\times(K-1)}$ proportional to the B matrix of Lemma D.1 and a $Z \in \mathbb{R}^{K\times(K-1)}$ matrix that is given by B without the first row. Since B is invertible, there exists a minor of Z of dimension $(K-1)\times(K-1)$ that is invertible as well. If we remove the corresponding column also from the matrix $\tilde{D}_x$, the resulting is also invertible. This can be seen by computing the determinant and looking at the minors that contribute to that computation, i.e., only the B matrices and of the invertible minor of Z. This means that $\tilde{D}_x$ is of full rank $N_x K - 1$ and, hence, possesses a kernel of dimension one generated by a nonconstant vector. □

Lemma D.2 (Invertibility of mixed derivative matrix)**.** *Consider $\{\hat{\varphi}_i\}_{i=0}^K$ the set of Lagrangian polynomials generated by $K+1$ Gauss Lobatto points in $\mathbb{P}^K([0,1])$ and $\{\hat{\psi}_i\}_{i=0}^{K-1}$ the set of Lagrangian polynomials generated by K Gauss Lobatto points in $\mathbb{P}^{K-1}([0,1])$. Consider the matrix $C_{ij} := \int_0^1 \partial_x \hat{\varphi}_i \hat{\psi}_j \mathrm{d}x$ for $i = 0, \dots, K$ and $j = 0, \dots, K-1$. Now, consider the square matrices obtained as the restriction of C that we denote by $D_{ij} = C_{ij}$ for $i = 1, \dots, K$, $j = 0, \dots, K-1$. Then, D is invertible.*

Proof. Instead of proving the invertibility of D, we will show the invertibility of its transpose, which is

$$(D^T)_{ij} = \int_0^1 \hat{\psi}_i(x) \partial_x \hat{\varphi}_j(x) \mathrm{d}x. \tag{152}$$

Now, the system is invertible if there exists one and only solution to the system of equations for the unknown $\{q_j\}_{j=1}^K$ and the right hand side $\{r_i\}_{i=0}^{K-1}$

$$\int_0^1 \hat{\psi}_i(x) \partial_x \hat{\varphi}_j(x) q_j \mathrm{d}x = r_i \iff \int_0^1 \hat{\psi}_i(x) q_h(x) \mathrm{d}x = r_i, \tag{153}$$

with $q_h = \sum_{j=1}^K \partial_x \hat{\varphi}_j(x) \in \mathbb{P}^{K-1}([0,1])$ for every $\hat{\psi}_i \in \mathbb{P}^{K-1}$. Classically, $q_h \in \mathbb{P}^{K-1}([0,1])$ could be rewritten in an expansion of the $\hat{\psi}_j$ basis functions, obtaining on the left hand side the classical symmetric and positive definite mass matrix for $\mathbb{P}^{K-1}([0,1])$. This is invertible, hence D is invertible. □

Proposition D.2 (Kernel characterization of $\tilde{D}_x^x$)**.** *$\tilde{D}_x^x : \mathbb{R}^{N_x\times K} \to \mathbb{R}^{N_x\times K-1}$ has kernel of dimension one and it is generated by the constant function* 1*.*

Proof. The matrix is defined for $\alpha = 1, \dots, N_x K - 1$ and $\beta = 1, \dots, N_x K$ (with the notation of Theorem 5.3) by

$$(\tilde{D}_x^x)_{\alpha;\beta} := \int_{\Omega_{\Delta x}^x} \partial_x \varphi_\alpha(x) \psi_\beta(x) \mathrm{d}x. \tag{154}$$

Clearly 1 belongs to the kernel of $\tilde{D}_x^x$, as

$$\int_{\Omega_{\Delta x}^x} \partial_x \varphi_\alpha(x) 1 \mathrm{d}x = [\varphi_\alpha(x) \partial_x 1]_{\partial \Omega_{\Delta x}^x} - \int_{\Omega_{\Delta x}^x} \varphi_\alpha(x) \partial_x 1 \mathrm{d}x = 0 \tag{155}$$

because $\varphi_\alpha \in V_{\Delta x,0}^K(\Omega_{\Delta x}^x)$ and $\partial_x 1 \equiv 0$. This corresponds to the space of all affine functions for the kernel of D_x^x.

$$\begin{array}{r} K \left\{ \right. \\ K \left\{ \right. \\ K-1 \left\{ \right. \end{array} \left[\begin{array}{ccc|ccc|ccc} & & & 0 & 0 & 0 & 0 & 0 & 0 \\ & D & & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & * & * & * & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & & & & 0 & 0 & 0 \\ 0 & 0 & 0 & & D & & 0 & 0 & 0 \\ 0 & 0 & 0 & & & & * & * & * \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & & E & \end{array} \right] \tag{156}$$

Now, looking at the structure of $\tilde{D}_x^x \in \mathbb{R}^{(N_x K-1)\times(N_x K)}$, we observe that the matrix $D \in \mathbb{R}^{K\times K}$ is proportional to the one of Lemma D.2. Matrix E is the matrix D without its last row. Since D is invertible, there exists a minor of E of dimension $(K-1)\times(K-1)$ that is invertible. Hence, if we consider the whole matrix without the column which is excluded by the invertible minor of E, then, we clearly see that it is invertible (the determinant is not 0 studying the relevant minors). Hence, the kernel of D_x^x is of dimension 1 and it is generated by the constant function 1. □

Proposition D.3 (Kernel characterization of $Z_x = D_x^x - D^x M^{-1} D_x$)**.** *Let $Z_x = D_x^x - D^x M^{-1} D_x : \mathbb{R}^{N_x\times K+1} \to \mathbb{R}^{N_x\times K-1}$, with $M^{-1}, D_x : \mathbb{R}^{N_x\times K+1} \to \mathbb{R}^{N_x\times K+1}$ be the matrices defined with test and trial functions in $V_{\Delta x}^K(\Omega_{\Delta x}^x)$ and $D_x^x, D^x : \mathbb{R}^{N_x\times K+1} \to \mathbb{R}^{N_x\times K-1}$ be defined with trial functions in $V_{\Delta x}^K(\Omega_{\Delta x}^x)$ and test in $V_{\Delta x,0}^K(\Omega_{\Delta x}^x)$. Then, the kernel of Z contains $\langle 1, x\rangle$ and does not contain w the vector of the kernel of D_x.*

Proof. Clearly, 1 and x belong to the kernel of ${}_xZ$ as they belong to the kernel of D_x^x and since $D_x 1 \equiv 0$ and

$$D^x M^{-1} D_x x \equiv D^x M^{-1} 1 \equiv D^x 1 \equiv 0.$$

Let us now restrict our operator on V_0^K, by taking the affine operation $\tilde{u} = u - u(0) + (u(0) - u(1))x$ that uses only elements in the kernel of Z_x to bring $u \in V_{\Delta x}^K(\Omega_{\Delta x}^x)$ into $\tilde{u} \in V_{\Delta x,0}^K(\Omega_{\Delta x}^x)$. We aim showing that Z_x on $V_{\Delta x}^K(\Omega_{\Delta x}^x)$ is symmetric non-negative definite. The symmetry is trivially shown. We focus on the non-negativeness.

Before proceeding, let us better describe Z_x on $V_{\Delta x}^K(\Omega_{\Delta x}^x)$. We have that

$$(D_x)_{\alpha;\beta} u^\beta = \begin{cases} \Delta x w_\alpha \left(\partial_x u(x_\alpha^+) + \partial_x u(x_\alpha^-)\right) & \text{if } \alpha = (i,0) = (i-1,K), \\ \Delta x w_\alpha \partial_x u(x_\alpha) & \text{else,} \end{cases} \tag{157}$$

with $\partial_x u \in V_{\Delta x,b}^{K-1}(\Omega_{\Delta x}^x)$, x_α being the Gauss–Lobatto composite quadrature points defined by $K+1$ points in each cell and w_α the quadrature weight referred to the k-th degree of freedom in the reference element $[0,1]$ with $w_{(i,0)} = w_{(i-1,K)}$. Then, if we study the bilinear form, we have

that

$$u^T(D^xM^{-1}D_x)u = \sum_{\alpha,\beta}\int \partial_x u\,\varphi_\alpha\,dx\,\frac{\delta_{\alpha,\beta}}{M_{\alpha,\alpha}}\int \varphi_\beta\,\partial_x u\,dx = \sum_\alpha \frac{1}{M_{\alpha,\alpha}}\left(\int \varphi_\alpha \partial_x u\right)^2 = \tag{158}$$

$$\sum_{\alpha\in\mathcal{I}}\frac{\Delta x^2 w_\alpha^2}{\Delta x w_\alpha}(\partial_x u(x_\alpha))^2 + \sum_{\alpha\in\mathcal{E}}\frac{\Delta x^2 w_\alpha^2}{2\Delta x w_\alpha}(\partial_x u(x_\alpha^-) + \partial_x u(x_\alpha^+))^2 = \tag{159}$$

$$\sum_{\alpha\in\mathcal{I}}\Delta x w_\alpha(\partial_x u(x_\alpha))^2 + \sum_{\alpha\in\mathcal{E}}\frac{\Delta x w_\alpha}{2}(\partial_x u(x_\alpha^-) + \partial_x u(x_\alpha^+))^2, \tag{160}$$

where we have introduced the set of internal degrees of freedom $\mathcal{I} = \{\alpha : \varphi_\alpha \in V^K_{\Delta x,0}(\Omega^x_{\Delta x}),\ \alpha = (i,k) \text{ with } k \in [1, K-1]\}$ and edges degrees of freedom $\mathcal{E} = \{\alpha : \varphi_\alpha \in V^K_{\Delta x,0}(\Omega^x_{\Delta x}),\ \alpha = (i,0)\}$.

Now, using this definition, we will show that the restriction of Z_x to $V^K_{\Delta x,0}(\Omega^x_{\Delta x})$ is symmetric non-negative definite. Take $u \in V^K_{\Delta x,0}(\Omega^x_{\Delta x}) \equiv \mathbb{R}^{N_x K-1} \subset V^K_{\Delta x}(\Omega^x_{\Delta x}) \equiv \mathbb{R}^{N_x K+1}$, using the previous computations and the definition of D^x_x, we compute

$$u^T Z_x u = u^T D^x_x u - u^T D^x M^{-1} D_x u = \int (\partial_x u)^2 - u^T D^x M^{-1} D_x u = \tag{161}$$

$$\begin{aligned}&\sum_{\alpha\in\mathcal{I}}\Delta x w_\alpha(\partial_x u(x_\alpha))^2 + \sum_{\alpha\in\mathcal{E}}\Delta x w_\alpha(\partial_x u(x_\alpha^-)^2 + \partial_x u(x_\alpha^+)^2)\\ -&\sum_{\alpha\in\mathcal{I}}\Delta x w_\alpha(\partial_x u(x_\alpha))^2 - \sum_{\alpha\in\mathcal{E}}\frac{\Delta x w_\alpha}{2}(\partial_x u(x_\alpha^-) + \partial_x u(x_\alpha^+))^2 =\end{aligned} \tag{162}$$

$$\sum_{\alpha\in\mathcal{E}}\frac{\Delta x w_\alpha}{2}(\partial_x u(x_\alpha^-) - \partial_x u(x_\alpha^+))^2 \geq 0. \tag{163}$$

We have just shown that Z_x is non-negative definite. So, the element in the kernel of Z_x, i.e., $Z_x u = 0$, must also be such that $u^T Z_x u = 0$ and hence, they must have continuous derivative at the interfaces (163). This was not the case for w the element generating with 1 the kernel of D_x. □

Unfortunately, we cannot say more about the matrix Z_x and, experimentally, we have noticed that the kernel is indeed much larger than just these vectors. In particular, the dimension of the kernel increases with the order of the method and the number of cells.

References

[1] Wasilij Barsukow. Stationarity preserving schemes for multi-dimensional linear systems. *Math.Comp.*, 88(318):1621–1645, 2019.

[2] Wasilij Barsukow and Christian Klingenberg. Exact solution and a truly multidimensional Godunov scheme for the acoustic equations. *ESAIM: M2AN*, 56(1), 2022.

[3] Jonathan Jung and Vincent Perrier. Steady low Mach number flows: identification of the spurious mode and filtering method. *J. Comput. Phys.*, 468:111462, 2022.

[4] Jonathan Jung and Vincent Perrier. Behavior of the discontinuous Galerkin method for compressible flows at low Mach number on triangles and tetrahedrons. *SIAM J. Sci. Comput.*, 46(1):A452–A482, 2024.

[5] Keith William Morton and Philip L Roe. Vorticity-preserving Lax-Wendroff-type schemes for the system wave equation. *SIAM J.Sci.Comp.*, 23(1):170–192, 2001.

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

[7] Rolf Jeltsch and Manuel Torrilhon. On curl-preserving finite volume discretizations for shallow water equations. *BIT Numerical Mathematics*, 46(1):35–53, 2006.

[8] Siddhartha Mishra and Eitan Tadmor. Constraint preserving schemes using potential-based fluxes II. Genuinely multi-dimensional central schemes for systems of conservation laws. *ETH preprint*, (2009-32), 2009.

[9] TB Lung and PL Roe. Toward a reduction of mesh imprinting. *Int.J.Numer.Meth.Fl.*, 76(7):450–470, 2014.

[10] Wasilij Barsukow. Truly multi-dimensional all-speed schemes for the Euler equations on Cartesian grids. *J.Comput.Phys.*, 435:110216, 2021.

[11] Wasilij Barsukow, Raphael Loubere, and Pierre-Henri Maire. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *submitted to Math.Comp.*, 2023.

[12] Martin Campos Pinto and Eric Sonnendrücker. Gauss-compatible Galerkin schemes for time-dependent Maxwell equations. *Math.Comp.*, 85(302):2651–2685, 2016.

[13] Golo A Wimmer, Colin J Cotter, and Werner Bauer. Energy conserving upwinded compatible finite element schemes for the rotating shallow water equations. *J.Comput.Phys.*, 401:109016, 2020.

[14] Douglas N Arnold, Richard S Falk, and Ragnar Winther. Finite element exterior calculus, homological techniques, and applications. *Acta numerica*, 15:1–155, 2006.

[15] Alexander N Brooks and Thomas JR Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput.Meth.Appl.Mech.Eng.*, 32(1-3):199–259, 1982.

[16] Thomas JR Hughes and Michel Mallet. A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems. *Comput.Meth.Appl.Mech.Eng.*, 58(3):305–328, 1986.

[17] Ramon Codina and Jordi Blasco. A finite element formulation for the Stokes problem allowing equal velocity-pressure interpolation. *Comput.Meth.Appl.Mech.Eng.*, 143(3):373–391, 1997.

[18] Ramon Codina. Stabilization of incompressibility and convection through orthogonal subscales in finite element methods. *Comput.Meth.Appl.Mech.Eng.*, 190(13):1579–1599, 2000.

[19] Santiago Badia and Ramon Codina. Unified Stabilized Finite Element Formulations for the Stokes and the Darcy Problems. *SIAM J.Numer.An.*, 47, 01 2009.

[20] Sixtine Michel, Davide Torlo, Mario Ricchiuto, and Rémi Abgrall. Spectral analysis of high order continuous FEM for hyperbolic PDEs on triangular meshes: influence of approximation, stabilization, and time-stepping. *J.Sci.Comp.*, 94:49, 2023.

[21] Ll. Gascón and J.M. Corberán. Construction of second-order TVD schemes for nonhomogeneous hyperbolic conservation laws. *J.Comput.Phys.*, 172(1):261–297, 2001.

[22] R. Donat and A. Martinez-Gavara. Hybrid second order schemes for scalar balance laws. *J.Sci.Comp.*, 48(1):52–69, 2011.

[23] Yogiraj Mantri, Philipp Öffner, and Mario Ricchiuto. Fully well-balanced entropy controlled discontinuous Galerkin spectral element method for shallow water flows: Global flux quadrature and cell entropy correction. *J.Comput.Phys.*, page 112673, 2023.

[24] Manuel J. Castro and Carlos Parés. Well-balanced high-order finite volume methods for systems of balance laws. *J.Sci.Comp.*, 82(2):48, 2020.

[25] Irene Gómez-Bueno, Manuel Jesús Castro Díaz, Carlos Parés, and Giovanni Russo. Collocation methods for high-order well-balanced methods for systems of balance laws. *Mathematics*, 9(15), 2021.

[26] A. Prothero and A. Robinson. On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations. *Math.Comp.*, 28:145–162, 1974.

[27] E. Hairer, G. Wanner, and S.P. Norset. *Solving Ordinary Differential Equations I. Nonstiff problems.* Springer, Berlin, Heidelberg, 1993.

[28] Erik Burman. Consistent SUPG-method for transient transport problems: Stability and convergence. *Comput.Meth.Appl.Mech.Eng.*, 199:1114–1123, 03 2010.

[29] Sixtine Michel, Davide Torlo, Mario Ricchiuto, and Rémi Abgrall. Spectral analysis of continuous FEM for hyperbolic PDEs: influence of approximation, stabilization, and time-stepping. *J.Sci.Comp.*, 89(2):1–41, 2021.

[30] Wasilij Barsukow. *Low Mach number finite volume methods for the acoustic and Euler equations.* Doctoral thesis, University of Wuerzburg, 2018.

[31] Leslie Fox and ET Goodwin. Some new methods for the numerical integration of ordinary differential equations. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 45, pages 373–388. Cambridge University Press, 1949.

[32] Alok Dutt, Leslie Greengard, and Vladimir Rokhlin. Spectral deferred correction methods for ordinary differential equations. *BIT*, 40(2):241–266, 2000.

[33] Michael L Minion. Semi-implicit spectral deferred correction methods for ordinary differential equations. *Comm.Math.Sci.*, 1(3):471–500, 2003.

[34] Rémi Abgrall. High order schemes for hyperbolic problems using globally continuous approximation and avoiding mass matrices. *J.Sci.Comp.*, 73:461–494, 2017.

[35] Lorenzo Micalizzi and Davide Torlo. A new efficient explicit Deferred Correction framework: analysis and applications to hyperbolic PDEs and adaptivity. *Comm.Appl.Math.Comp.*, 2023.

[36] Maria Han Veiga, Philipp Öffner, and Davide Torlo. Dec and Ader: similarities, differences and a unified framework. *J.Sci.Comp.*, 87(1):1–35, 2021.

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

[38] Mirco Ciallella, Davide Torlo, and Mario Ricchiuto. Arbitrary High Order WENO Finite Volume Scheme with Flux Globalization for Moving Equilibria Preservation. *J.Sci.Comp.*, 96:53, 2023.

[39] Wasilij Barsukow and Christian Klingenberg. Exact solution and the multidimensional Godunov scheme for the acoustic equations. *ESAIM: M2AN*, 56(1):317–347, 2022.