Spectral analysis of high order continuous FEM for hyperbolic PDEs on triangular meshes: influence of approximation, stabilization, and time-stepping

Sixtine Michel, Davide Torlo, Mario Ricchiuto, Rémi Abgrall

Spectral analysis of high order continuous FEM for hyperbolic PDEs on triangular meshes: influence of approximation, stabilization, and time-stepping

Sixtine Michel*, Davide Torlo†, Mario Ricchiuto‡, Rémi Abgrall§

June 14, 2022

Abstract

In this work we study various continuous finite element discretization for two dimensional hyperbolic partial differential equations, varying the polynomial space (Lagrangian on equispaced, Lagrangian on quadrature points (*Cubature*) and Bernstein), the stabilization techniques (streamline-upwind Petrov-Galerkin, continuous interior penalty, orthogonal subscale stabilization) and the time discretization (Runge-Kutta (RK), strong stability preserving RK and deferred correction). This is an extension of the one dimensional study by Michel S. et al *J. Sci. Comput.* (2021), whose results do not hold in multi-dimensional frameworks. The study ranks these schemes based on efficiency (most of them are mass-matrix free), stability and dispersion error, providing the best CFL and stabilization coefficients. The challenges in two-dimensions are related to the Fourier analysis. Here, we perform it on two types of periodic triangular meshes varying the angle of the advection, and we combine all the results for a general stability analysis. Furthermore, we introduce additional high order viscosity to stabilize the discontinuities, in order to show how to use these methods for tests of practical interest. All the theoretical results are thoroughly validated numerically both on linear and non-linear problems, and error-CPU time curves are provided. Our final conclusions suggest that *Cubature* elements combined with SSPRK and OSS stabilization is the most promising combination.

Keywords: Continuous finite elements, dispersion analysis, stabilization techniques, high order accuracy, non-standard elements, mass lumping.

1 Introduction

We study several continuous finite element formulations to approximate the solution of the two dimensional hyperbolic conservation laws

$$\partial_t u(x,t) + \nabla \cdot f(u(x,t)) = 0 \quad x \in \Omega \subset \mathbb{R},\, t \in \mathbb{R}^+, \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ is the domain, $f : \mathbb{R}^D \to \mathbb{R}^{2\times D}$ is the flux function and $u : \Omega \to \mathbb{R}^D$ is the unknown of the system of equations.

The largest part of the paper is dedicated to the two-dimensional spectral analysis of different stabilized approaches applied to the scalar ($D = 1$) transport equations obtained for

$$f(u(x,t)) = \mathbf{a}u(x,t) \qquad \mathbf{a} \in \mathbb{R}^2. \tag{2}$$

One of the main objectives of this paper is to identify strategies to build (linearly) stable fully explicit high order continuous finite element schemes to discretize (1) on triangulations of the spatial domain Ω. To this end we will vary

*CEA CESTA, 15 av. des Sablières, 33114 Le Barp, France.
†Mathematics Area, SISSA Mathlab, SISSA, via Bonomea, 265, 34136 Trieste, Italy.
‡Team CARDAMOM, Inria Bordeaux sud-Ouest, 200 av. de la vieille tour, 33405 Talence, France.
§Institut für Mathematik, Winterthurstrasse 190, CH 8057 Zürich, Switzerland.

the basis functions, the stabilization technique and the time discretization. In general, the standard Finite Element Method (FEM) derived by this approach require the inversion of a large sparse mass matrix. This procedure can be expensive as either inverting the mass matrix and, hence, the matrix multiplications must be repeated for every time step or the linear solver must be applied at each time step. Various techniques have been introduced to overcome the mass matrix inversion while keeping the high order accuracy of the scheme.

The first strategy we study is the one proposed in [1]. In the reference it is suggested to combine mass lumping with a deferred correction (DeC) iterative time integration method allowing to introduce appropriate corrections in the right–hand side in order to recover the original order of accuracy. This approach can only be used in combination with finite elements whose basis functions have positive integrals. Another approach is based on a careful choice of approximation points defining sufficiently accurate quadrature formulas with all positive weights. If the variational form is evaluated with this underlying quadrature, as in spectral element methods, we obtain a diagonal mass matrix without loosing the order of accuracy. We refer to this case as *cubature* elements [40]. For this choice, the classical use of Runge–Kutta methods will provide the high order accuracy also for the time discretization.

Secondly, we will study the influence of the stabilization strategy. When solving (1) with continuous finite elements some additional stabilization operator is necessary to enforce the $\mathbb{L}_2$ stability. Several stabilization techniques can be devised to introduce a level of dissipation comparable to that of discontinuous Galerkin methods with upwind fluxes [46, 47]. Three approaches will be studied: the streamline upwind Petrov–Galerkin (SUPG) stabilization [18, 12], which is strongly consistent, but it is also introducing new terms in the mass matrix; thecontinuous interior penalty (CIP) method [16, 19, 14], consisting in adding edge penalty terms proportional to the jump of the first derivative of the solution; the orthogonal subscale stabilization (OSS) [23], a term that penalizes the $\mathbb{L}_2$ projection of the gradient of the error within the elements. As the CIP stabilization, this technique does not affect the mass matrix, but it requires the solution of another linear system for the $\mathbb{L}_2$ projection. In this respect, the choice of the finite element space and of the quadrature have enormous impact on the cost of the method. Note that the strategy to impose boundary conditions also plays a major role in ensuring stability [4, 5], but this will not be considered here.

Our objective is to perform a fully discrete spectral analysis on triangulations of the spatial domain to characterize the stability and accuracy of different combinations of approximation, quadrature, stabilization, and time stepping. In the linear case, this allows to propose optimal values of the CFL and stabilization parameters. Moreover, we provide some heuristic strategy to include in the analysis the impact of residual based high order diffusion operators aiming at stabilizing discontinuities. Numerical simulations for both linear and non-linear scalar problems, and for the shallow water system confirm the theoretical results, and allow to further investigate the impact of the discretization choices on the performance of the schemes and on their cost.

The paper is organized as follows. In Section 2 we describe the continuous Galerkin discretization, the stabilization techniques, the basis functions and the time integration techniques. In Section 3 we introduce the Fourier analysis space definitions that lead to von Neumann analysis, we discuss some technical details on the passage from physical functions to Fourier modes for different meshes and we find the parameters for which the schemes are stable for some mesh configurations. In Section 3.9, we also propose to introduce a viscosity term in order to enforce stability when the previous von Neumann analysis reveals instabilities. In Section 4 and Section 5 we test the found parameters on some linear and nonlinear problems, checking the order of accuracy and the computational times. Finally, in Section 6 we derive some conclusions on the presented schemes and possible applications of the found results.

2 Numerical discretization

In this section we describe the discretization of the hyperbolic conservation law (1). We consider a tessellation of the spatial domain Ω consisting of non overlapping (triangular) cells, which we denote by $\Omega_h \subset \mathbb{R}^2$. The generic element of the tessellation Ω_h will be denoted by K, so that $\Omega_h = \bigcup K$. We denote the set of internal element boundaries (edges) of Ω_h by $\mathcal{F}_h$, using f for a general element. h denotes the characteristic mesh size of Ω_h. Despite of the fact that most of the discussion is performed for the scalar case, most of it generalizes readily to systems. If a significant difference arises in this generalization, this will be explicitly discussed.

The discrete solution is sought in a continuous finite element space $V_h^p = \{v_h \in \mathcal{C}^0(\Omega_h) : v_h|K \in \mathbb{P}_p(K), \forall K \in \Omega_h\}$. We will use nodal and modal finite elements, and we will denote by φ_j the basis functions associated to the

degree of freedom j, so that $V_h^p = \text{span}\,\{\varphi_j\}_{j\in\Omega_h}$ and we can write

$$u_h(x) = \sum_{j\in\Omega_h} u_j\varphi_j(x),$$

where, with an abuse of notation, with $j \in \Omega_h$ we mean the set of degrees of freedom with support in Ω_h. With a similar meaning, we will also use the notation $j \in K$ to mean the degrees of freedom with support on the cell K.

The unstabilized CG approximation of (1) reads: find $u_h \in V_h^p$ such that for any $v_h \in W_h \subset \mathbb{L}_2(\Omega_h) := \{v : \Omega_h \to \mathbb{R} : \int_{\Omega_h} |v|^2 < \infty\}$

$$\int_{\Omega_h} v_h\partial_t u_h dx - \int_{\Omega_h} \nabla v_h f(u_h)\, dx + \int_{\partial\Omega_h} v_h f(u_h)\cdot \mathbf{n} d\Gamma = 0, \tag{3}$$

where $\mathbf{n}$ is the normal to the boundary facing outward the domain. The choice of W_h will be based on V_h, but it may take different forms for different stabilizations.

As already said, we will consider several stabilized variants of Equation (3) which can be all formulated in the form: find $u_h \in V_h^p$ that satisfies

$$\int_\Omega v_h(\partial_t u_h + \nabla\cdot f(u_h))dx + S(v_h, u_h) = 0, \quad \forall v_h \in V_h^p \tag{4}$$

where the flux term is written before the integration by part as we will consider only continuous piecewise polynomials approximations, whose derivatives are integrable. Here, S denotes a bilinear stabilization operator defined on $V_h^p \times V_h^p$. Several different choices for S exist, and are discussed in detail in the following sections.

2.1 Stabilization Terms

2.1.1 Streamline-Upwind/Petrov-Galerkin - SUPG

The SUPG method was introduced in [31] (see also [32, 12] and references therein) and is strongly consistent in the sense that it vanishes when replacing the discrete solution with the exact one. It can be written as a Petrov-Galerkin method replacing v_h in Equation (3) with a test function belonging to the space

$$W_h := \{w_h : \quad w_h = v_h + \tau_K \nabla_u f(u_h)\cdot\nabla v_h; \quad v_h \in V_h^p\}. \tag{5}$$

Here, $\nabla_u f(u_h) \in \mathbb{R}^{D\times D\times 2}$ is the Jacobian of the flux, D the dimensions of the system, τ_K denotes a positive definite stabilization parameter with the dimensions of $D \times D$ that we will assume to be constant for every element. Although other definitions are possible, here we will evaluate this parameter as

$$\tau_K = \delta h_K (J_K)^{-1} \tag{6}$$

where h_K is the cell diameter and J_K represents the norm of the flux Jacobian on a reference value of the element K. In the scalar case, $J_K = ||\nabla_u f(u)||_K$.

The final stabilized variational formulation of (4) reads

$$\int_\Omega v_h\partial_t u_h\, dx + \int_\Omega v_h\nabla\cdot f(u_h)\, dx + \underbrace{\sum_{K\in\Omega}\int_K \big(\nabla_u f(u_h)\cdot\nabla v_h\big)\tau_K\left(\partial_t u_h + \nabla\cdot f(u_h)\right)\, dx}_{S(v_h,u_h)} = 0. \tag{7}$$

The main problem of this stabilization method is that it depends on the time derivative of u and, hence, it does not maintain the structure of the mass matrix in most cases.

To characterize the accuracy of the method, we can use the consistency analysis discussed *inter alia* in [7, §3.1.1 and §3.2]. In particular, of a finite element polynomial approximation of degree p we can easily show that given a

smooth exact solution $u^e(t,x)$, replacing formally u_h by the projection of u^e on the finite element space, we can write

$$\begin{aligned}\epsilon(\psi_h) :=& \Big| \int_\Omega \psi_h \partial_t (u_h^e - u^e)\, dx - \int_\Omega \nabla\psi_h \cdot (\nabla f(u_h^e) - \nabla f(u^e))\, dx \\ &+ \sum_{K\in\Omega} \sum_{l,m\in K} \frac{\psi_l - \psi_m}{k+1} \int_K \left(\nabla_u f(u_h)\cdot\nabla\varphi_i\right)\tau_K \left(\partial_t(u_h^e - u^e) + \nabla\cdot(f(u_h^e) - f(u^e))\right)\, dx \Big| \le C h^{p+1},\end{aligned} \tag{8}$$

with C a constant independent of h, for all functions ψ of class at least $\mathcal{C}^1(\Omega)$, of which ψ_h denotes the finite element projection. A key point in this estimate is the strong consistency of the method allowing to subtract its formal application to the exact solution (thus subtracting zero), and obtaining the above expression featuring differences between the exact solution/flux and its evaluation on the finite element space. Preserving this error estimate precludes the possibility of lumping the mass matrix, and in particular the entries associated to the stabilization term. This makes the scheme relatively inefficient when using standard explicit time stepping.

As a final note, for a linear flux Equation (2), exact integration, with $\tau_K = \tau$ and in the time continuous case, a classical result is obtained for homogeneous boundary conditions by testing with $v_h = u_h + \tau\,\partial_t u_h$ [12]:

$$\int_{\Omega_h} \partial_t \left(\frac{u_h^2}{2} + \tau^2 \frac{(\mathbf{a}\cdot\nabla u_h)^2}{2} \right) + \int_{\Omega_h} \mathbf{a}\cdot\nabla \left(\frac{u_h^2}{2} + \tau^2 \frac{(\partial_t u_h)^2}{2} \right) = -\int_{\Omega_h} \tau(\partial_t u_h + \mathbf{a}\cdot\nabla u_h)^2. \tag{9}$$

For periodic, or homogeneous boundary conditions, this shows that the norm $|||u|||^2 := \int_{\Omega_h} \frac{u_h^2}{2} + \tau^2 \frac{(\mathbf{a}\cdot\nabla u_h)^2}{2} dx$ is non-increasing. The interested reader can refer to [12] for the analysis of some (implicit) fully discrete schemes.

2.1.2 Note on the SUPG technique applied to non scalar problems

The extension of the SUPG method to a non scalar problem is not straightforward. Here we used the following formulation. First, we define the following system of dimension D:

$$\begin{cases} \partial_t U + \nabla\cdot\mathcal{F}(U) = \mathbf{S}(U) \\ \mathcal{F} = (F_1, F_2) \end{cases} \tag{10}$$

with $U \in \mathbb{R}^D$, $\mathcal{F}(U) \in \mathbb{R}^{2\times D}$ and $\mathbf{S}(U) \in \mathbb{R}^D$. For example, in the results section we will consider the shallow water equations with $D = 3$ which read

$$U = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix} \quad F_1(U) = \begin{pmatrix} hu \\ hu^2 + g\frac{h^2}{2} \\ huv \end{pmatrix} \quad F_2(U) = \begin{pmatrix} hv \\ huv \\ hv^2 + g\frac{h^2}{2} \end{pmatrix} \quad \text{and} \quad \mathbf{S}(U) = \begin{pmatrix} 0 \\ -ghb_x \\ -ghb_y \end{pmatrix}$$

where $\mathbf{S}(U)$ is the source term given by a topography term. Equation (10) can also be written in its quasi-linear form

$$\partial_t U + \nabla_U \mathcal{F}(U)\cdot\nabla U = \mathbf{S}(U), \tag{11}$$

where $\nabla_U \mathcal{F}(U_h) \in \mathbb{R}^{D\times D\times 2}$ is the Jacobian of the flux $\mathcal{F}(U_h)$.

Following the definition of the SUPG method and [52, sec. 5] we define a positive-definite stabilization matrix $\tau_{\mathbf{K}} \in \mathbb{R}^{D\times D}$constant for every element K. Here this matrix is evaluated as [52]

$$\tau_K = \delta h_K \left(\sum_{j\in S_K} \left| \nabla_U \mathcal{F}(\bar{U}_K)\cdot n_j \right| \right)^{-1}, \tag{12}$$

with S_K the set of vertices of K, and n_j the outward normal of the edge opposite to the vertex $j \in S_K$. h_K is the cell diameter and $\nabla_u \mathcal{F}(\bar{U}_K)$ represents the flux Jacobian of the the average value of U_h on the element K.

The SUPG stabilized formulation reads, for each equation of the system $i = 1, \dots, D$

$$\int_\Omega v_h \left(\partial_t U_h + \nabla \cdot \mathcal{F}(U_h) - \mathbf{S}(U_h)\right)_i + \underbrace{\left(\sum_{K \in \Omega} \int_K \left(\nabla v_h \cdot \nabla_U \mathcal{F}(U_h)\right) \tau_K \left(\partial_t U_h + \nabla \cdot \mathcal{F}(U_h) - \mathbf{S}(U_h)\right) \, dx \right)_i}_{S(v_h, U_h)_i} = 0, \tag{13}$$

where $(V)_i$ denotes the i-th component of a vector $V \in \mathbb{R}^D$.

2.1.3 Continuous Interior Penalty - CIP

Another stabilization technique, which maintains sparsity and symmetry of the Galerkin matrix, is the continuous interior penalty (CIP) method. It was developed by Burman and Hansbo originally in [15] and then in a series of works [16, 19, 14]. It can also be seen as a variation of the method proposed by Douglas and Dupont [26].

The method stabilizes the Galerkin formulation by adding edge penalty terms proportional to the jump of the gradient of the derivatives of the solution across the cell interfaces. The CIP introduces high order viscosity to the formulation, allowing the solution to tend to the vanishing viscosity limit. This term is does not affect the structure of the mass matrix. The method reads

$$\int_{\Omega_h} v_h \partial_t u_h \, dx + \int_{\Omega_h} v_h \nabla \cdot f(u_h) \, dx + \underbrace{\sum_{\mathsf{f} \in \mathcal{F}_h} \int_{\mathsf{f}} \tau_{\mathsf{f}} [\![n_{\mathsf{f}} \cdot \nabla v_h]\!] \cdot [\![n_{\mathsf{f}} \cdot \nabla u_h]\!] \, d\Gamma}_{S(v_h, u_h)} = 0, \tag{14}$$

where $[\![\cdot]\!]$ denotes the jump of a quantity across a face f, n_{f} is a normal to the face f and where $\mathcal{F}_h$ is the collection of internal boundaries, and f are its elements. Although other definitions are possible, we evaluate the scaling parameter in the stabilization as

$$\tau_{\mathsf{f}} = \delta \, h_{\mathsf{f}}^2 \|\nabla_u f\|_{\mathsf{f}}, \tag{15}$$

where $\|\nabla_u f\|_{\mathsf{f}}$ a reference value of the norm of the flux Jacobian on f and h_{f} a characteristic size of the mesh neighboring f.

As stated above, a clear advantage of CIP is that it does not modify the mass matrix, allowing to obtain efficient schemes if a mass lumping strategy can be devised. On the other side, the stencil of the scheme increases as the jump of a degree of freedom interacts with cells which are not next to the degree of freedom itself (up to 2 cells distance). Note that for higher order approximations [17, 38] suggest the use of jumps in higher derivatives to improve the stability of the method. However, here we consider the jump in the first derivatives in order to be able to apply the stability analysis and to study the influence of δ on the stability of the method. Some results might be definitely improved adding these stabilizations on higher derivatives.

The accuracy of CIP can be assessed with a consistency analysis as discussed in [7, §3.1.1 and §3.2]. This consists in, formally substituting u_h by the projection onto the finite element polynomial of degree p space of u^e, a given smooth exact solution $u^e(t, x)$, we can show that for all functions ψ of class at least $\mathcal{C}^1(\Omega)$, of which ψ_h denotes the finite element projection, we have the truncation error estimate

$$\begin{aligned} \epsilon(\psi_h) := \Big| \int_\Omega \psi_h \partial_t (u_h^e - u^e) \, dx - \int_\Omega \nabla \psi_h \cdot (f(u_h^e) - f(u^e)) \, dx \\ + \sum_{\mathsf{f} \in \mathcal{F}_h} \int_{\mathsf{f}} \tau_{\mathsf{f}} [\![n_f \cdot \nabla \psi_h]\!] \cdot [\![n_f \cdot \nabla (u_h^e - u^e)]\!] \Big| \leq C h^{p+1}, \end{aligned} \tag{16}$$

with C a constant independent of h. The estimate can be derived from standard approximation results applied to $u_h^e - u^e$ and to its derivatives, noting that τ_{f} is an $\mathcal{O}(h^2)$, which allows to obtain the estimation with the right order.

The symmetry of the stabilization allows to easily derive an energy stability estimate for the space discretized scheme only. In particular, for periodic boundary conditions and a linear flux we can easily show that

$$\int_{\Omega_h} \partial_t \frac{u_h^2}{2} = - \sum_{\mathsf{f} \in \mathcal{F}_h} \int_{\mathsf{f}} \tau_{\mathsf{f}} [\![n_f \cdot \nabla u_h]\!]^2, \tag{17}$$

which gives a bound in time on the $\mathbb{L}_2$ norm of the solution.

Note that for higher than second order it may be relevant to consider additional penalty terms based on higher derivatives (see e.g. [17, 13, 3]). We did not do this in this work.

2.1.4 Orthogonal Subscale Stabilization - OSS

Another symmetric stabilization approach is the Orthogonal Subscale Stabilization (OSS) method. Originally introduced as Pressure Gradient Projection (GPS) in [24] for Stokes equations, it was extended to the OSS method in [23, 11] for different problems with numerical instabilities, such as convection–diffusion–reaction problems. This stabilization penalizes the fluctuations of the gradient of the solution with a projection of the gradient onto the finite element space. The method applied to Equation (3) reads: find $u_h \in V_h^p$ such that $\forall v_h \in V_h^p$

$$\begin{cases} \int_{\Omega_h} v_h \partial_t u_h \, dx + \int_{\Omega_h} v_h \nabla \cdot f(u_h) \, dx + \underbrace{\sum_{K \in \Omega_h} \int_K \tau_K \nabla v_h \cdot (\nabla u_h - w_h) \, dx}_{S(v_h, u_h)} = 0, \\ \int_{\Omega_h} v_h w_h \, dx - \int_{\Omega_h} v_h \nabla u_h \, dx = 0. \end{cases} \tag{18}$$

For this method, the stabilization parameter is evaluated as

$$\tau_K = \delta h_K \|\nabla_u f\|_K. \tag{19}$$

The drawback of this method, with respect to CIP, is the requirement of a matrix inversion to project the gradient of the solution in the second equation of (18). This cost can be alleviated by the choice of elements and quadrature rules if they result in a diagonal mass matrix, as it will be the case for *Cubature* elements that we will describe below.

As before we can easily characterize the accuracy of this method. The truncation error estimate for a polynomial approximation of degree p reads in this case

$$\begin{aligned} \epsilon(\psi_h) := \Big| \int_{\Omega_h} \psi_h \partial_t (u_h^e - u^e) \, dx - \int_{\Omega_h} \nabla \psi_h \cdot (f(u_h^e) - f(u^e)) \, dx \\ + \sum_{K \in \Omega_h} \tau_K \int_K \nabla \psi_h \cdot \nabla (u_h^e - u^e) + \sum_{K \in \Omega_h} \tau_K \int_K \nabla \psi_h \cdot (\nabla u^e - w_h^e) \Big| \leq C h^{p+1}, \end{aligned} \tag{20}$$

where the last term is readily estimated using the projection error and the boundness of ψ_h as

$$\int_{\Omega_h} \psi_h (w_h^e - \nabla u^e) \, dx = \int_{\Omega_h} \psi_h (\nabla u_h^e - \nabla u^e) = \mathcal{O}(h^p).$$

Finally, for a linear flux, periodic boundaries and taking $\tau_K = \tau$ constant along the mesh, we can test with $v_h = u_h$ in the first equation of (18), and with $v_h = \tau w_h$ in the second one and sum up the result to get

$$\int_{\Omega_h} \partial_t \frac{u_h^2}{2} = -\sum_K \int_K \tau_K (\nabla u_h - w_h)^2, \tag{21}$$

which can be integrated in time to obtain a bound on the $\mathbb{L}_2$ norm of the solution.

The truncation consistency error analysis presented above for the three stabilization terms is completely formal and it does not comprehend an entire classical error analysis. These estimations tell us that the stabilization terms that we introduced are of the wanted order of accuracy and that they are usable to aim at the prescribed order of accuracy. This type of analysis has been already done for multidimensional problems inter alia in [2]. More rigorous proof of error bounds with $h^{p+\frac{1}{2}}$ estimates can be found in [13] for the CIP. We did not consider in this work projection stabilizations involving higher derivatives.

2.2 Finite Element Spaces and Quadrature Rules

In this section we describe three finite element polynomial approximation strategies used in the paper. In particular, on a triangular element K of Ω_h, we define in this section the restriction of the basis functions of V_h^p on each element K, which are polynomials of degree at most p. We denote by $\{\varphi_1, \dots, \varphi_N\}$ the basis functions and they will have degree at most p, and their definitions amounts to describe the degrees of freedom, i.e., the dual basis.

2.2.1 *Basic* Lagrangian equispaced elements

On triangles, we consider Lagrange polynomials with degrees at most p: $\mathbb{P}^p = \{\sum_{\alpha+\beta\le p} c_{\alpha,\beta} x^\alpha y^\beta\}$. We define the barycentric coordinates $\lambda_i(x, y)$ which are affine functions on $\mathbb{R}^2$ verifying the following relations

$$\lambda_i(v_j) = \delta_{ij}, \quad \forall i, j = 1, \dots, 3, \tag{22}$$

where $v_j = (x_j, y_j)$ are the vertexes of the triangle and, with an abuse of notation, they can be written in barycentric coordinates as $v_j = (\delta_{1j}, \delta_{2j}, \delta_{3j})$. Using these coordinates, we can define the Lagrangian polynomials on equispaced points on triangles. The equispaced points are defined on the intersection of the lines $\lambda_j = \frac{k}{p}$ for $k = 0, \dots, p$. A way to define the basis functions corresponding to the point $(x_\alpha, y_\alpha) = (\alpha_1/p, \alpha_2/p, \alpha_3/p)$ in barycentric coordinates, with $\alpha_i \in \{0, \dots, p\}$ and $\sum_i \alpha_i = 1$, is in Algorithm 1.

Algorithm 1 Lagrangian basis function in barycentric coordinates

Require: Point $(x_\alpha, y_\alpha) = (\alpha_1/p, \alpha_2/p, \alpha_3/p)$ in barycentric coordinates
  $\varphi_\alpha(x) \leftarrow 1$
  for $i = 1, 2, 3$ **do**
    for $z = 0, \dots, a_i$ **do**
      $\varphi_\alpha(x, y) \leftarrow \varphi_\alpha(x, y) \cdot (\lambda_i(x, y) - \frac{z}{p})$
    end for
  end for

The polynomials so defined in a triangle form a partition of unity, but they have also negative values. This leads to negative or zero values of their integrals. This is problematic for some time discretization and we will see why. We will use these polynomials in combination with exact Gauss–Lobatto quadrature formulae for such polynomials and we will refer to them as *Basic* elements.

2.2.2 Bernstein polynomials

Bernstein polynomials are as well a basis of $\mathbb{P}^p$ but they are not Lagrangian polynomials, hence, there is not a unique correspondence between point values and coefficients of the polynomials. Anyway, there exist a geometrical identification with the Greville points $(x_\alpha, y_\alpha) = (\alpha_1/p, \alpha_2/p, \alpha_3/p)$. Given a triplet $\alpha \in \mathbb{N}^3$ with $\alpha_i \in [\![0, \dots, p]\!]$ and $\sum_i \alpha_i = p$, the Bernstein polynomials are defined as

$$\varphi_\alpha(x, y) = p! \prod_{i=1}^{3} \frac{\lambda_i^{\alpha_i}(x, y)}{\alpha_i!}. \tag{23}$$

Bernstein polynomials verify additional properties besides the one already cited for Lagrangian points. As before, they form a partition of unity, the basis functions are nonnegative in any point of the triangle, and so their integrals are strictly positive. More precisely

$$\int_K \varphi_\alpha = \frac{|K|}{S}, \qquad S = \#\left\{\alpha \in \mathbb{N}^3 : |\alpha|_1 = p\right\}.$$

These properties lead also to the fact that the value at each point is a convex combination of the coefficients of the polynomials, so that it is easy to bound minimum and maximum of the function by the minimum and maximum of

the coefficients. This has been used in different techniques to preserve positivity of the solution [10, 37]. We will use these polynomials with corresponding high order accurate quadrature formulae. We will denote these elements with the symbol $\mathbb{B}^p$ and we refer to them as *Bernstein* elements.

2.2.3 *Cubature* elements

Contrary to the work done in 1D [42], the extension of Legendre–Gauss–Lobatto points which minimize the interpolation error do not exist for the triangle. They have to be computed numerically such as *Fekete* points [34, 55, 57]. The problem of this approach is that it requires as classical finite elements the inversion of a sparse global mass matrix. *Cubature* elements were introduced by G. Cohen and P. Joly in 2001 [25] for the wave equation (second order hyperbolic equation), and are an extension of Lagrange polynomials with the goal of optimizing the underlying quadrature formula error. We will denote the with the symbol $\tilde{\mathbb{P}}^p$ and they will be contained in another larger space of Lagrange elements, i.e., $\mathbb{P}^p \subseteq \tilde{\mathbb{P}}^p \subseteq \mathbb{P}^{p'}$, with p' the smallest possible integer. Similar techniques have been used to minimize the interpolation error [34, 55, 57]. The objective of these polynomials is to use the points of the Lagrangian interpolation of the polynomials as quadrature points. This means that the obtained quadrature is $\int_K f(x,y) = \sum_\alpha \omega_\alpha f(x_\alpha, y_\alpha)$, where $\int_K \varphi_\alpha = \omega_\alpha$ and $\varphi_\alpha(x_\beta, y_\beta) = \delta_{\alpha\beta}$. This approach can be considered an extension of the Gauss–Lobatto quadrature in 1D for non Cartesian meshes. The biggest advantage of this approach is to obtain a diagonal mass matrix. The drawback is that one needs to increase the number of basis function inside one element to obtain an accurate enough quadrature rule. In our work, we propose to extend this approach to first order hyperbolic equations. A successful extension to elliptic problem is proposed in [51]. A comparison between the equispace repartition and the *Cubature* repartition for elements of degree $p = 3$ is shown in Figure 1.

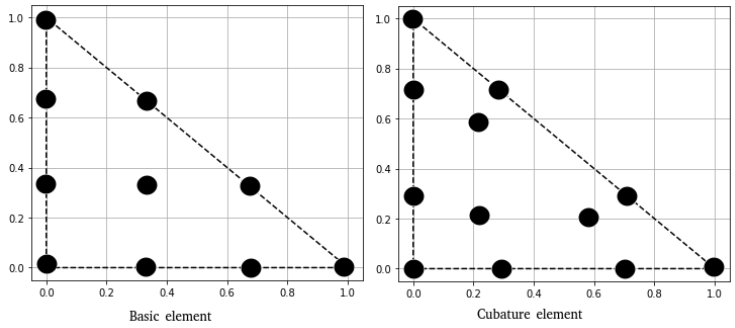


Figure 1: Comparison of the equispace repartition at left and the cubature repartition at right for elements of degree $p = 3$.

For completeness we detail further the construction of the basis functions. The challenges of this approach are the following:

- Obtain a quadrature which is highly accurate, at least $p + p' - 2$ order accurate [22];
- Obtain positive quadrature weights $\omega_\alpha > 0$ for stability reasons [58];
- Minimize the number of basis functions of $\tilde{\mathbb{P}}^p$;
- The set of quadrature points has to be $\tilde{\mathbb{P}}^p$-unisolvent;
- The number of quadrature points of edges as to be sufficient ensure the conformity of the finite element.

The optimization procedure that lead to these elements consists of several steps where the different goals are optimized one by one. The optimization strategy exploits heavily the symmetry properties that the quadrature point must have.

For $p = 1$ the *Cubature* elements do not differ from the *Basic* elements but in the quadrature formula. For $p = 2$ the *Cubature* elements introduce an other degree of freedom at the center of the triangle, leading to 7 quadrature points

and basis functions per element. For $p = 3$ the additional degree of freedom in the triangle are 3, leading to 13 basis functions per triangle. All the details of such elements can be found in [25, 33]. We provide in Appendix A the detailed expressions of the polynomials used in this work. We will use the symbol $\tilde{\mathbb{P}}^p$ and the name *Cubature* elements to refer to them.

Other elements such as *Fekete-Gauss* points [29, 50] exist in the literature. They are optimized to interpolate and integrate with high accuracy. However, it is shown that they require more computing time to achieve similar results than cubature points for high order of accuracy.

2.3 Time integration

The spatial discretization leads to a coupled system of ordinary differential equation which can be written as

$$\mathbb{M}\frac{dU}{dt} = \mathtt{r}(t) \tag{24}$$

where U is the vector of all the degrees of freedom on all the domain, $\mathbb{M}$ and $\mathtt{r}$ are the global mass matrix and right-hand side terms obtained through the discretization of the previous section with some finite elements and stabilization terms. We remark that $\mathbb{M}$ is diagonal only in the case of the *Cubature* elements without the SUPG stabilization, while, for all other choices, it is a sparse non–diagonal matrix.

In the following, we describe two different time integration method: explicit Runge–Kutta (RK) methods and their strong stability preserving (SSP) variants; and the Deferred Correction (DeC) algorithm, which allows to avoid the mass matrix inversion through the correction iterations.

2.3.1 Explicit Runge–Kutta and Strong Stability Preserving Runge–Kutta schemes

Runge–Kutta time integration methods are one step methods consisting in S stages defined by

$$\begin{aligned} U^{(0)} &:= U^n, \\ U^{(s)} &:= U^n + \Delta t \sum_{j=0}^{s-1} \alpha_j^s \mathbb{M}^{-1}\mathtt{r}(U^{(j)}) \quad s = 1, \dots, S, \\ U^{n+1} &:= U^n + \Delta t \sum_{s=0}^{S} \beta_s \mathbb{M}^{-1}\mathtt{r}(U^{(s)}). \end{aligned} \tag{25}$$

Here, we use for the solution the superscript n to indicate the timestep and the superscript in brackets (s) to denote the stage of the method. The coefficients α_j^s and β_j^s can be defined in many different ways. In particular, we will refer to Heun's method with RK2, to Kutta's method with RK3 and the original Runge–Kutta fourth order method as RK4. The respective Butcher tables can be found in Appendix B in Table 12, see [20].

A subset of the RK methods are the SSPRK introduced in [56]. They consist in convex combinations of forward Euler steps, and can be rewritten as follows

$$\begin{aligned} U^{(0)} &:= U^n, \\ U^{(s)} &:= \sum_{j=0}^{s-1} \left(\gamma_j^s U^{(j)} + \Delta t \mu_j^s \mathbb{M}^{-1}\mathtt{r}(U^{(j)}) \right) \quad s = 1, \dots, S, \\ U^{n+1} &:= U^{(S)}, \end{aligned} \tag{26}$$

with $\gamma_j^s, \mu_j^s \geq 0$ for all $j, s = 1, \dots, S$. We will consider here the second order 3 stages SSPRK(3,2) presented by Shu and Osher in [56], the third order SSPRK(4,3) presented in [54, Page 189], and the fourth order SSPRK(5,4) defined in [54, Table 3]. For complete reproducibility of the results, we put all their Butcher's tableaux in Appendix B in Table 13.

2.3.2 The *Deferred Correction* scheme

Deferred Correction methods were introduced in [27] as explicit time integration methods for ODEs, but soon implicit [45], linearly implicit positivity preserving [48] versions and extensions to PDE solvers [1] were studied. In particular, in [1, 8, 3, 6] the DeC is used in a different formulation for finite element methods and it introduces two operator through which it is possible to use a diagonal mass matrix without losing the accuracy order. This is only achievable when the lumped matrix (defined as the sum on the rows of the full mass matrix) has only positive values on its diagonal. Hence, the use of *Bernstein* polynomials is recommended in [1], but also *Cubature* elements can serve the purpose.

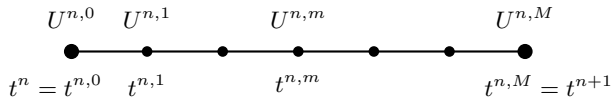


Figure 2: Subtimesteps inside the time step $[t^n, t^{n+1}]$

Consider a discretization of each timestep into M subtimesteps as in Figure 2. For each subtimestep we define a high order approximation of the integral form of the ODE (24) from $t^{n,0}$ to $t^{n,m}$, i.e.,

$$\mathbb{M}\left(U^{n,m} - U^{n,0}\right) - \int_{t^{n,0}}^{t^{n,m}} \mathtt{r}(U(s))ds \approx \mathcal{L}^2(\underline{U})^m := \mathbb{M}\left(U^{n,m} - U^{n,0}\right) - \Delta t \sum_{z\in[\![0,M]\!]} \rho_z^m \mathtt{r}(U^{n,z}) = 0, \tag{27}$$

with $\underline{U} = \left(U^{n,0}, \dots, U^{n,M}\right)$. Moreover, the quadrature rule in time uses the subtimesteps $t^{n,m}$ as quadrature points. The corresponding weights ρ_z^m for every different subinterval are defined by Lagrangian basis functions in these subtimesteps (see [1, 8, 3] for details). The algebraic system $\mathcal{L}^2(\underline{U}^*) = 0$ is in general implicit and nonlinear and, in order not to recast to nonlinear solvers, the DeC procedure approximates the solution of $\mathcal{L}^2(\underline{U}^*) = 0$ by successive iterations relying on a low order easy–to–invert operator $\mathcal{L}^1$. This operator is typically a first order forward Euler approximation with a lumped mass matrix, i.e.,

$$\mathbb{M}\left(U^{n,m} - U^{n,0}\right) - \int_{t^{n,0}}^{t^{n,m}} \mathtt{r}(U(s))ds \approx \mathcal{L}^1(\underline{U})^m := \mathbb{D}\left(U^{n,m} - U^{n,0}\right) - \Delta t \beta^m \mathtt{r}(U^{n,0}) = 0. \tag{28}$$

Here, $\mathbb{D}$ denotes a diagonal matrix obtained from the lumping of $\mathbb{M}$, i.e., $\mathbb{D}_{ii} := \sum_j \mathbb{M}_{ij}$, and $\beta^m := \frac{t^{n,m} - t^{n,0}}{t^{n+1} - t^n}$. The values of the coefficients β^m and ρ_z^m for equispaced subtimesteps can be found in Appendix B. Denoting with the superscript (k) index the iteration step, we describe the DeC algorithm as

$$U^{n,m,(0)} := U^n \qquad m = 0, \dots, M, \tag{29a}$$

$$U^{n,0,(k)} := U^n \qquad k = 0, \dots, K, \tag{29b}$$

$$\mathcal{L}^1(\underline{U}^{(k)}) = \mathcal{L}^1(\underline{U}^{(k-1)}) - \mathcal{L}^2(\underline{U}^{(k-1)}) \qquad k = 1, \dots, K, \tag{29c}$$

$$U^{n+1} := U^{n,M,(K)}. \tag{29d}$$

It has been proven [1] that if $\mathcal{L}^1$ is coercive, $\mathcal{L}^1 - \mathcal{L}^2$ is Lipschitz with a constant $\alpha_1 \Delta t > 0$ and the solution of $\mathcal{L}^2(\underline{U}^*) = 0$ exists and is unique, then, the method converges with an error of $\mathcal{O}(\Delta t^K)$. Hence, choosing $K = M + 1$ we obtain a K-th order accurate scheme.

Relying only on the inversion of the low order operator, the method gets rid of the computational costs of the solution of the linear systems, leaving in the right hand side the mass matrix of the $\mathcal{L}^2$ operator, that should not be inverted. The only requirement that is necessary for the DeC approach is the invertibility of the lumped mass matrix $\mathbb{D}$, which limits its application to spatial elements which possess this property.*Basic* Lagrange polynomials do not guarantee such constraint already for degree 2. Hence, only other polynomials as *Bernstein* and *Cubature* can be used in combination with DeC.

Finally, for the following analysis we note that the DeC method can be cast in a form similar to a Runge–Kutta method by rewriting Equation (29c) as

$$U^{n,m,(k+1)} = U^{n,m,(k)} - \mathbb{D}^{-1}\mathbb{M}\left(U^{n,m,(k)} - U^{n,0,(k)}\right) + \sum_{j=0}^{M} \Delta t \rho_j^m \mathbb{D}^{-1}\mathtt{r}(U^{n,j,(k)}). \tag{30}$$

Comparing with the system of equations (26), we can immediately define the SSPRK coefficients associated to DeC as $\gamma_{m,(k)}^{m,(k+1)} = \mathbb{I} - \mathbb{D}^{-1}\mathbb{M}$ with $\mathbb{I}$ the identity matrix, $\gamma_{0,(0)}^{m,(k+1)} = \mathbb{D}^{-1}\mathbb{M}$, $\mu_{r,(k)}^{m,(k+1)} = \rho_r^m$ for $m, r = 0, \ldots, M$ and $k = 0, \ldots, K-1$ and instead of the mass matrix, we use the diagonal one.

Remark 1 (DeC with SUPG) *The iterative procedure of the DeC method allows even to overcome the difficulties that some implicit stabilization as the SUPG has. Indeed, the SUPG stabilization term can be added only to the $\mathcal{L}^2$ operator, keeping the high order accuracy of this operator. Since the $\mathcal{L}^2$ operator is applied to the previously computed iteration, all the terms of the SUPG, included the time derivative of u in Equation* (7)*, can be explicitly computed on $U^{(k-1)}$, keeping then the diagonal mass matrix for the whole scheme.*

3 Fourier analysis

3.1 Preliminaries and time continuous analysis

In order to study the stability and the dispersion properties of the previously presented numerical schemes, we will perform a dispersion analysis on the linear advection problem with periodic boundary conditions:

$$\partial_t u(t,\mathbf{x}) + \mathbf{a}\cdot\nabla u(t,\mathbf{x}) = 0, \quad \mathbf{a}\in\mathbb{R}^2, \quad (t,\mathbf{x})\in\mathbb{R}^+\times\Omega, \tag{31}$$

with $\Omega = [0,1]\times[0,1]$. For simplicity, we consider $\mathbf{a} = (\cos(\Phi), \sin(\Phi))$ with $\Phi \in [0, 2\pi]$. We then introduce the ansatz

$$u_h(\mathbf{x},t) = Ae^{i(\mathbf{k}\cdot\mathbf{x}-\xi t)} = Ae^{i(\mathbf{k}\cdot\mathbf{x}-\omega t)}e^{\epsilon t} \tag{32}$$

$$\text{with} \quad \xi = \omega + i\epsilon, \quad i = \sqrt{-1}, \quad \mathbf{k} = (k_x, k_y)^T. \tag{33}$$

Here, ϵ denotes the damping rate, while the wavenumbers are denoted by $\mathbf{k} = (k_x, k_y)$, with $k_x = 2\pi/L_x$ and $k_y = 2\pi/L_y$ with L_x and L_y the wavelengths in x and y directions respectively. The phase velocity $\mathbf{c}$ can be defined from

$$\mathbf{c}\cdot\mathbf{k} = \omega \tag{34}$$

and represents the celerity with which waves propagate in space. It is in general a function of the wavenumber. Substituting (32) in the advection equation (31) for an exact solution we obtain that

$$\omega = \mathbf{k}\cdot\mathbf{a}\,, \quad \mathbf{c} = \mathbf{a} \quad \text{and} \quad \epsilon = 0. \tag{35}$$

In other words

$$u_h(\mathbf{x},t) = Ae^{i\mathbf{k}\cdot(\mathbf{x}-\mathbf{a}t)}\,. \tag{36}$$

The objective of the next sections is to provide the semi- and fully-discrete equivalents of the above relations for the finite element methods introduced earlier. We will consider polynomial degrees up to 3, for all combinations of stabilization methods and time integration techniques. This will also allow to investigate the parametric stability with respect to the time step (through the CFL number) and stabilization parameter δ. In practice, for each choice we will evaluate the accuracy of the discrete approximation of ω and ϵ, and we will provide conditions for the non-positivity of the damping ϵ, i.e., the von Neumann stability of the method.

3.2 The eigenvalue system

The Fourier analysis for numerical schemes on the periodic domain is based on a discrete Parseval theorem. Thanks to this theorem, we can study the amplification and the dispersion of the basis functions of the Fourier space. The key ingredient of this study is the repetition of the stencil of the scheme from one cell to another one. In particular, using the ansatz (32) we can write local equations coupling degrees of freedom belonging to neighbouring cells through a multiplication by factors $e^{i\theta_x}$ and $e^{i\theta_y}$ representing the shift in space along the oscillating solution. The dimensionless coefficient

$$\theta_x := k_x \Delta x \text{ and } \theta_y := k_y \Delta y \tag{37}$$

are the discrete reduced wave numbers which naturally appear all along the analysis. Here, Δx and Δy are defined by the size of the elementary periodic unit that is highlighted with a red square as an example in Figure 3.

Formally replacing the ansatz in the scheme we end up with a dense algebraic problem of dimension N_{dof}, where N_{dof} is the number of all the degrees of freedom in the mesh. The obtained system with dimension N_{dof} in the time continuous case reads

$$\text{Equations (31) and (32)} \quad \Rightarrow \quad -i\xi\mathbb{M}\mathbf{U} + \mathbf{a}\cdot(\mathcal{K}_x\mathbf{U}, \mathcal{K}_y\mathbf{U}) + \delta\mathbb{S}\mathbf{U} = 0 \tag{38}$$

$$\text{with} \quad (\mathbb{M})_{ij} = \int_\Omega \phi_i\phi_j dx, \qquad (\mathcal{K}_x)_{ij} = \int_\Omega \phi_i\partial_x\phi_j dx, \qquad (\mathcal{K}_y)_{ij} = \int_\Omega \phi_i\partial_y\phi_j dx \tag{39}$$

with ϕ_j being any finite element basis functions, $\mathbf{U}$ the array of all the degrees of freedom and $\mathbb{S}$ being the stabilization matrix defined through one of the stabilization techniques of Section 2.1. Although system (38) is in general a global eigenvalue problem, we can reduce its complexity by exploiting more explicitly the ansatz (32). The choice of the mesh is crucial in order to exploit the ansatz and to find a unit block that repeats periodically in space. Hence, we must consider structured periodic meshes and we will focus, in particular, on two types of meshes. The first one is the X-mesh that is depicted in Figure 3 and the second one is the T-mesh depicted in Figure 4. In those pictures also the distribution of some $\mathbb{P}_2$ elements are represented as an example.

More precisely, as it is done in [55] we can introduce elemental vectors of unknowns $\widetilde{\mathbf{U}}_{Z_{ij}}$, where Z_{ij} is the stencil denoted by the red square in Figure 3, which repeats periodically on the domain. So that $\widetilde{\mathbf{U}}_{Z_{ij}}$, for continuous finite elements, is an array of d degrees of freedom inside a periodic unitary block Z_{ij}, excluding two boundaries (one on the top and one on the right for example). This number depends on the chosen (periodic) mesh type and on the elements. As an example, in Figure 3 we display for the *X* type mesh the periodic elementary unit (in the red square) with *Basic* and cubature degrees of freedom with $p = 2$. In the *X* mesh for *Basic* elements $p = 2$ we have $d = 8$, while for *Cubature* $p = 2$ we have $d = 12$. Using the periodicity of the solution and the ansatz (32) and denoting by $Z_{i\pm1,j\pm1}$ the neighboring elementary units, we can write the neighboring degrees of freedom by

$$\widetilde{\mathbf{U}}_{Z_{i\pm1,j}} = e^{\pm\theta_x}\widetilde{\mathbf{U}}_{Z_{i,j}}, \qquad \widetilde{\mathbf{U}}_{Z_{i,j\pm1}} = e^{\pm\theta_y}\widetilde{\mathbf{U}}_{Z_{i,j}}, \tag{40}$$

and by induction all other degrees of freedom of the mesh. This allows to show that the system (38) is equivalent to a compact system of dimension d (we drop the subscript $_K$ as they system is equivalent for all cells)

$$-i\xi\widetilde{\mathbb{M}}\widetilde{\mathbf{U}} + a_x\widetilde{\mathcal{K}}_x\widetilde{\mathbf{U}} + a_y\widetilde{\mathcal{K}}_y\widetilde{\mathbf{U}} + \delta\widetilde{\mathbb{S}}\widetilde{\mathbf{U}} = 0, \tag{41}$$

where the matrices $\widetilde{\mathbb{M}}$, $\widetilde{\mathcal{K}}_x$, $\widetilde{\mathcal{K}}_y$ and $\widetilde{\mathbb{S}}$ are readily obtained from the elemental discretization matrices by using Equations (40).

For the discrete Parseval theorem, we know that the norm or the reduced variable $\widetilde{\mathbf{U}}$ is equivalent to the norm of the discrete vector $\mathbf{U}$. Hence, studying the amplification factor of the two is equivalent.

We apply the same analysis to stabilized methods. The interested reader can access all 2D dispersion plots online [43]. From the plot we can see that the increase in polynomial degree provides the expected large reduction in dispersion error, while retaining a small amount of numerical dissipation, which permits the damping of *parasite* modes.

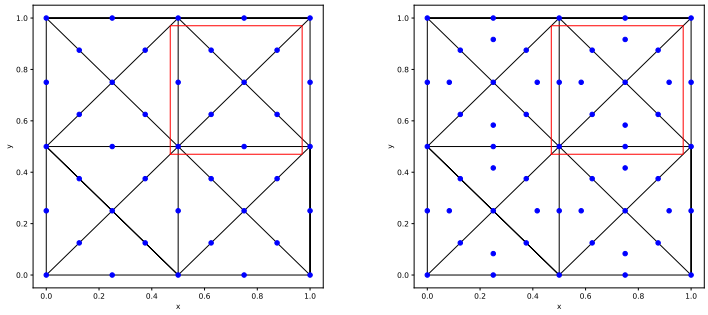
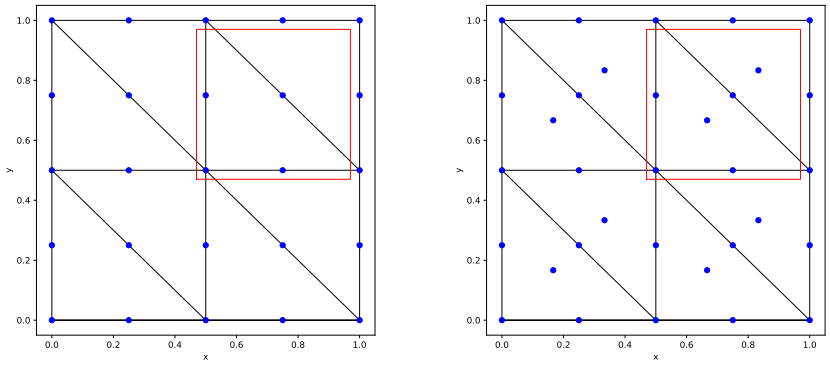


Figure 3: The *X* type triangular mesh. At left, the *Basic* finite element discretisation with $\mathbb{P}_2$ elements. At right, the grid configuration for $\tilde{\mathbb{P}}_2$ *Cubature* elements. The red square represents the periodic elementary unit that contains the degrees of freedom of interest for the Fourier analysis



(a) *Basic* $\mathbb{P}_2$ elements

(b) *Cubature* $\tilde{\mathbb{P}}_2$ elements

Figure 4: The *T* type triangular mesh with degrees of freedom in blue and periodic unit in the red square for the Fourier analysis

An example of dispersion curves is given in Figure 5. The method used *Cubature* $\tilde{\mathbb{P}}_2$ elements, the CIP stabilization technique, and a wave angle $\theta = 5\pi/4$. We here show all 12 *parasite* modes (see Figure 3). The *principal* mode of this system is represented in green. This figure also show the complexity of the analysis because of the number of modes to consider.

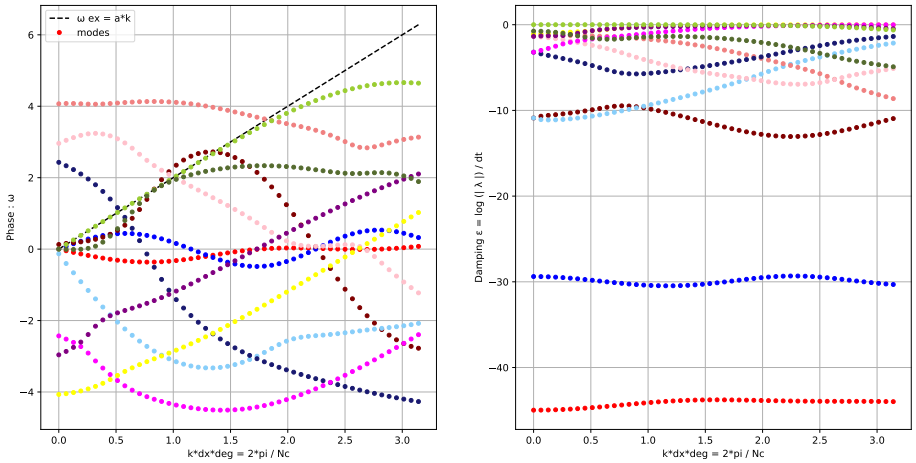


Figure 5: Dispersion curves related to the 12 modes of $\widetilde{\mathbf{U}}_{Z_{ij}}$ of the system given by *Cubature* $\tilde{\mathbb{P}}_2$ elements, the CIP stabilization technique, and a wave angle $\theta = 5\pi/4$ on an *X* mesh.
Phases ω (left) and amplifications ϵ (right).

We summarize the number of modes for the *X* mesh in 1. A representation of each mesh is done in C.1 for element of degree $p = 2$ and 3.

Element	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Cub.	2	12	26
Basic.	2	8	18
Bern.	2	8	18

Table 1: *X* mesh: Summary table of number of modes per systems.

3.3 The fully discrete analysis

We analyze now the fully discrete schemes obtained using the RK, SSPRK and DeC time marching methods. Let us consider as an example the SSPRK schemes. If we define as $A := \mathbb{M}^{-1}(a_x\mathcal{K}_x + a_y\mathcal{K}_y + \delta\mathbb{S})$ we can write the schemes as follows

$$\begin{cases} \mathbf{U}^{(0)} := & \mathbf{U}^n \\ \mathbf{U}^{(s)} := & \sum_{j=0}^{s-1} \left(\gamma_{sj}\mathbf{U}^{(j)} + \Delta t\mu_{sj}A\mathbf{U}^{(j)}\right), \quad s \in [\![1, S]\!], \\ \mathbf{U}^{n+1} := & \mathbf{U}^{(S)}. \end{cases} \tag{42}$$

Expanding all the stages, we can obtain the following representation of the final stage:

$$\mathbf{U}^{n+1} = \mathbf{U}^{(0)} + \sum_{j=1}^{S} \nu_j \Delta t^j A^j \mathbf{U}^{(0)} = \left(\mathcal{I} + \sum_{j=1}^{S} \nu_j \Delta t^j A^j\right) \mathbf{U}^n, \tag{43}$$

where coefficients ν_j in Equation (43) are obtained as combination of coefficient γ_{sj} and μ_{sj} in Equation (42) and $\mathcal{I}$ is the identity matrix. For example, coefficients of the fourth order of accuracy scheme RK4 are $\nu_1 = 1$, $\nu_2 = 1/2$, $\nu_3 = 1/6$ and $\nu_4 = 1/24$.

We can now compress the problem proceeding as in the time continuous case. In particular, using Equations (40) one easily shows that the problem can be written in terms of the local $d \times d$ matrices $\widetilde{A} := \widetilde{\mathbb{M}}^{-1}\left(a_x\widetilde{\mathcal{K}_x} + a_y\widetilde{\mathcal{K}_y} + \delta\widetilde{\mathbb{S}}\right)$ and in particular that

$$\widetilde{\mathbf{U}}^{n+1} = G\widetilde{\mathbf{U}}^n \quad \text{with} \quad G := \left(\widetilde{\mathcal{I}} + \sum_{j=1}^{S} \nu_j \Delta t^j \widetilde{A}^j\right) = e^{\epsilon\Delta t}e^{-i\omega\Delta t}, \tag{44}$$

where $G \in \mathbb{R}^{d\times d}$ is the amplification matrix depending on θ, δ, Δt, Δx and Δy. Considering each eigenvalue λ_i of G, we can write the following formulae for the corresponding phase ω_i and damping coefficient ϵ_i

$$\begin{cases} e^{\epsilon_i\Delta t}\cos(\omega_i\Delta t) = \mathrm{Re}(\lambda_i), \\ -e^{\epsilon_i\Delta t}\sin(\omega_i\Delta t) = \mathrm{Im}(\lambda_i), \end{cases} \Leftrightarrow \begin{cases} \omega_i\Delta t = \arctan\left(\frac{-\mathrm{Im}(\lambda_i)}{\mathrm{Re}(\lambda_i)}\right), \\ (e^{\epsilon_i\Delta t})^2 = \mathrm{Re}(\lambda)^2 + \mathrm{Im}(\lambda)^2, \end{cases} \Leftrightarrow \begin{cases} \frac{\omega_i}{k} = \arctan\left(\frac{-\mathrm{Im}(\lambda_i)}{\mathrm{Re}(\lambda_i)}\right)\frac{1}{k\Delta t}, \\ \epsilon_i = \log\left(|\lambda_i|\right)\frac{1}{\Delta t}. \end{cases}$$

For the DeC method we can proceed with the same analysis transforming also the other involved matrices into their Fourier equivalent ones. Using Equation (30) these terms would contribute to the construction of G not only in the $\widetilde{A}$ matrix, but also in the coefficients ν_j, which become matrices as well. At the end we just study the final matrix G and its eigenstructure, whatever process was needed to build it up.

The matrix G describes one timestep evolution of the Fourier modes for all the d different types of degrees of freedom. The damping coefficients ϵ_i tell if the modes are increasing or decreasing in amplitude and the phase coefficients ω_i describe the phases of such modes.

We remark that a necessary condition for stability of the scheme is that $|\lambda_i| \leq 1$ or, equivalently, $\epsilon_i \leq 0$ for all the eigenvalues. The goal of our study is to find the largest CFL number for which the stability condition is fulfilled and such that the dispersion error is *not too large*.

For our analysis, we focus on the *X* type triangular mesh in Figure 3 with elements of degree 1, 2 and 3. This *X* type triangular mesh is also used in [39] for Fourier analysis of the acoustic wave propagation system.

3.4 Methodology

The methodology we explain in the following, will be applied to all the combination of schemes we presented above (in time: RK, SSPRK and DeC, discretisation in space: *Basic*, *Cubature* and *Bernstein*, stabilization techniques: CIP, OSS and SUPG), in order to find the best coefficients (CFL, δ), as in [42].

It must be remarked that the dispersion analysis must satisfy the Nyquist stability criterion, i.e., $\Delta x_{max} \leq \frac{L}{2}$ with Δx_{max} the maximal distance between two nodes on edges. In other words, $k_{max} = \frac{2\pi}{L_{min}} = \frac{2\pi}{2\Delta x_{max}} = \frac{\pi}{\Delta x_{max}}$. This tells us where k should vary, i.e., $k \in]0, \pi/\Delta x_{max}]$.

What we aim to do is an optimization process also on the stabilization parameter and the CFL number. With the notation of [42], we will set for the different stabilizations

$$\text{OSS}: \ \tau_K = \delta\Delta x|a|,$$

$$\text{CIP}: \ \tau_f = \delta\Delta x^2|a|,$$

$$\text{SUPG}: \ \tau_K = \delta\Delta x/|a|.$$

One of our objectives is to explore the space of parameters (CFL,δ), and to propose criteria allowing to set these parameters to provide the most stable, least dispersive and least expensive methods. A clear and natural criterion is

to exclude all parameter values for which there exists at least a wavenumber θ or an angle $\Phi \in [0, 2\pi]$ such that we obtain an amplification of the mode, i.e., $\epsilon(\theta) > 10^{-12}$ (taking into account the machine precision errors that might occur). Doing so, we obtain what we will denote as *stable area* in (CFL, θ) space. For all the other points we propose 3 strategies to minimize a combination of dispersion error and computational cost.

In the following we describe the strategy we adopt to find the best parameters couple (CFL,δ) that minimizes a global solution error, denoted by η_u, while maximizing the CFL in the stable area. In particular, we start from the relative square error of u

$$\left|\frac{u(t) - u_{ex}(t)}{u_{ex}(t)}\right|^2 = \left|e^{\epsilon t - it(\omega - \omega_{ex})} - 1\right|^2 \tag{45}$$

$$= \left[e^{\epsilon t}\cos(t(\omega - \omega_{ex})) - 1\right]^2 + \left[e^{\epsilon t}\sin(t(\omega - \omega_{ex}))\right]^2 \tag{46}$$

$$= e^{2\epsilon t} - 2e^{\epsilon t}\cos(t(\omega - \omega_{ex})) + 1. \tag{47}$$

Here, we denote with ϵ and ω the damping and phase of the *principal* mode and with $\omega_{ex} = \mathbf{k}\cdot\mathbf{a}$ the exact phase. For a small enough dispersion error $|\omega - \omega_{ex}| \ll 1$, we can expand the cosine in the previous formula in a truncated Taylor series as

$$\left|\frac{u(t) - u_{ex}(t)}{u_{ex}(t)}\right|^2 \approx \underbrace{\left[e^{\epsilon t} - 1\right]^2}_{\text{Damping error}} + \underbrace{e^{\epsilon t} t^2 \left[\omega - \omega_{ex}\right]^2}_{\text{Dispersion error}}. \tag{48}$$

We then compute an error at the final time $T = 1$, over the whole phase domain, using at least 3 points per wave $0 \le k\Delta x_p \le \frac{2\pi}{3}$, with $\Delta x_p = \frac{\Delta x}{p}$, and p the degree of the polynomials. We obtain the following $\mathbb{L}_2$ error definition,

$$\eta_u(\omega, \epsilon)^2 := \frac{3}{2\pi}\left[\int_0^{\frac{2\pi}{3}} (e^{\epsilon} - 1)^2 dk + \int_0^{\frac{2\pi}{3}} e^{\epsilon}(\omega - \omega_{ex})^2 dk\right]. \tag{49}$$

Recalling that $\epsilon = \epsilon(k\Delta x, \text{CFL}, \delta, \Phi)$ and $\omega = \omega(k, \Delta x, \text{CFL}, \delta, \Phi)$, we need to further set the parameter Δx_p. We choose it to be large $\Delta x_p = 1$, with the hope that for finer grids the error will be smaller. Moreover, we need to check that the stability condition holds for all the possible angles $\Phi \in [0, 2\pi]$.

Finally, we seek for the couple (CFL^*, δ^*) such that

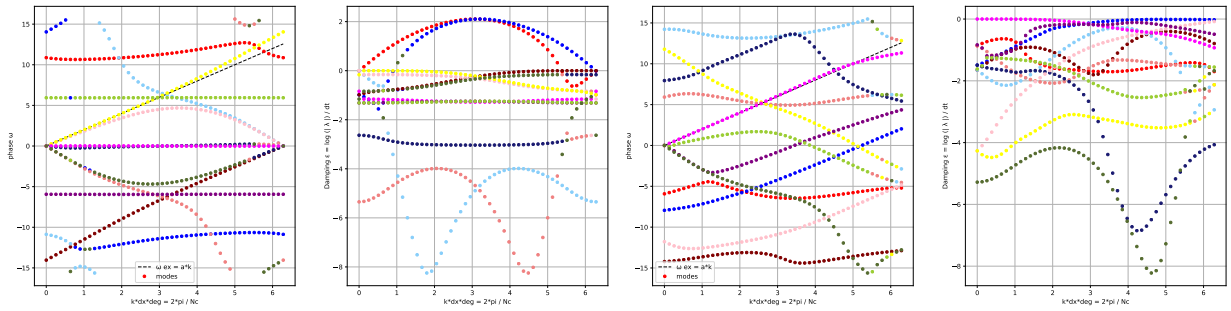
$$(\text{CFL}^*, \delta^*) = \arg\max_{\text{CFL}} \left\{ \eta(\omega, \epsilon, \Phi') < \mu \min_{\text{stable (CFL},\delta)} \max_{\Phi} \eta(\omega, \epsilon, \Phi), \quad \forall\, \Phi' \in [0, 2\pi] \right\}, \tag{50}$$

where the dependence on Φ of η is highlighted with an abuse of notation. For this strategy, the parameter μ must be chosen in order to balance the requirements on stability and accuracy. After having tried different values, we have set μ to 10 providing a sufficient flexibility to obtain results of practical usefulness. Indeed, the found values will be tested in the numerical section.

To show the influence of the angle Φ on the optimization problem we show an example for the *X* mesh. For a given couple of parameters (CFL,δ) = $(0.4, 0.01)$ we compare the results for $\Phi = 0$ and $\Phi = 3\pi/16$. In Figure 6 we compare the phases ω_i and the damping coefficients ϵ_i for the two angles. It is clear that for the angle $\Phi = 0$, on the left, there are some modes which are not stable $\epsilon_i > 0$, while for $\Phi = 3\pi/16$ all modes are stable.

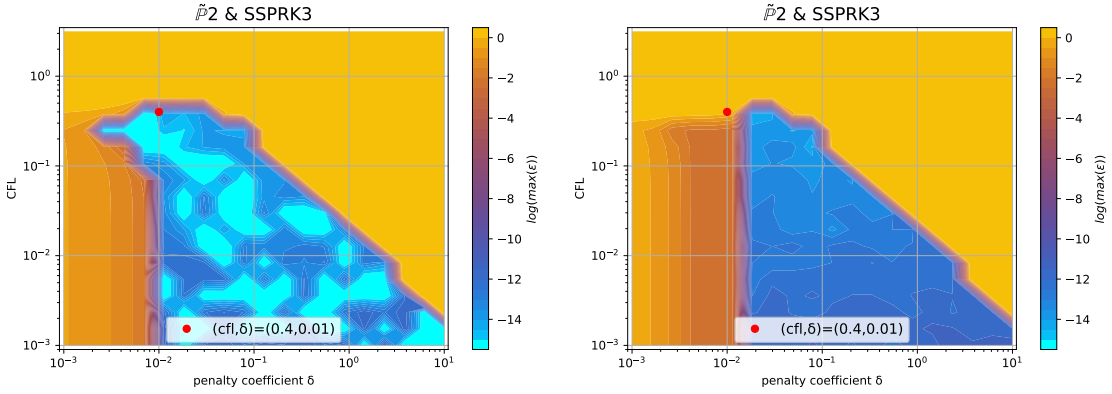
The angle can widely influence the whole analysis as one can observe in the plot of $\max_i \epsilon_i$ in Figure 7, where we observe that for the only angle $\Phi = 3\pi/16$ we would obtain an optimal parameter in (CFL,δ) = $(0.4, 0.01)$, while, using all angles, this value is not stable anymore.

Remark 2 *To define the stable region, we should only consider configurations for which the damping is below machine accuracy. In practice, this cannot be done due to the fact that the eigenvalue problem arising from* (44) *is only solved approximately using the linear algebra package of* `numpy`*. This introduces some uncertainty in the definition of the stability region as machine accuracy needs to be replaced by some other finite threshold.*



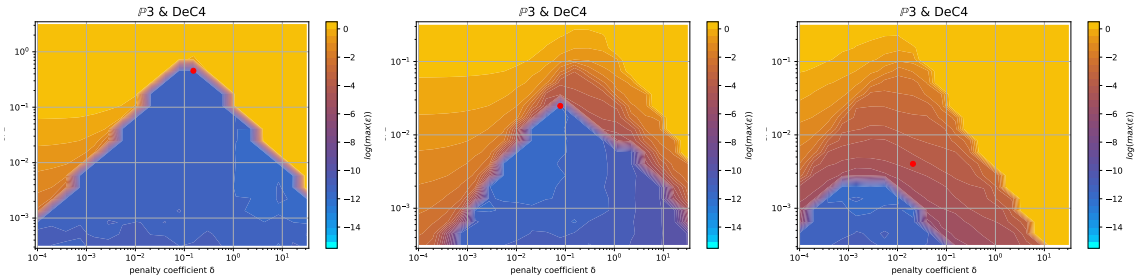
(a) $\Phi = 0$ (b) $\Phi = 3\pi/16$

Figure 6: Comparison of dispersion curves ω_i and damping coefficients ϵ_i, for *Cubature* $\tilde{\mathbb{P}}_2$ elements, with SSPRK time discretization and OSS stabilization. $\Phi = 0$ at the left and $\Phi = 3\pi/16$ at the right.



(a) $\Phi = 0$ (b) $\forall \Phi \in [0, 2\pi]$

Figure 7: Plot of $\log(\max_i \epsilon_i)$ for *Cubature* $\tilde{\mathbb{P}}_2$ elements, SSPRK time discretization and OSS stabilization. The blue and light blue region is the stable one. At the left only for $\Phi = 3\pi/16$, at the right we plot the maximum over all Φ.



(a) SUPG (b) OSS (c) CIP

Figure 8: Damping coefficients $\log(\max_i \epsilon_i)$ for $\mathbb{B}_3$ *Bernstein* elements and the DeC method with, from left to right, SUPG, OSS and CIP stabilization. The red dot is the optimum according to (50).

3.5 Results of the fourier analysis using the *X* type mesh

In this section, we illustrate the result obtained with the methodology explained above. For clarity not all the results are reported in this work, however we place all the plots for all possible combination of schemes in an online repository [42]. We will provide some examples here and a summary of the main results that we obtained.

The first type of plot we introduce is helping us in understanding how we can define the stability region in the (CFL, δ) plane. So, for every (CFL, δ) we plot the maximum of $\log(\epsilon_i)$ over all modes and angles $\Phi \in [0, 2\pi]$ (thanks to the symmetry of the mesh we can reduce this interval). An example is given in the right plot of Figure 7, it is clear that the whole blue area is stable and the yellow/orange area is unstable. In other cases, this boundary is not so clear and setting a threshold to determine the stable area can be challenging. In Figure 8 we compare different stabilizations for DeC with $\mathbb{B}_3$ elements. In the CIP stabilization case, we clearly see that there is no clear discontinuity between unstable values and stable ones, as in SUPG, because there is a transient region where $\max_i \epsilon_i$ varies between 10^{-7} and 10^{-4}.

The second type of plot combines the chosen stability region with the error η_u. We plot on the (CFL, δ) plane some black crosses on the unstable region, where there exists an i and Φ such that $\epsilon_i > 10^{-7}$. The color represents $\log(\eta_u)$ and the best value according to the previously described method is marked with a red dot. In Figures 9 to 12 we show some examples of these plots for some schemes, for different $p = 1, 2, 3$. In Figures 9 and 10 we test the *Basic* elements with the SSPRK time discretization, while in Figures 11 and 12 we use the *Cubature* elements with DeC time discretization. We compare also different stabilization technique: in Figures 9 and 11 we use the OSS, while in Figures 10 and 12 the CIP. One can observe many differences among the schemes. For instance, for $p = 3$ we see a much wider stable area for SSPRK than with DeC and, in the *Cubature* DeC case, we see that the CIP requires a reduction in the CFL number with respect to the OSS stabilization.

We summarize the results obtained by the optimization strategy in Table 2 for all the combinations of spatial, time and stabilization discretization. The CFL and δ presented there are optimal values obtained by the process above described, which we aim to use in simulations to obtain stable and efficient schemes. Unfortunately, as already mentioned above, for some schemes the stability area is not so well defined for several reasons. One of these reasons is the *"shape"* of the stability area as for one-dimensional problems, see [42]. Other issues that affect this analysis are the numerical precision, see Section 3.6, and the mesh configuration, see Section 3.7. In the following we study more in details these cases and how one can find better values.

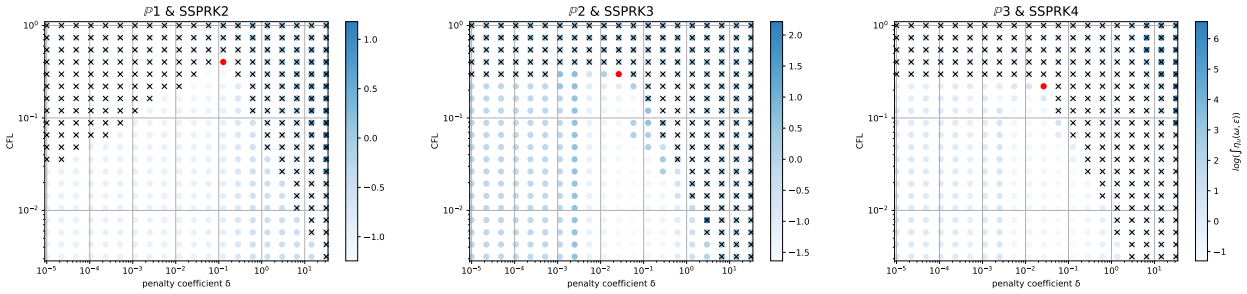


Figure 9: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ *Basic* elements with SSPRK scheme and OSS stabilization

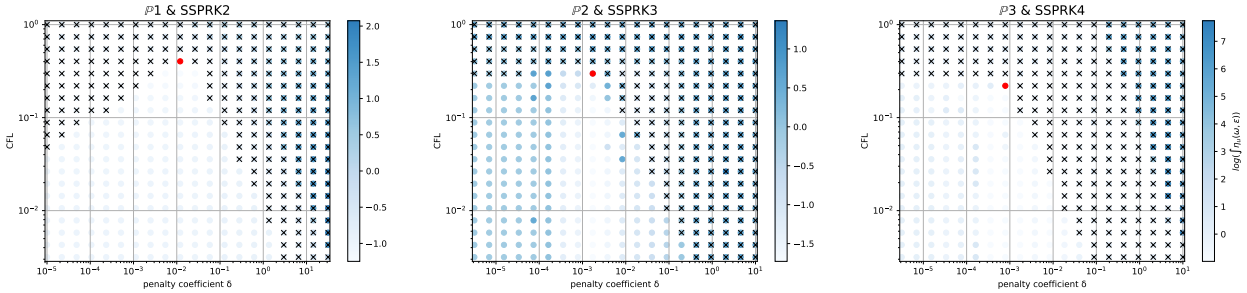


Figure 10: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ *Basic* elements with SSPRK scheme and CIP stabilization

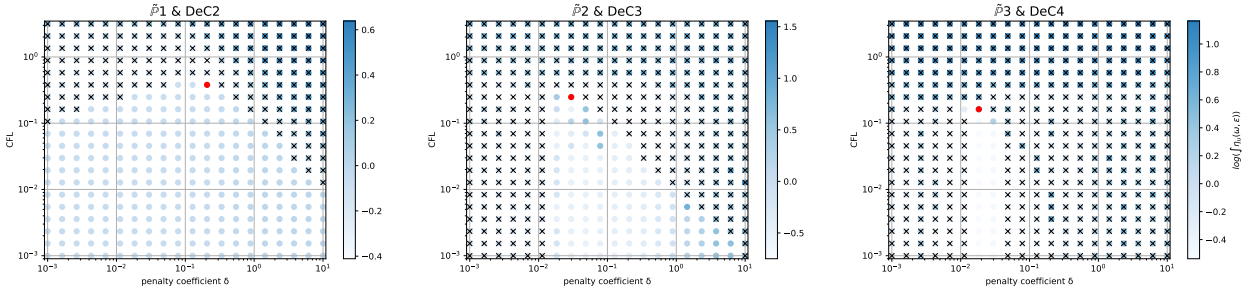


Figure 11: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ *Cubature* elements with DeC scheme and OSS stabilization

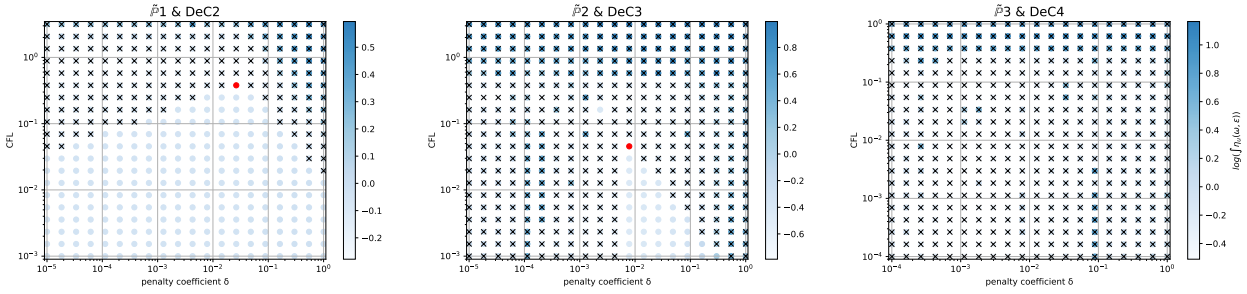


Figure 12: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ *Cubature* elements with DeC scheme and CIP stabilization

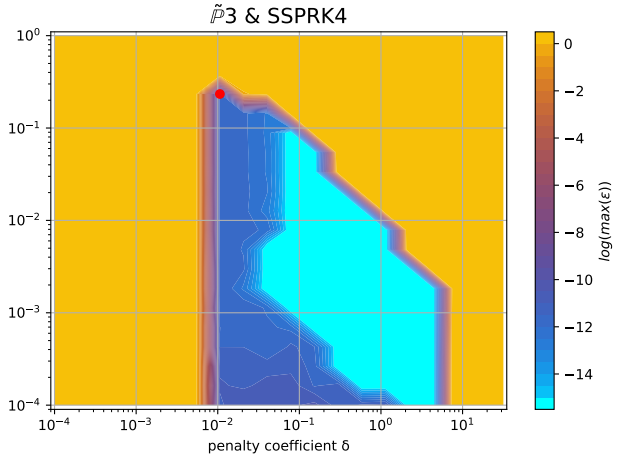


Figure 13: Logarithm of the amplification coefficient $\log(\max_i(\varepsilon_i))$ for SUPG stabilization with $\tilde{\mathbb{P}}_3$ *Cubature* elements and the SSPRK method. Unstable region in yellow, the red dot is the optimal parameter according to (50)

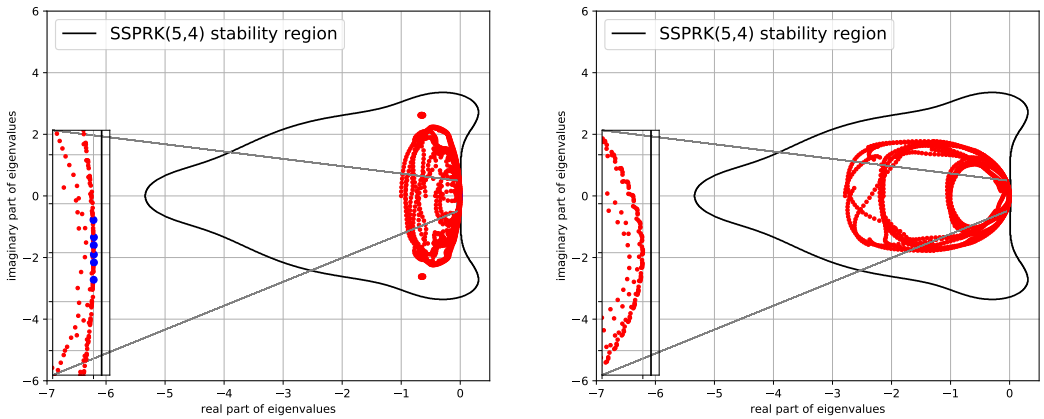
Element &		SUPG		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	0.739 (0.127)	0.298 (0.058)	0.22 (0.026)
	RK	0.403 (0.127)	0.298 (0.026)	0.22 (5.46e-03)
Cub.	DeC	0.616 (0.28)	0.234 (0.04)*	0.144 (0.04)
	SSPRK	1.062 (0.28)	0.379 (0.021)*	0.234 (0.011)*
	RK	0.616 (0.28)	0.234 (0.04)	0.144 (0.04)
Bern.	DeC	0.739 (0.298)	0.455 (0.298)*	0.455 (0.153)*
	SSPRK	0.739 (0.127)	0.298 (0.058)	0.22 (0.026)
	RK	0.403 (0.127)	0.298 (0.026)	0.22 (5.46e-03)

Element &		OSS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	0.403 (0.127)	0.298 (0.026)	0.22 (0.026)	0.403 (0.012)	0.298 (1.73e-03)	0.22 (7.85e-04)*
	RK	0.22 (0.058)	0.22 (0.026)	0.22 (0.012)	0.298 (0.012)	0.22 (1.73e-03)	0.22 (3.57e-04)
Cub.	DeC	0.379 (0.207)	0.248 (0.03)	0.162 (0.018)	0.379 (0.026)	0.045 (7.85e-03)*	/
	SSPRK	0.58 (0.336)	0.379 (0.03)	0.248 (0.018)	0.58 (0.048)	0.07 (7.85e-03)*	/
	RK	0.379 (0.207)	0.248 (0.03)	0.162 (0.018)	0.379 (0.026)	0.045 (7.85e-03)	/
Bern.	DeC	0.173 (0.58)	0.036 (0.298)	0.025 (0.078)*	0.173 (0.153)	0.012 (0.021)	0.004 (0.021)*
	SSPRK	0.403 (0.127)	0.298 (0.026)	0.22 (0.026)	0.403 (0.012)	0.298 (1.73e-03)	0.22 (7.85e-04)
	RK	0.22 (0.058)	0.22 (0.026)	0.22 (0.012)	0.298 (0.012)	0.22 (1.73e-03)	0.22 (3.57e-04)

Table 2: *X* mesh: Optimized CFL and penalty coefficient δ in parenthesis, minimizing η_u.
"/" means that the fourier analysis shown that the scheme is unstable.
* These values are not reliable, see Section 3.6.

3.6 Comparison with a space-time split stability analysis

In this section, we show another stability analysis to slightly improve the results obtained above. Indeed, the solution of the eigenvalue problem (44) is only obtained within some approximation from the `numpy` numerical library. In some cases, the threshold used to define the stability region is defined in a somewhat heuristic manner. So to confirm the results, we use independently another criterion. To this end we treat independently the temporal and



(a) (CFL,δ)=$(0.234, 0.011)$ (b) (CFL,δ)=$(0.18, 0.04)$

Figure 14: Eigenvalues of $\widetilde{A}$ using cubature discretization and the SUPG stabilization (varying k) and stability area of the SSPRK method. In red the stable eigenvalues, in blue the unstable ones.

spatial discretizations as in the method of lines. We then study only the spectral properties of the spatial discretization alone, computing the eigenvalues of the corresponding matrix A (cf. (42)). With this information, we then check whether they belong to the stability area of the time discretization.

In particular, following [21], we write the time discretization for Dahlquist's equation

$$\partial_t u - \lambda u = 0, \tag{51}$$

in this example, we consider the SSPRK discretization (42). From Equation (43) we can write the amplification coefficient $\Gamma(\lambda)$, i.e.,

$$\mathbf{U}^{n+1} = \mathbf{U}^{(0)} + \sum_{j=1}^{S} \nu_j \Delta t^j \lambda^j \mathbf{U}^{(0)} = \underbrace{\left(\mathcal{I} + \sum_{j=1}^{S} \nu_j \Delta t^j \lambda^j \right)}_{\Gamma(\lambda)} \mathbf{U}^n. \tag{52}$$

The stability condition for this SSPRK scheme is given by $\Gamma(\lambda) \leq 1$. Now, when we substitute the Fourier transform of the spatial semidiscretization $\widetilde{A}$ to the coefficient λ and we diagonalize the system (or we put it in Jordan's form), we obtain a condition on the eigenvalues of $\widetilde{A}$. Then, using the parameters provided by the previous analysis (CFL,δ)=$(0.234, 0.011)$, in Table 2, we plot the eigenvalues of $\widetilde{A}$ and the stability region of the SSPRK scheme for different $\theta \in [0, \pi]$. We notice that for some values of θ some of the eigenvalues fall slightly outside the stable area, see Figure 14(a). There are, indeed, few eigenvalues dangerously close to the imaginary axis and some of them have actually positive real part (blue dots). As suggested before, if we decrease the CFL and increase δ, we move towards a safer region, so considering (CFL,δ)=$(0.18, 0.04)$ with the same θ, we obtain all stable eigenvalues, as shown in Figure 14(b).

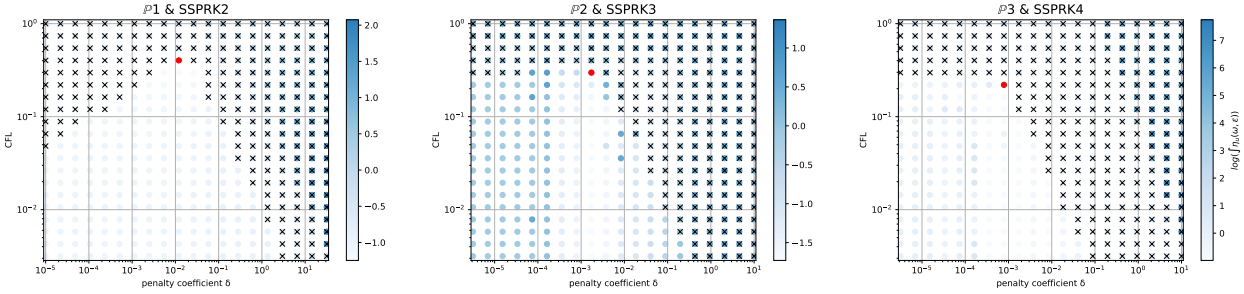
The summary of the optimal parameters of Table 2 updated taking into account also a larger safety region in the (CFL, δ) plane (as explained in this section) can be found in Table 15 in Appendix C.2.

3.7 Different mesh patterns

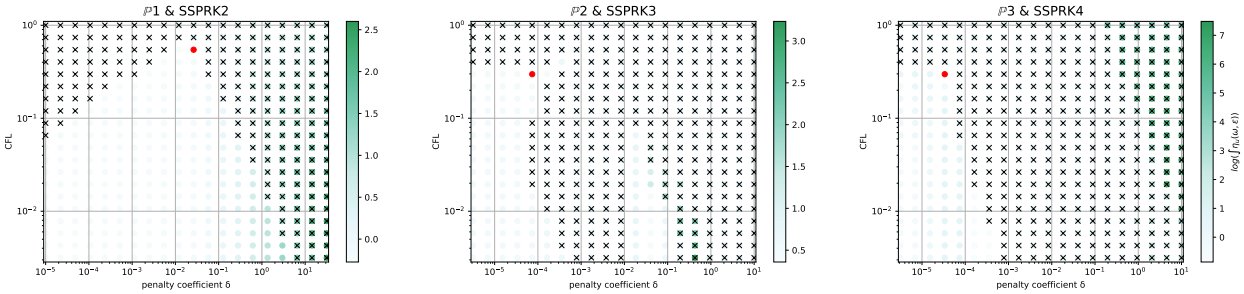
Another important aspect about this stability analysis is the influence of the mesh structure on the results. As an example, we use the *T*-mesh, another regular and structured mesh type depicted in Figure 4. In Figure 4 we plot also the degrees of freedom for elements of degree 2 and the periodic elementary unit that we take into consideration for

Element	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Cub.	1	6	13
Basic.	1	4	9
Bern.	1	4	9

Table 3: Number of modes in the periodic unit for different elements in the T mesh.



(a) X mesh



(b) T mesh

Figure 15: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ *Basic* elements with SSPRK scheme and CIP stabilization

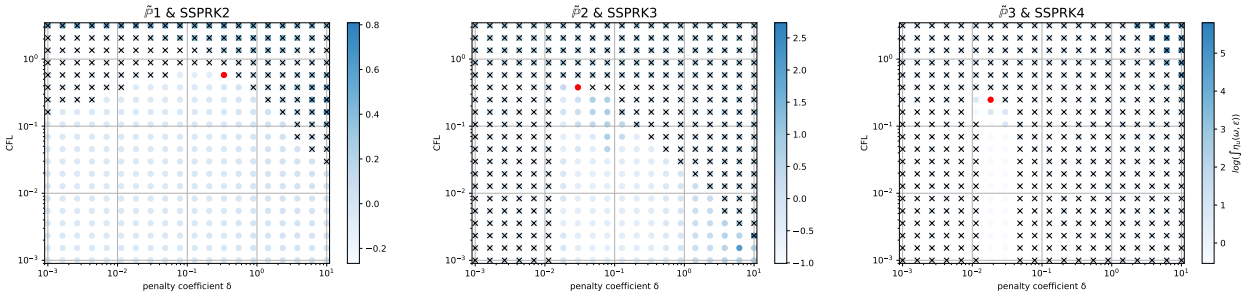
the Fourier analysis. The number of modes in the periodic unit for this mesh type are summarized in Table 3. The elements of degree 3 can be found in Figure 28 in Appendix C.1.

Even if for several methods we observe comparable results for the two mesh types, for some of them the analyses are quite different. An example is given by the *Basic* elements with SSPRK schemes and CIP stabilization. For this method, we plot the dispersion error (49) and the stability area in Figure 15(a) for the X mesh and in Figure 15(b) for the T mesh. We see huge differences in $\mathbb{P}_2$ and $\mathbb{P}_3$ where in the former a wide region becomes unstable for $\delta_L \leq \delta \leq \delta_R$ and for the latter we have to decrease a lot the value of δ to obtain stable schemes.

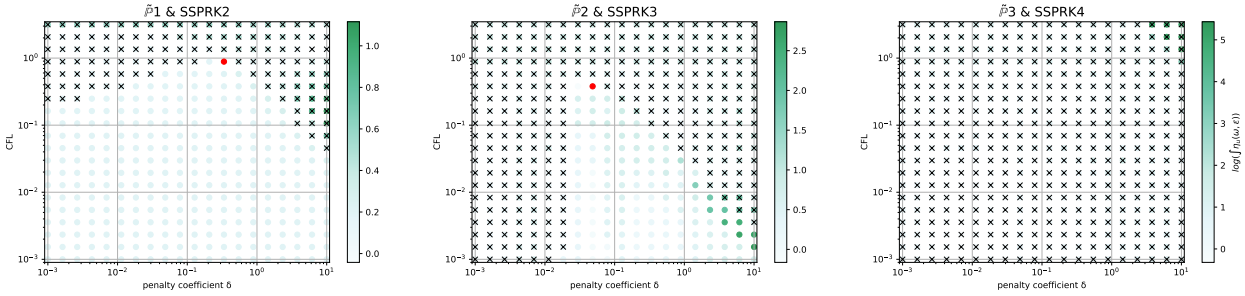
In the case of *Cubature* elements with the OSS stabilization and SSPRK time integration, we have already seen in the previous section that the optimal parameters found were in a dangerous area. Repeating the stability analysis for the T mesh we see that the situation is even more complicated. In Figure 16(a) we plot the analysis for the X mesh and in Figure 16(b) the one for the T mesh. $\tilde{\mathbb{P}}_3$ elements, though being stable for some parameters for the X mesh, are never stable on the T mesh. This means, that, when searching general parameters for the schemes, we have to keep in mind that different meshes leads to different results.

For completeness, we present the optimal parameters also for the T mesh in Table 16 in Appendix C.2.

In general, it is important to consider more mesh types when doing this analysis. In practice, we will use the two

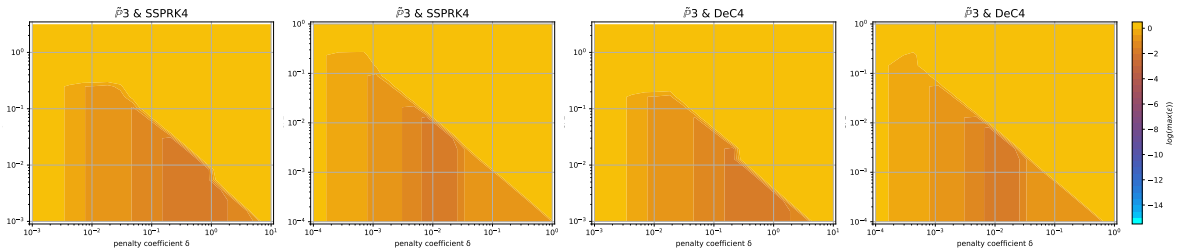


(a) X mesh



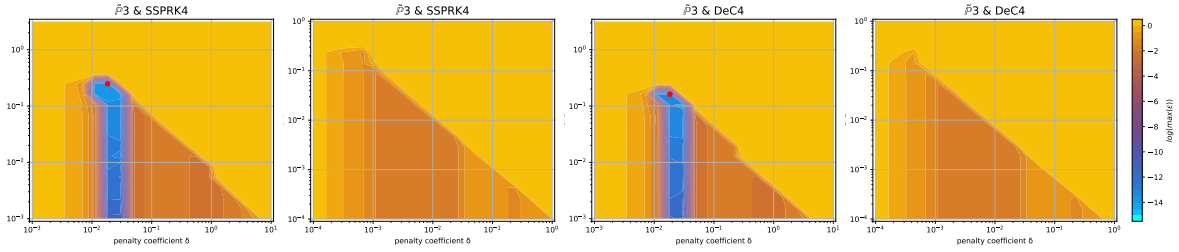
(b) T mesh

Figure 16: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ *Cubature* elements with SSPRK scheme and OSS stabilization



(a) SSPRK with OSS (b) SSPRK with CIP (c) DeC with OSS (d) DeC with CIP

Figure 17: Maximum logarithm of the amplification coefficient $\log(\max_i(\varepsilon_i))$ for $\tilde{\mathbb{P}}_3$ *Cubature* elements on the *X* and *T* meshes



(a) SSPRK with OSS (b) SSPRK with CIP (c) DeC with OSS (d) DeC with CIP

Figure 18: Logarithm of the amplification coefficient $\log(\max_i(\varepsilon_i))$ for $\tilde{\mathbb{P}}_3$ *Cubature* elements on the *X* mesh

presented above (*X* and *T* meshes). In the following, we will consider the stability region as the intersection of stability regions of both meshes.

3.8 Final results of the stability analysis

Taking into consideration all the aspects seen in the previous sections, it is important to have a comprehensive result, which tells which parameters can be used in the majority of the situations. A summary of the parameters obtained for the *X* and *T* mesh is available in Appendix C.2. In Table 4, instead, we present parameters obtained using the most restrictive case among different meshes and that insure an enough big area of stability around them, as explained in Section 3.6. These parameters can be safely used in many cases and we will validate them in the numerical sections, where, first, we validate the results of the *X* mesh on a linear problem on an *X* mesh, then we used the more general parameters in Table 4 for nonlinear problems on unstructured meshes.

A special remark must be done for *Cubature* $\tilde{\mathbb{P}}_3$ elements combined with the OSS and the CIP stabilizations. In Figure 17 we see how the amplification coefficient $\max_i \varepsilon_i$ has always values far away from zero. For the CIP stabilization this is always true and even for the $\tilde{\mathbb{P}}_2$ elements the stability region is very thin. As suggested in [17, 38] higher order derivatives jump stabilization terms might fix this problem, but it introduces more parameters. This has not been considered here. Another remark is that the *T* configuration is very peculiar and, as we will see, on classical Delauney triangulations the issue seem to not affect the results. Moreover, the use of additional discontinuity capturing operators may alleviate this issue as some additional, albeit small, dissipation is explicitly introduced in smooth regions.

In Section 3.9, we propose to add an additional stabilization term for these unstable schemes, i.e., *Cubature* $\tilde{\mathbb{P}}_3$

Element &		SUPG		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	0.739 (0.127)	0.2 (0.1)*	0.22 (0.026)
Cub.	SSPRK	1.062 (0.28)	0.12 (0.13)*	0.09 (0.05)*
	DeC	0.616 (0.28)	0.144 (0.078)	0.05 (0.05)*
Bern.	DeC	0.739 (0.298)	0.12 (0.45)*	0.2 (0.153)*

Element &		OSS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	0.403 (0.127)	0.2 (0.05)*	0.22 (0.026)	0.403 (0.012)	0.1 (1.00e-03)*	0.1 (5.00e-04)*
Cub.	SSPRK	0.58 (0.336)	0.2 (0.08)*	0.28 (0.018)**	0.58 (0.048)	0.06 (0.01)*	/
	DeC	0.379 (0.207)	0.12 (0.07)*	0.162 (0.018)**	0.379 (0.026)	0.025 (0.01)*	/
Bern.	DeC	0.173 (0.58)	0.02 (0.2)*	0.015 (0.078)*	0.173 (0.153)	0.012 (0.01)*	0.001 (0.01)*

Table 4: Optimized CFL and penalty coefficient δ in parenthesis, combining the two mesh configurations. The values denoted by * are not the optimal one, but they lay in a safer region, see Section 3.6. The values marked by ** cannot be used on the *T* mesh. "/" means that it is unstable for every parameter.

elements and OSS or CIP stabilization techniques. This term is based on viscous term [2, 30, 36, 41] and allows to stabilize numerical schemes for any mesh configuration.

For the OSS stabilization we observe a similar behavior in Figure 17. The stability that we see in that plot are only due to the the T mesh. Indeed, for the OSS stabilization on the *X* mesh there exists a corridor of stable values, which turn out to be unstable for the *T* mesh, see Figure 18. In practice, also on unstructured grids we have not noticed instabilities when running with the parameters found with the *X* mesh. Hence, we suggest anyway some values of CFL and δ for these schemes, which are valid for the *X* mesh, noting that they might be dangerous for very simple structured meshes. The validation on unstructured meshes also for more complicated problems will be done in the next sections.

Overall, Table 4 gives some insight on the efficiency of the schemes. We remind that, in general, we prefer matrix free schemes, so this aspect must be kept in mind while evaluating the efficiency of the schemes. All the SUPG schemes, except when with DeC, and all the *Basic* element schemes have a mass matrix that must be inverted. Among the others we see that for first degree polynomials schemes the DeC with *Bernstein* polynomials and SUPG stabilization gives one of the largest CFL result, while for second degree polynomials the OSS *Cubature* SSPRK scheme seems the one with best performance and, for fourth order schemes, again the *Bernstein* DeC SUPG is one of the best.

In conclusion of this section, there are important points to highlight:

- The extension of the Fourier analysis to the two-dimensional space leads to significantly different results with respect to the one-dimensional one. Both in terms of global stability of the schemes, and in terms of optimal parameters. Moreover, in opposition to [42], *Bernstein* elements with SUPG stabilization technique lead to stable and efficient schemes. *Cubature* elements, which were the most efficient in one-dimensional problems, have stability issues on the two-dimensional mesh topologies studied.
- The complexity of the analysis in two-dimensional space is increased. This not only implies a larger number of degrees of freedom, but also more parameters to keep into account, including the angle of the advection term and the possible different configuration of the mesh. The visualization of the stability region of the time scheme as shown in Figure 14 with the eigenvalues of the semi-discretization operators helps in understanding the effect of CFL and penalty coefficient on the stability of the scheme, only for methods of lines. This helps in choosing and optimizing the couple of parameters.

Remark 3 *Another possibility to characterize the linear stability of numerical method is proposed by J. Miller [44].*

This method is based on the study of the characteristic polynomial of the amplification matrix G. However, this method does not provide information about the phase ω, since it does not compute eigenvalues of G. For this reason, we choose the eigenanalysis.

3.9 Accounting for discontinuity capturing corrections

The stabilization terms accounted for so far are linear stabilization operators. For more challenging simulations, additional non-linear stabilization techniques might be added to control the numerical solution in vicinity of strong non-linear fronts and/or discontinuities. We consider here the effect of adding an extra viscosity term, as in the entropy stabilization formulations proposed e.g. in [2, 35, 30, 36, 41]. We in particular look at the approach proposed in [30], and used for shallow water waves in [49, 41] and in [9, 28]. In this approach the viscosity is designed to provide a first order correction $\mu_K = \mathcal{O}(h)$ close to discontinuities, while for smooth enough solutions $\mu_K = ch^{p+1}$.

Our idea is to embed this high order correction explicitly in the analysis of the previous section to provide a heuristic characterization of the fully discrete stability of the resulting stabilized formulation: find $u_h \in V_h^p$ that satisfies for any $v_h \in W_h$

$$\int_\Omega v_h(\partial_t u_h + \nabla \cdot f(u_h))dx + \underbrace{S(v_h, u_h)}_{\text{Diffusive term}} + \underbrace{\sum_K \int_K \mu_K(u_h)\nabla v_h \cdot \nabla u_h}_{\text{Viscosity term}} = 0. \tag{53}$$

3.9.1 Note on the stability of the method

As it is done for previous stabilization terms in Section 2.1, we can characterize the accuracy of this method estimating the truncation error for a polynomial approximation of degree p. Considering the smooth exact solution $u^e(t,x)$ of (53), for all functions ψ of class at least $\mathcal{C}^1(\Omega)$ of which ψ_h denotes the finite element projection, we obtain

$$\begin{aligned}\epsilon(\psi_h) := \Big| \int_{\Omega_h} \psi_h \partial_t(u_h^e - u^e)\, dx - \int_{\Omega_h} \nabla\psi_h \cdot (f(u_h^e) - f(u^e))\, dx \\ + \sum_{K\in\Omega_h} \mu_K \int_K \nabla\psi_h \cdot \nabla(u_h^e - u^e)dx \Big| \leq Ch^{p+1},\end{aligned} \tag{54}$$

with C a constant independent of h. The estimate can be derived from standard approximation results applied to $u_h^e - u^e$ and to its derivatives, knowing that $\mu_K = \mathcal{O}(h^{p+1})$.

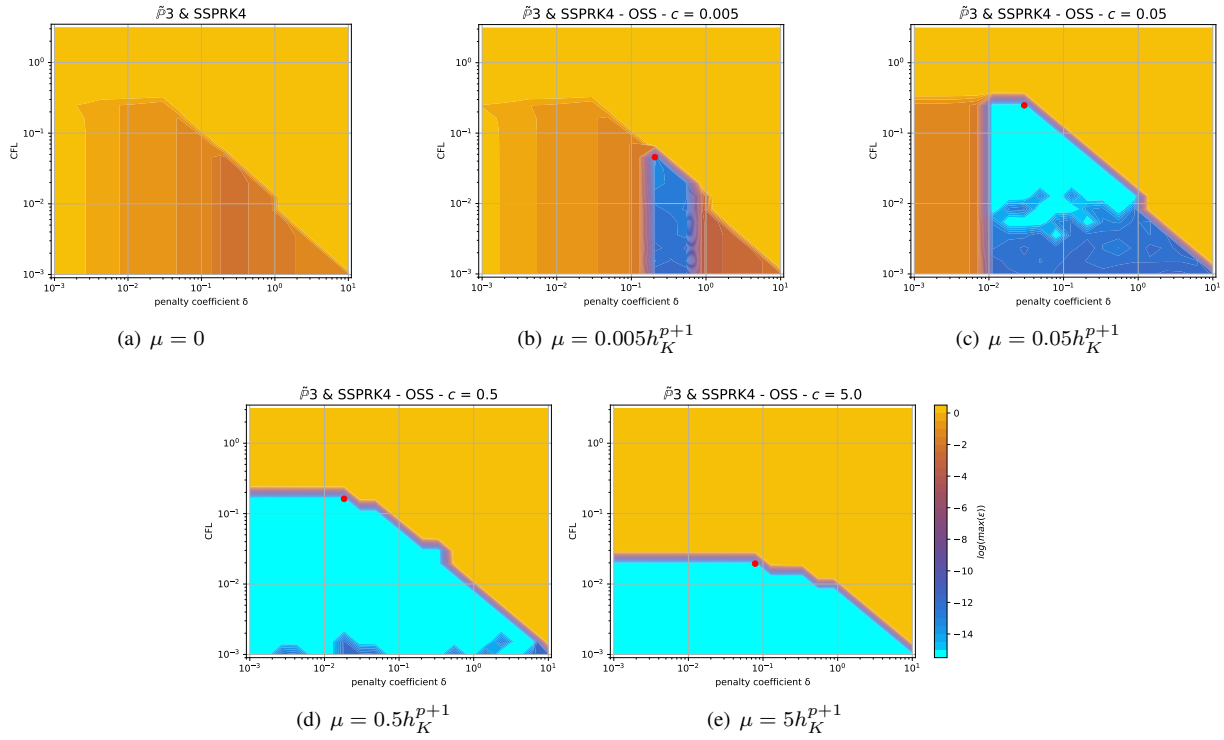
Then, for a linear flux, periodic boundaries and taking $\mu_K = \mu$ constant along the mesh, we can test with $v_h = u_h$ in (53), we get

$$\int_{\Omega_h} d_t \frac{u_h^2}{2} = -\sum_K \int_K \mu(\nabla u_h)^2 \leq 0, \tag{55}$$

which can be integrated in time to obtain a bound on the $\mathbb{L}_2$ norm of the solution.

3.9.2 The von Neumann analysis

As we saw in Section 3.8, the *T* mesh configuration has stability issues. In particular, the numerical schemes using *Cubature* $\tilde{\mathbb{P}}_3$ elements, SSPRK and DeC time integration methods, and the OSS and the CIP stabilization techniques are unstable. We propose to evaluate these schemes adding the viscosity term in (53). For the von Neumann analysis, we use $\mu_K(u) = ch_K^{p+1}$ in (53), with $c \in \mathbb{R}^+$, h_K the cell diameter and p the degree of polynomial approximation. We show the plot of $\max_i \epsilon_i$ to understand how the stability region behaves with respect to c using *Cubature* $\tilde{\mathbb{P}}_3$ elements. In Figure 19 the maximum amplification factor ϵ is represented for varying c, using the OSS stabilization technique and the SSPRK time integration method. We note that the same behaviour is observed with CIP and DeC. Plots are available online [43].



(a) $\mu = 0$ (b) $\mu = 0.005 h_K^{p+1}$ (c) $\mu = 0.05 h_K^{p+1}$

(d) $\mu = 0.5 h_K^{p+1}$ (e) $\mu = 5 h_K^{p+1}$

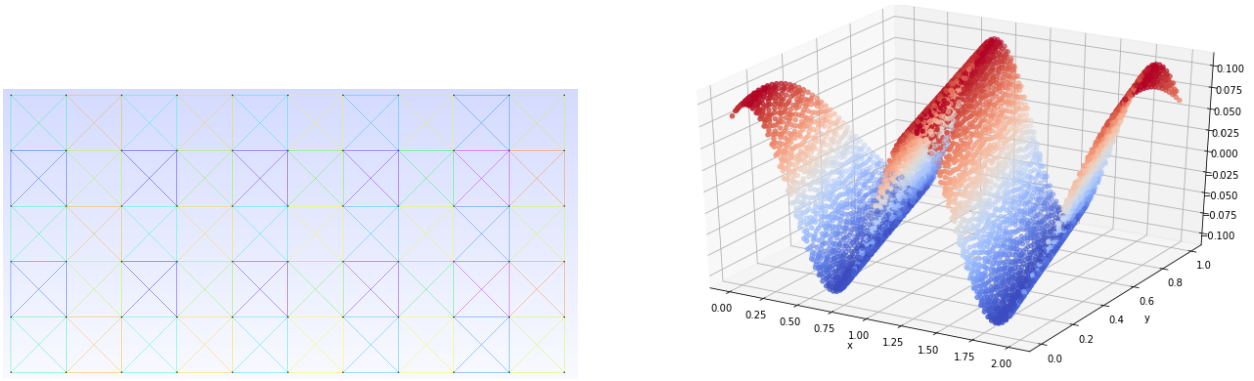
Figure 19: *T* mesh - Von Neumann analysis using an additional viscosity term (see (53)). *Cubature* $\tilde{\mathbb{P}}_3$ elements with SSPRK and OSS. Comparison of different μ.

We can observe two main results. First, increasing the parameter c up to around 0.1 allows to expand the stability region. Second, when the viscosity coefficients reaches too high values, it is necessary to decrease the CFL (see Figure 19(c) with $\mu = 0.05$ and Figure 19(d) with $\mu = 0.5$ as an example).

4 Numerical verification

We now perform numerical tests to check the validity of our theoretical findings. We initially focus on the structured grids, and in particular on the *X* mesh configuration, although similar verifications have been performed on the *T* mesh. We will use elements of degree p, with p up to 3, with time integration schemes of the corresponding order of accuracy to ensure an overall error of $\mathcal{O}(\Delta x^{p+1})$, under the CFL conditions discussed earlier (see also Table 15 in Appendix C.2). As already stressed, numerical integration is performed with Gauss–Legendre *formulae* of the appropriate order to exactly integrate the variational form for *Basic* and *Bernstein* elements, while for *Cubature* elements we use those associated to the interpolation points.

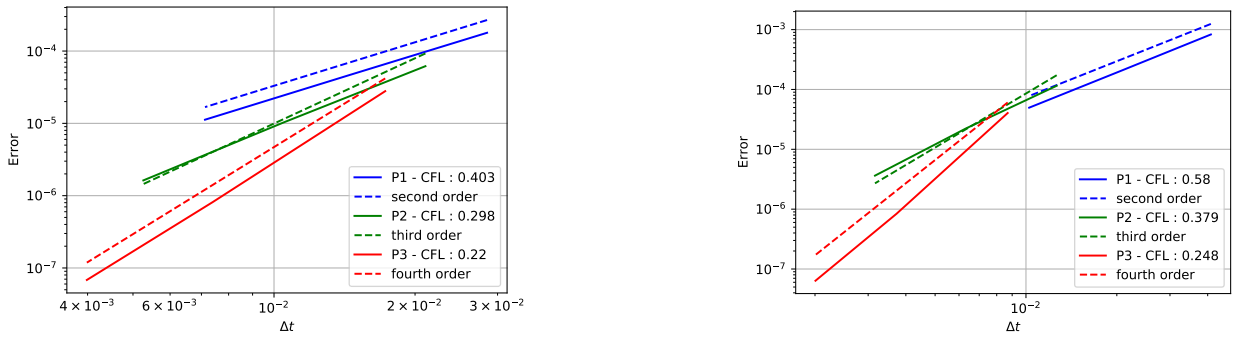
The mesh used in the Fourier analysis is the basis of the one we will use in the numerical simulations. We will extend it periodically for the whole domain, see an example in Figure 20(a).



(a) X mesh on $\Omega = (0, 2) \times (0, 1)$

(b) Cosinus test case with $\theta = 3\pi/16$

Figure 20: Linear advection simulation on the X mesh



(a) *Basic* elements

(b) *Cubature* elements

Figure 21: Error decay for linear advection problem with different elements and OSS stabilization and SSPRK time discretization: $\mathbb{P}_1$ in blue, $\mathbb{P}_2$ in green and $\mathbb{P}_3$ in red

4.1 Linear advection equation test

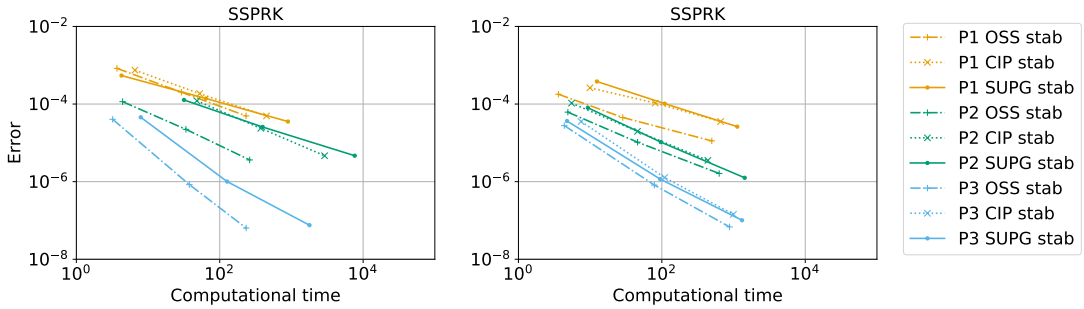
We start with the linear advection equation 1 on the domain $\Omega = [0, 2] \times [0, 1]$ using Dirichlet inlet boundary conditions:

$$\begin{cases} \partial_t u(t, \mathbf{x}) + \mathbf{a} \cdot \nabla u(t, \mathbf{x}) = 0, & (t, \mathbf{x}) \in [t_0, t_f] \times \Omega, \quad \mathbf{a} = (a_x, a_y)^T \in \mathbb{R}^2, \\ u(0, \mathbf{x}) = u_0(\mathbf{x}), & \\ u(t, \mathbf{x_D}) = u_{ex}(t, \mathbf{x_D}), & \mathbf{x_D} \in \Gamma_D = \{(x, y) \in \mathbb{R}^2, x \in \{0, 2\} \text{ or } y \in \{0, 1\}\}, \end{cases} \tag{56}$$

where $u_0((x, y)^T) = 0.1 \cos(2\pi\, r(x, y))$, with $r(x, y) = \cos(\theta)x + \sin(\theta)y$ the rotation by an angle θ around $(0, 0)$, $\mathbf{a} = (a_x, a_y)^T = (\cos(\theta), \sin(\theta))^T$ and $\theta = 3\pi/16$. The final time of the simulation is $t_f = 2s$.

The exact solution is $u_{ex}(\mathbf{x}, t) = u_0(x - a_x\, t, y - a_y\, t)$ for all $\mathbf{x} = (x, y) \in \Omega$ and $t \in \mathbb{R}^+$. The initial conditions are displayed in Figure 20(b). We discretize the domain with the X mesh pattern, see Figure 20(a). To have approximately the same number of degrees of freedom for different degrees p, we use different mesh sizes for each order of accuracy: $\Delta x_1 = \{0.1, 0.05, 0.025\}$ for $\mathbb{P}_1$, $\Delta x_2 = 2\Delta x_1$ for $\mathbb{P}_2$, and $\Delta x_3 = 3\Delta x_1$ for $\mathbb{P}_3$ elements.

A representative result is provided in Figures 21(a) and 21(b): it shows a comparison between *Cubature* and *Basic* elements with OSS stabilization and SSPRK time integration. As we can see, the two schemes have very similar



(a) *Cubature* elements

(b) *Basic* elements

Figure 22: Error for linear advection problem (56) with respect to computational time for SSPRK time discretization, comparing *Basic* and *Cubature* elements and all stabilization techniques

errors except for $\mathbb{P}^1$ where the larger CFL increases the error. The *Basic* elements require stricter CFL conditions, see Table 15, and have larger computational costs because of the inversion of the mass matrix.

To show the main benefit of using the *Cubature* elements (diagonal mass matrix), we plot in Figure 22 the computational time of *Basic* and *Cubature* elements for the SSPRK time scheme and all stabilization techniques. As a first interesting result of numerical test, looking at the Figure 22, we can clearly see that, for a fixed accuracy, *Cubature* elements obtain better computational times with respect to *Basic* elements. Moreover, as expected, the SUPG stabilization technique requires more computational time as it requires the inversion of a mass matrix, even in the case where the CFL used in is larger than the ones for OSS or CIP stabilization, see Table 15.

The order of accuracy reached by each simulations is shown in Table 5. The plots and all the errors are available at the repository [43].

Element &		SUPG			OSS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	1.93	2.96	4.02	2.0	2.62	4.1	1.44	2.45	3.77
Cub.	SSPRK	1.97	2.39	4.38	2.03	2.49	4.41	1.96	2.35	/
	DeC	1.97	2.27	4.34	2.02	2.49	4.41	2.01	2.35	/
Bern.	DeC	1.97	2.61	1.8	2.29	2.52	2.27	1.97	2.7	2.06

Table 5: Convergence order for all schemes on linear advection test, using coefficients obtained in Table 15. "/" means that the Fourier analysis showed that the scheme is unstable.

Looking at the table 5, we observe that almost all the stabilized schemes provide the expected order of accuracy. Exception to this rule are several $\mathbb{P}_2$ discretization which reach an order of accuracy of ≈ 2.5, and all *Bernstein* $\mathbb{B}_3$ polynomials with the *DeC* which reach an order of accuracy of 2. This result is very disappointing and it does not improve even adding more corrections, as suggested in [3, 1]. Moreover, it has been independently verified that also in Fourier space the accuracy of DeC with Bernstein polynomials of degree 3 is only of order 2. This problem do not show up for steady problems, as there only the spatial discretization determines the order of accuracy. We will show it in Section 5.3, where we study also some steady vortexes. The authors still do not understand why the optimal order of accuracy is not reached. This opens doors to further research on this family of schemes.

Note that we do not show results for *Bernstein* elements with *SSPRK* technique because they are identical to *Basic* elements, but are more expensive because of the projection in the *Bernstein* element space and the interpolation in the quadrature points.

More comparisons on different grids (unstructured) will be done in Section 5.

4.2 Shallow water equations

We consider the non linear shallow water equations (no friction and constant topography):

$$\begin{cases} \partial_t h + \partial_x(hu) + \partial_y(hv) & = 0, \\ \partial_t(hu) + \partial_x(hu^2 + g\frac{h^2}{2}) + \partial_y(huv) & = 0, \\ \partial_t(hv) + \partial_x(huv) + \partial_y(hv^2 + g\frac{h^2}{2}) & = 0, \end{cases} \qquad \begin{array}{l} x \in \Omega = [0,2] \times [0,1], \\ t \in [0, t_f] \\ t_f = 1s. \end{array} \tag{57}$$

An analytical solution of this system is given by travelling vortexes [53]. We use here a vortex with compact support and in $\mathcal{C}^6(\Omega)$ described by

$$\begin{pmatrix} h(x,t) \\ u(x,t) \\ v(x,t) \end{pmatrix} = \begin{cases} \begin{pmatrix} h_c + \frac{1}{g}\frac{\Gamma^2}{\omega^2} \cdot (\lambda(\omega\mathcal{R}(\mathbf{x},t)) - \lambda(\pi)), \\ u_c + \Gamma(1+\cos(\omega\mathcal{R}(\mathbf{x},t)))^2 \cdot (-\mathcal{I}(\mathbf{x},t)_y), \\ v_c + \Gamma(1+\cos(\omega\mathcal{R}(\mathbf{x},t)))^2 \cdot (\mathcal{I}(\mathbf{x},t)_x), \end{pmatrix}, & \text{if } \omega\mathcal{R}(\mathbf{x},t) \leq \pi, \\ \begin{pmatrix} h_c & u_c & v_c \end{pmatrix}^T, & \text{else,} \end{cases} \tag{58}$$

with

$$\begin{aligned} \lambda(r) = & \frac{20\cos(r)}{3} + \frac{27\cos(r)^2}{16} + \frac{4\cos(r)^3}{9} + \frac{\cos(r)^4}{16} + \frac{20r\sin(r)}{3} \\ & + \frac{35r^2}{16} + \frac{27r\cos(r)\sin(r)}{8} + \frac{4r\cos(r)^2\sin(r)}{3} + \frac{r\cos(r)^3\sin(r)}{4}. \end{aligned}$$

where $\mathbf{X_c} = (0.5, 0.5)$ is the initial vortex center, $(h_c,\, u_c,\, v_c) = (1.,\, 0.6,\, 0)$ is the far field state, $r_0 = 0.45$ is the vortex radius, $\Delta h = 0.1$ is the vortex amplitude, and the remaining paramters are defined as

$$\begin{cases} \omega = \pi / r_0 & \text{angular wave frequency,} \\ \Gamma = \frac{12\pi\sqrt{g\Delta h}}{r_0\sqrt{315\pi^2 - 2048}} & \text{vortex intensity parameter,} \\ \mathcal{I}(\mathbf{x},t) = \mathbf{x} - \mathbf{X_c} - (u_c t, v_c t)^T & \text{coordinates with respect to the vortex center,} \\ \mathcal{R}(\mathbf{x},t) = \|\mathcal{I}(\mathbf{x},t)\| & \text{distance from the vortex center.} \end{cases} \tag{59}$$

We discretize the mesh with uniform square intervals of length Δx (see figure 20(a)), and as before we perform a grid convergence by respecting the constraint $\Delta x_2 = 2\Delta x_1$ for $\mathbb{P}_2$ elements and $\Delta x_3 = 3\Delta x_1$ for $\mathbb{P}_3$ elements. Because of the high cost of the SUPG technique, we only compare the OSS and the CIP stabilization techniques. As an example of results, we again show the benefit of using *Cubature* elements in 23. We can see that since the dimension of the discretized system is even larger than before (three times larger), the differences between *Cubature* and *Basic* elements are even more highlighted in the error-computational time plot.

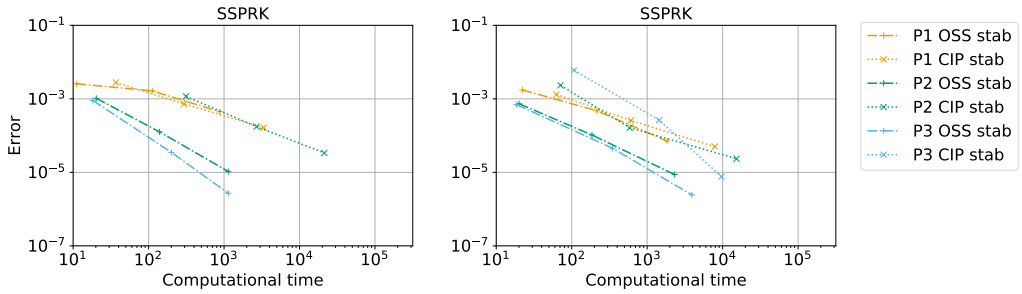


Figure 23: Error for shallow water system (57) with respect to computational time for SSPRK method with *Cubature* (left) and *Basic* (right) elements and CIP and OSS stabilizations.

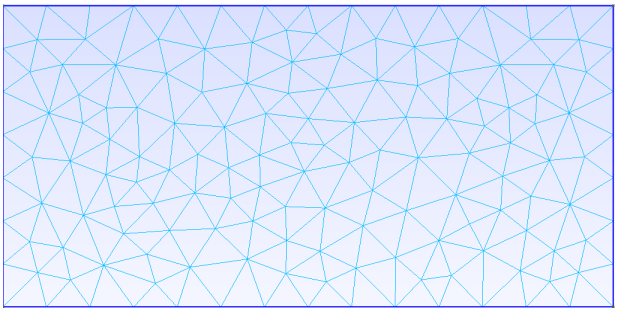


Figure 24: Unstructured mesh on $\Omega = [0, 2] \times [0, 1]$.

In Table 6 we show the convergence orders for this shallow water problem with the CFL and δ coefficients found in Table 15.

Element &		OSS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	2.3	3.18	3.8	2.34	3.3	4.47
Cub.	SSPRK	1.25	3.31	3.94	2.03	2.56	/
	DeC	1.45	3.31	3.94	1.98	2.56	/
Bern.	DeC	1.52	2.93	2.97	2.92	2.12	2.91

Table 6: Convergence order on shallow water, using coefficients obtained in Table 15.
"/" means that the fourier analysis shown that the scheme is unstable.

The results obtained are similar to those of the *linear advection* case. We can also notice the $\mathbb{P}_2$ discretization reaching the proper convergence order, i.e., 3, and *Bernstein* $\mathbb{B}_3$ elements reaching an order of accuracy of ≈ 3 which is more satisfying than the results obtained for the linear advection test, but still disappointing knowing that we were expecting 4.

5 Simulations on unstructured meshes

We now perform numerical tests to check the validity of our theoretical findings using an unstructured mesh, and the most restrictive parameters in Table 4. These parameters make sure that we are stable for both *T* and *X* mesh configurations. The results have similar convergence rate to the tests on the structured meshes of the previous section. The unstructured mesh used in this section is shown in Figure 24, and it was created by the mesh generator *gmsh*[1].

[1] https://gmsh.info/

5.1 Linear advection test

Element &		SUPG			OSS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	1.9	2.57	3.76	1.99	2.5	3.76	1.57	2.14	3.66
Cub.	SSPRK	1.73	2.4	3.83	1.81	2.53	3.98**	1.8	2.17	/
	DeC	1.81	2.21	2.56	1.82	2.48	3.98**	1.83	2.17	/
Bern.	DeC	1.78	2.12	1.94	2.31	2.48	2.12	1.56	2.03	2.24

Table 7: Convergence order for linear advection on unstructured mesh, using coefficients obtained in Table 4.
** These values are found using only the X mesh (see Figure 17).
"/" means that the scheme is clearly unstable.

We use the same test case of Section 4.1. Convergence orders for all schemes are summarized in Table 7. We observe that all $\mathbb{P}_1$ discretizations provide the proper convergence order. For $\mathbb{P}_2$ discretization we spot a slight reduction of the order of accuracy, which lays for most of the schemes between 2 and ≈ 2.5 instead of being 3. For polynomials of degree 3, we observe an order reduction to 2 for the same schemes that lost the right order of accuracy also for *X* mesh in the previous section. In particular, we have that *Bernstein* $\mathbb{B}_3$ polynomials with the *DeC* result in an order of accuracy of ≈ 2 instead of 4, as well as the $\tilde{\mathbb{P}}_3$ discretization with the combination DeC and SUPG stabilization. As for the X mesh, the *Basic* $\mathbb{P}_3$ discretization reach order of accuracy ≈ 4 for all stabilization techniques, as well as *Cubature* $\tilde{\mathbb{P}}_3$ with SUPG and OSS stabilizations.
Also in this case, the results obtained with $\tilde{\mathbb{P}}_3$ *Cubature* elements and OSS stabilization are stable as we can see from the convergence analysis. This might mean that just few unfortunate mesh configurations, as the T one, result in an unstable scheme and that, most of the time, the parameters found in Table 4 are reliable for this scheme. On the other hand, the combination $\tilde{\mathbb{P}}_3$ and CIP gives an unstable scheme.

We compare error and computational time for all methods presented above in Figure 25. Looking at $\mathbb{P}_2$ and the $\mathbb{P}_3$ discretizations, as expected, the mass-matrix free combination, i.e., *Cubature* elements with SSPRK and OSS, gives smaller computational costs than other combinations with *Basic* elements. Conversely, the SUPG technique increase the computational costs with respect to all other stabilizations for all schemes. That is why we will not use it for the next test. The plots and all the errors are available at the repository [43].

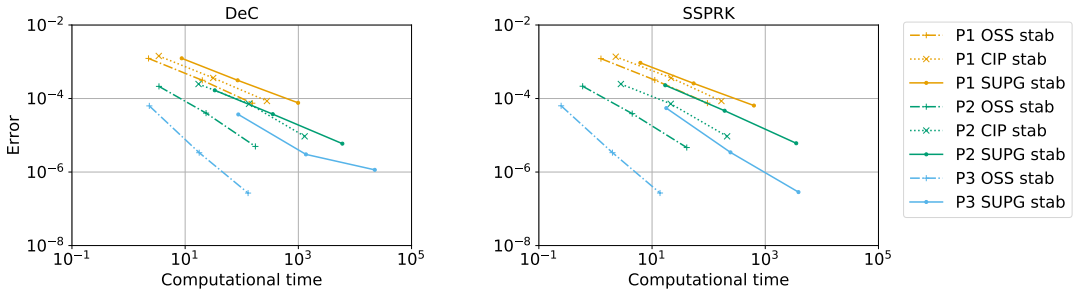
Remark 4 (Entropy viscosity) *As remarked in Section 3.9, we can improve the stability of some schemes (Cubature OSS) with extra entropy viscosity. Here, we test the convergence rate on the T mesh configuration, i.e., the one with more restrictive CFL conditions and most unstable. This test is performed using Cubature* $\tilde{\mathbb{P}}_3$ *elements, SSPRK and DeC time integration methods, and the OSS and the CIP stabilization techniques. We solve again problem* (56)*.*

Using formulation (53) *and tuning stability coefficient* δ*, CFL and viscosity coefficient* c *found in Figure 19, we obtain fourth order accurate schemes. These tuned coefficients, and the corresponding convergence orders are summarized in Table 8.*

Element &		*Cubature* $\tilde{\mathbb{P}}_3$ OSS			*Cubature* $\tilde{\mathbb{P}}_3$ CIP		
Time scheme		CFL (δ)	c	order	CFL (δ)	c	order
Cub.	SSPRK	0.15 (0.02)	0.05	4.08	0.12 (0.0004)	0.5	3.60
	DeC	0.15 (0.02)	0.05	4.09	0.08 (0.001)	0.2	3.76

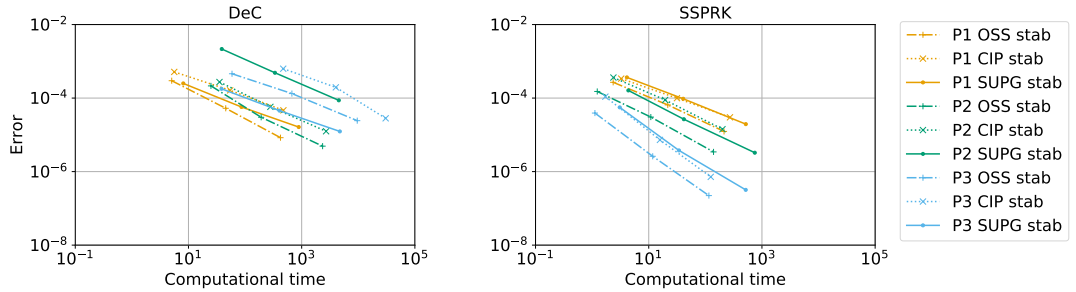
Table 8: Convergence order of methods using *Cubature* $\tilde{\mathbb{P}}_3$ elements and viscosity term (53) with tuned parameters

Many other formulations of viscosity terms exist in literature and can ensure convergent methods of order $p+1$ *(using* $\mathbb{P}_p$ *elements) [30, 36, 41]. The majority use a nonlinear evaluation of the parameter* μ_K*, based on the local entropy production.*



(a) *Cubature* elements with DeC

(b) *Cubature* elements with SSPRK



(c) *Bernstein* elements with DeC

(d) *Basic* elements with SSPRK

Figure 25: Error for linear advection problem (56) with respect to computational time for all elements and stabilization techniques

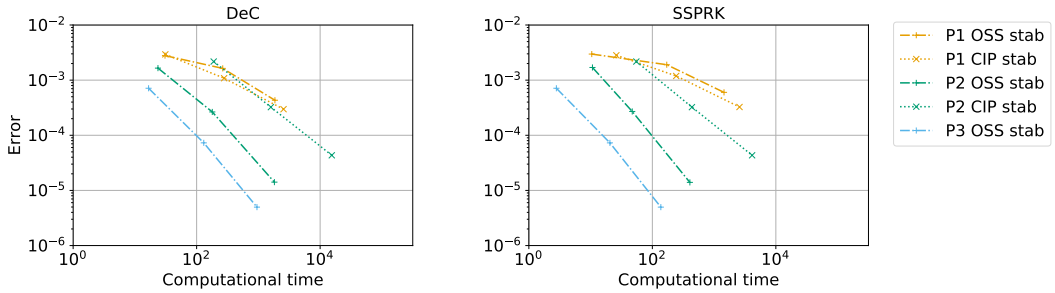
5.2 Shallow water equations

Element &		OSS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	1.94	2.98	4.25	2.15	2.52	4.11
Cub.	SSPRK	1.03	3.17	3.59**	1.39	2.57	/
	DeC	1.2	3.14	3.59**	1.48	2.57	/
Bern.	DeC	1.28	3.14	3.15	1.36	2.73	2.66

Table 9: Convergence order on shallow water for unstructured mesh, using coefficients obtained in Table 4.
** These values are found using only the X mesh (see Figure 17).
"/" means that the scheme is clearly unstable.

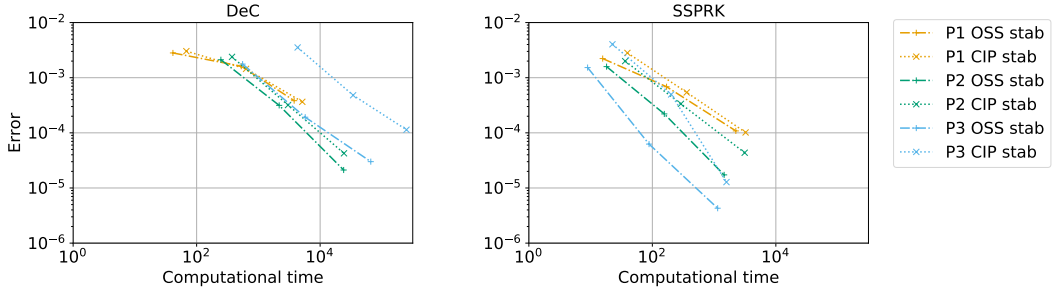
In this section we test the proposed schemes on the test case of Section 4.2 with the unstructured mesh in Figure 24. Convergence orders are summarized in Table 9. Also for the shallow water equations, we have results that resemble the ones of the structured mesh. There are small differences in the order of accuracy in both directions in different schemes. Comparing also the computational time of all the schemes in Figure 26, we can choose what we consider the best numerical method for these test cases: *Cubature* discretization with the OSS stabilization technique. This performance seems fully provided by the free mass-matrix inversion, as the CFLs for the OSS technique (with SSPRK scheme) is approximately the same between *Basic* and *Cubature* elements (see Table 4).

The plots and all the errors are available at the repository [43].



(a) *Cubature* elements with DeC

(b) *Cubature* elements with SSPRK



(c) *Bernstein* elements with DeC

(d) *Basic* elements with SSPRK

Figure 26: Error for shallow water problem (57) with respect to computational time for all elements and stabilization techniques

5.3 Remark on the steady vortex case

For completeness we consider now a steady vortex, similarly to what reported in [3] for the isentropic Euler equations. So, we consider again the traveling vortex proposed in Section 4.2 with $t_f = 0.1s$. We compare the convergence orders between $u_c = 0$ (steady case) and $u_c = 0.6$ (unsteady case) in Table 10 and Table 11. As we can see, in the steady case we obtain, *without any additional viscous stabilization*, the expected convergence order for all schemes, in particular for the DeC with Bernstein polynomial function. These results agree with the ones in [3]. Comparing with the unsteady case, all the other schemes reach similar order of accuracy as obtained in Table 9. Running the test with additional corrections in DeC scheme, as often suggested in [3, 1], does not improve the convergence order in the unsteady case (even with $K = 50$).

Element &		OSS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	2.31	2.67	3.89	1.97	2.64	3.62
Cub.	SSPRK	2.05	3.2	3.56	1.79	2.83	/
	DeC	2.17	3.18	3.57	1.74	2.83	/
Bern.	DeC	2.33	3.28	3.65	1.85	3.0	3.63

Table 10: Convergence order for steady vortex, $t_f = 0.1s$. "/" means that the scheme is clearly unstable.

OSS			CIP		
$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
2.34	2.68	3.86	1.94	2.53	3.61
2.03	3.13	3.57	1.74	2.7	/
2.13	3.09	3.57	1.71	2.7	/
2.33	3.19	2.87	1.75	2.77	2.76

Table 11: Convergence order for unsteady vortex, $t_f = 0.1s$.

These results show that a numerical error appears in the spatio-temporal integration part of the solution (27), which might be related to the fact that the high order derivatives are never penalized in our stabilizations and might produce

some small oscillations.

6 Conclusion

This work shows also that the stability results obtained in the one dimensional analysis [42] can not be generalized for two dimensional problems on triangular meshes. In this direction, it could be interesting to perform the stability analysis on Cartesian quadrilateral meshes, to check whether in that situation the one dimensional results still hold true.

In the numerical test section, the order of accuracy found is not the expected one for all the methods, i.e., $p+1$ using $\mathbb{P}_p$ elements. For several cases, we reach only $p+1/2$ or p. Among the schemes that are stable and with the right order of accuracy, the method that uses *Cubature* elements with OSS stabilization technique and SSPRK method of order 4 has proven to be the most accurate and less expensive. Secondly, comparing to the SUPG stabilization technique, very often used in the literature for hyperbolic system, we showed that other stabilization techniques such as CIP and OSS can provide the same accuracy and are cheaper in term of computational costs.

In this direction, it would be interesting to evaluate the stability of the CIP adding a additional penalty term on the jump of higher order derivatives as suggested in [17, 13, 3]. Moreover, it could be interesting to see the stability of *Cubature* elements using higher degree polynomials. Another interesting point to explore is the loss of accuracy obtained using the DeC with *Bernstein* third order polynomial basis functions for unsteady cases.

Finally, we provided a heuristic approach characterized by additional discontinuity capturing viscous operators such as those proposed in [30, 36]. Even for smooth solutions, the very small additional dissipation introduced by these terms is enough to stabilize some of the symmetric mass-matrix-free approaches, otherwise linearly unstable. This allows to obtain interesting schemes for practical purposes.

Acknowledgment

This work was performed within the Ph.D. project of Sixtine Michel: "Finite element method for shallow water equations", supported by INRIA and the BRGM, co-funded by in INRIA–Bordeaux Sud–Ouest and the Conseil Régional de la Nouvelle Aquitaine. Davide Torlo has been funded by a postdoctoral fellowship in the team CARDAMOM in INRIA–Bordeaux Sud–Ouest and by a postdoctoral fellowship in SISSA. Rémi Abgrall has been supported by the Swiss National Foundation grant "Solving advection dominated problems with high order schemes with polygonal meshes: application to compressible and incompressible flow problems" under Grant Agreement No 200020_175784.

Declarations

Funding Sixtine Michel was funded by in INRIA–Bordeaux Sud–Ouest and the Conseil Régional de la Nouvelle Aquitaine. Davide Torlo has been funded by a postdoctoral fellowship in the team CARDAMOM in INRIA–Bordeaux Sud–Ouest and by a postdoctoral fellowship in SISSA Rémi Abgrall has been supported by the Swiss National Foundation grant "Solving advection dominated problems with high order schemes with polygonal meshes: application to compressible and incompressible flow problems" under Grant Agreement No 200020_175784.
Conflicts of interest/Competing interests The authors certify that there is no actual or potential conflict of interest in relation to this article.
Availability of data and material The images for all the parameters of the stability analysis and convergence plots are available at [43].

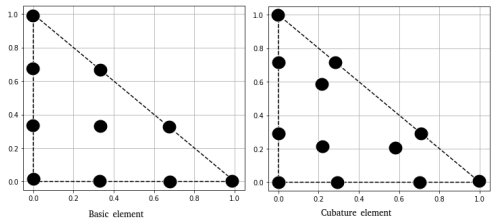


Figure 27: Comparison of two element of degree three: at left the classical one $\mathbb{P}_3$, at right the *Cubature* one $\tilde{\mathbb{P}}_3$.

A *Cubature* elements, definition and construction

In this section we give a description of the *Cubature* finite elements [29, 25]. In Figure 27 we show the $\tilde{\mathbb{P}}_3$ example comparing the Lagrangian nodes of *Basic* and *Cubature* elements. As defined in Section 2.2.3, there are several requirements and optimization procedures in order to obtain the *Cubature* elements. These elements are very import in our study because they permit to obtain diagonal mass matrix, and so they decrease considerably the time of computation. We describe for $p = 1, 2, 3$ the basis functions of the *Cubature* elements.

A.1 *Cubature* elements of degree 1

The $\tilde{\mathbb{P}}_1$ element contains 3 degree of freedom. Their nodes are located at the vertices $v_1 = (1, 0, 0)$, $v_2 = (0, 1, 0)$ and $v_3 = (0, 0, 1)$ of the triangle.

- At vertices of the triangle:
$$\phi_{v_i}(\lambda) = \lambda_i, \qquad \text{for } i = 1, 2, 3.$$

Corresponding weights are $w_{v_i} = \frac{1}{3}$.

A.2 *Cubature* elements of degree 2

The $\tilde{\mathbb{P}}_2$ element contains 7 degrees of freedom: three at the vertices v_1, v_2 and v_3 and three at the midpoint of the edges that we denote as $e_{ij} = \frac{v_i+v_j}{2}$ for $(i, j) \in \{(1, 2), (2, 3), (3, 1)\}$ and one at the centroid point $G_\beta := \frac{v_1+v_2+v_3}{3}$. Respectively, we have the following basis functions and weights:

- At vertices of the triangle
$$\phi_{v_i}(\lambda) = \lambda_i(2\lambda_i - 1) + 3\lambda_1\lambda_2\lambda_3, \text{ for } i \in [\![1, \dots, 3]\!],$$
$$w_v = \frac{1}{20};$$

- At edge midpoints
$$\phi_{e_{ij}}(\lambda) = 4\lambda_i\lambda_j(1 - 3\lambda_k), \text{ for all } i \neq j \neq k \neq i \in [\![1, \dots, 3]\!],$$
$$w_e = \frac{2}{15};$$

- At the centroid
$$\phi_{G_\beta}(\lambda) = 27\lambda_1\lambda_2\lambda_3,$$
$$w_\beta = \frac{9}{20}.$$

A.3 *Cubature* elements of degree 3

Following [25, 29] we derive the definitions of all the basis functions and points of *Cubature* elements $\tilde{\mathbb{P}}_3$. The notations are not uniform among different works, so we use the following one which can be used with all the different elements we have used in this work.

The space $\tilde{\mathbb{P}}_3$ contains 12 degrees of freedom: 3 vertices v_1, v_2 and v_3, 6 on edges: e_{ij}^{α} for $i,j \in [\![1,\dots,3]\!]$ with $i \neq j$ defined by

$$e_{ij}^{\alpha} = (\delta_{1i}\alpha + \delta_{1j}(1-\alpha), \delta_{2i}\alpha + \delta_{2j}(1-\alpha), \delta_{3i}\alpha + \delta_{3j}(1-\alpha))$$
$$\text{with } \alpha = \frac{-15\sqrt{7} - 21 + \sqrt{168 + 174\sqrt{7}}}{2(-15\sqrt{7}-21)},$$

with δ_{ij} is the Kronecker delta and three internal points G_i^{β} for $i \in [\![1,\dots,3]\!]$, with

$$G_i^{\beta} = \left(\beta\delta_{i1} + \frac{1-\beta}{2}(1-\delta_{i1}), \beta\delta_{i2} + \frac{1-\beta}{2}(1-\delta_{i2}), \beta\delta_{i3} + \frac{1-\beta}{2}(1-\delta_{i3})\right) \text{ with } \beta = \frac{1}{3} + \frac{2\sqrt{7}}{21},$$

where α and β are found through an optimization process [25, 29]. Let us start giving the definitions of the weights for the different types of points. We have that $w_v = \frac{1369+767\sqrt{7}}{120(859+395\sqrt{7})}$ is the weight for the vertices of the triangle, $w_\alpha = \frac{287+115\sqrt{7}}{40(173+49\sqrt{7})}$ is the weight on edges points, and w_β the weight for barycentric points.
The weights corresponding to these types of points are $w_v = \frac{1369+767\sqrt{7}}{120(859+395\sqrt{7})}$, $w_\alpha = \frac{287+115\sqrt{7}}{40(173+49\sqrt{7})}$ and $w_\beta = \frac{21\sqrt{7}}{40(2\sqrt{7}+1)}$. In order to simply the formulation of the basis functions, let us introduce some polynomials:

$$p_i(\lambda) := \lambda_i \left(\sum_{l=1}^{3}\lambda_l^2 - \frac{1-2\alpha+2\alpha^2}{\alpha(1-\alpha)}\lambda_i(\lambda_j+\lambda_k) + A_i\lambda_j\lambda_k\right), \qquad \text{with } j \neq i \neq k, \tag{60}$$

with

$$A_i = \left(w_v - \frac{1}{10} - \frac{1}{15}\left(1 - \frac{1-2\alpha+2\alpha^2}{\alpha(1-\alpha)}\right) - \frac{1}{90}\frac{8}{\beta(1-\beta)^2(3\beta-1)}\left(\sum_{l=1}^{3}p_i(G_l)\right)\right)\frac{360}{6 + \frac{8(1+\beta)}{\beta(1-\beta)(3\beta-1)}}; \tag{61}$$

$$p_{ij}(\lambda) := \frac{1}{\alpha(1-\alpha)(2\alpha-1)}\lambda_i\lambda_j(\alpha\lambda_i - (1-\alpha)\lambda_j + (1-2\alpha)\lambda_k), \text{ with } i \neq j \neq k \neq i. \tag{62}$$

We can then write the definition of the basis functions:

- At vertices of the triangle

$$\phi_{v_i}(\lambda) = p_i(\lambda) - \frac{8}{\beta(1-\beta)^2(3\beta-1)}\left(\sum_{l=1}^{3}p_i(G_l)\left(\lambda_l - \frac{1-\beta}{2}\right)\right)\prod_{l=1}^{3}\lambda_l, \text{ for } i \in [\![1,\dots,3]\!];$$

- At the nodes on edges

$$\phi_{e_{ij}^{\alpha}}(\lambda) = p_{ij}(\lambda) - \frac{8}{\beta(1-\beta)^2(3\beta-1)}\left(\sum_{l=1}^{3}p_{ij}(G_l)\left(\lambda_l - \frac{1-\beta}{2}\right)\right)\prod_{l=1}^{3}\lambda_l, \text{ for } i \neq j \in [\![1,\dots,3]\!];$$

- At the internal points

$$\phi_{G_i^{\beta}}(\lambda) = \frac{8}{\beta(1-\beta)^2(3\beta-1)}\left(\lambda_i - \frac{1-\beta}{2}\right)\prod_{l=1}^{3}\lambda_l, \text{ for } i \in [\![1,\dots,3]\!].$$

B Time discretization coefficients

In this appendix we introduce the time integration coefficients used in this work, to make the study fully reproducible. In Table 12 there are the RK coefficients, in Table 13 the SSPRK coefficients and in Table 14 the DeC coefficients.

RK2		
α	1	
β	$\frac{1}{2}$	$\frac{1}{2}$

RK3			
α	$\frac{1}{2}$		
	-1	2	
β	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

RK4				
α	$\frac{1}{2}$			
	0	$\frac{1}{2}$		
	0	0	1	
β	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Table 12: Butcher Tableau of RK methods

SSPRK(3,2) by [56]					
γ			μ		
1			$\frac{1}{2}$		
0	1		0	$\frac{1}{2}$	
$\frac{1}{3}$	0	$\frac{2}{3}$	0	0	$\frac{1}{3}$
CFL = 2.					

SSPRK(4,3) by [54, Page 189]							
γ				μ			
1				$\frac{1}{2}$			
0	1			0	$\frac{1}{2}$		
$\frac{2}{3}$	0	$\frac{1}{3}$		0	0	$\frac{1}{6}$	
0	0	0	1	0	0	0	$\frac{1}{2}$
CFL = 2.							

SSPRK(5,4) by [54, Table 3]				
γ				
1				
0.444370493651235	0.555629506348765			
0.620101851488403	0	0.379898148511597		
0.178079954393132	0	0	0.821920045606868	
0	0	0.517231671970585	0.096059710526147	0.386708617503269
μ				
0.391752226571890				
0	0.368410593050371			
0	0	0.251891774271694		
0	0	0	0.544974750228521	
0	0	0	0.063692468666290	0.226007483236906
CFL = 1.50818004918983				

Table 13: Butcher Tableau of SSPRK methods

Order 2			
m	β^m	ρ_z^m	
1	1	$\frac{1}{2}$	$\frac{1}{2}$

Order 3				
m	β^m	ρ_z^m		
1	$\frac{1}{2}$	$\frac{5}{24}$	$\frac{1}{3}$	$-\frac{1}{24}$
2	1	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{3}$

Order 4					
m	β^m	ρ_z^m			
1	$\frac{1}{3}$	$\frac{1}{8}$	$\frac{19}{72}$	$-\frac{5}{72}$	$\frac{1}{72}$
2	$\frac{2}{3}$	$\frac{1}{9}$	$\frac{4}{9}$	$\frac{1}{9}$	0
3	1	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$

Table 14: DeC coefficients for equispaced subtimesteps.

C Fourier analysis

In this section we collect all the plots and results that are essential to show the results of this work, but for structural reasons were not put in the main text.

C.1 Mesh types and degrees of freedom

We represent in Figure 28 the mesh configurations used in the Fourier analysis and the degrees of freedom of the elements of degree 3. The red square represents the periodic elementary unit that contains the degrees of freedom of interest for the Fourier analysis.

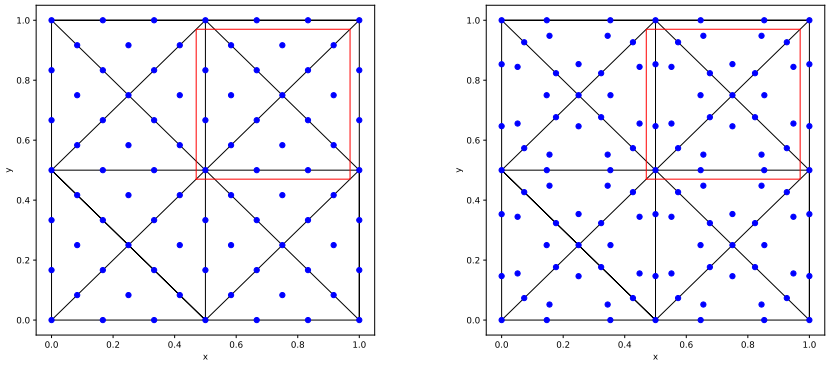
C.2 Fourier analysis results - Optimal Parameters

In this section, we put the optimal values of the stability analysis of Section 3.5 after the modification proposed in Section 3.6. In Table 15 we show the parameters for the *X* mesh and in Table 16 we show the parameters for the *T* mesh.

Element &		SUPG		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	0.739 (0.127)	0.298 (0.058)	0.22 (0.026)
Cub.	SSPRK	1.062 (0.28)	0.1 (0.1)*	0.18 (0.04)*
	DeC	0.616 (0.28)	0.1 (0.04)*	0.144 (0.04)
Bern.	DeC	0.739 (0.298)	0.2 (0.2)*	0.2 (0.153)*

Element &		OSS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	0.403 (0.127)	0.298 (0.026)	0.22 (0.026)	0.403 (0.012)	0.298 (1.73e-03)	0.1 (1.00e-03)*
Cub.	SSPRK	0.58 (0.336)	0.379 (0.03)	0.248 (0.018)	0.58 (0.048)	0.06 (0.01)*	/
	DeC	0.379 (0.207)	0.248 (0.03)	0.162 (0.018)	0.379 (0.026)	0.06 (0.01)*	/
Bern.	DeC	0.173 (0.58)	0.036 (0.298)	0.015 (0.078)*	0.173 (0.153)	0.012 (0.021)	0.002 (8.00e-03)*

Table 15: *X* mesh: Optimized CFL and penalty coefficient δ in parenthesis. The symbol "/" means that the fourier analysis for the scheme results always in instability. The values denoted by * are not the optimal one, but they lay in a safer region, see Section 3.6.



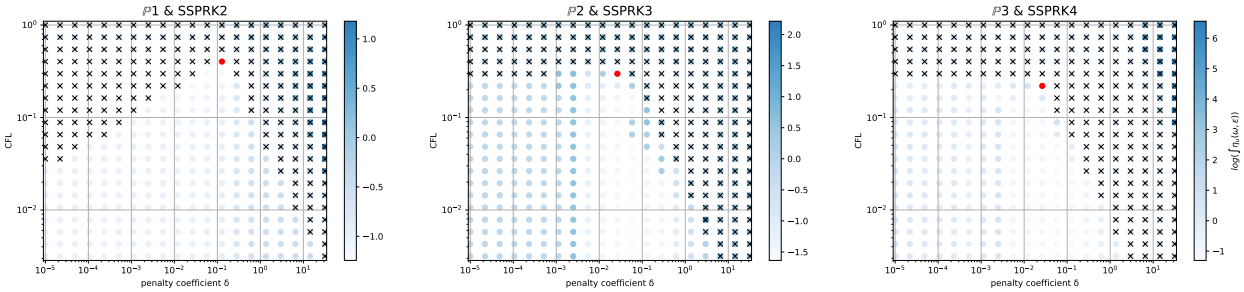
(a) *X* mesh, *Basic* elements

(b) *X* mesh, *Cubature* elements

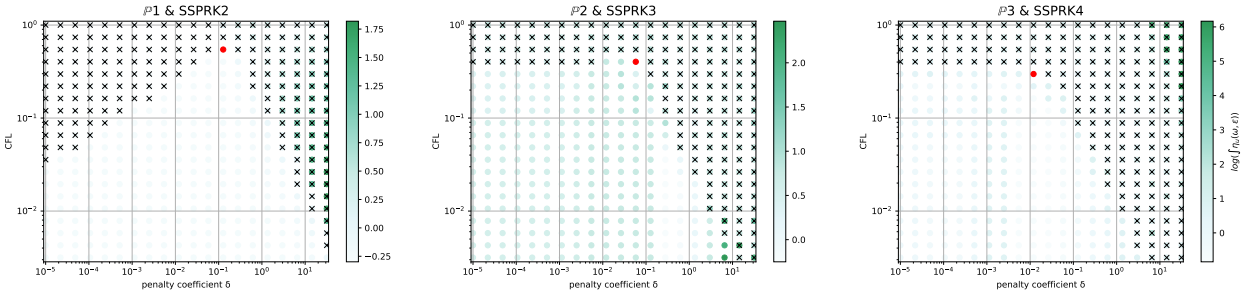
(c) *T* mesh, *Cubature* elements

(d) *T* mesh, *Cubature* elements

Figure 28: Degrees of freedom and periodic unit for different mesh patterns and elements of degree 3



(a) *X* mesh



(b) *T* mesh

Figure 29: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ *Basic* elements with SSPRK scheme and OSS stabilization.

Element &		SUPG		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	0.739 (0.127)	0.403 (0.026)	0.298 (0.012)
Cub.	SSPRK	1.062 (0.28)	0.234 (0.078)	0.055 (0.153)
	DeC	1.062 (0.127)	0.144 (0.078)	0.034 (0.153)
Bern.	DeC	0.739 (0.298)	0.739 (0.153)	0.455 (0.153)

Element &		OSS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	0.546 (0.127)	0.403 (0.058)	0.298 (0.012)	0.546 (0.026)	0.298 (7.39e-05)	0.298 (3.36e-05)
Cub.	SSPRK	0.886 (0.336)	0.379 (0.048)	/	0.886 (0.048)	0.106 (7.85e-03)	/
	DeC	0.58 (0.207)	0.379 (0.03)	/	0.58 (0.026)	0.045 (7.85e-03)	/
Bern.	DeC	0.28 (0.58)	0.025 (0.153)	0.074 (0.078)	0.455 (0.078)	0.025 (5.46e-03)	0.017 (0.04)

Table 16: *T* mesh: Optimized CFL and penalty coefficient δ in parenthesis. The symbol "/" means that the fourier analysis for the scheme results always in instability.

C.3 Fourier analysis results - stability area

Finally, we present a comparison of stability area between the *T* and the *X* mesh. This comparison if perform as before, for all wave angles θ. We choose as example the comparison using *Basic* element, SSPRK time integration method and the OSS stabilization technique in Figure 29.The interested reader can access to results for all methods online [43].

References

[1] R. Abgrall. High order schemes for hyperbolic problems using globally continos approximation and avoiding mass matrices. *Journal of Scientific Computing*, 73, 07 2017.

[2] R. Abgrall. A general framework to construct schemes satisfying additional conservation relations. application to entropy conservative and entropy dissipative schemes. *Journal of Computational Physics*, 372:640 – 666, 2018.

[3] R. Abgrall, P. Bacigaluppi, and S. Tokareva. High-order residual distribution scheme for the time-dependent Euler equations of fluid dynamics. *Computers & Mathematics with Applications*, 78(2):274–297, 2018.

[4] R. Abgrall, J. Nordström, P. Öffner, and S. Tokareva. Analysis of the SBP-SAT Stabilization for Finite Element Methods Part I: Linear Problems. *Journal of Scientific Computing*, 85(43):1573–7691, 2020.

[5] R. Abgrall, J. Nordström, Ö. Philipp, and S. Tokareva. Analysis of the SBP-SAT Stabilization for Finite Element Methods Part II: Entropy Stability. *Commun. Appl. Math. Comput.*, pages 2661–8893, 2021.

[6] R. Abgrall, P. Öffner, and H. Ranocha. Reinterpretation and Extension of Entropy Correction Terms for Residual Distribution and Discontinuous Galerkin Schemes: Application to Structure Preserving Discretization. *Journal of Computational Physics*, page 110955, 2022.

[7] R. Abgrall and M. Ricchiuto. High order methods for CFD. In R. d. B. Erwin Stein and T. J. Hughes, editors, *Encyclopedia of Computational Mechanics, Second Edition*. John Wiley and Sons, 2017.

[8] R. Abgrall and D. Torlo. High order asymptotic preserving deferred correction implicit-explicit schemes for kinetic models. *SIAM Journal on Scientific Computing*, 42(3):B816–B845, 2020.

[9] L. Arpaia, M. Ricchiuto, A. G. Filippini, and R. Pedreros. An efficient covariant frame for the spherical shallow water equations: Well balanced DG approximation and application to tsunami and storm surge. *Ocean Modelling*, 169:101915, 2022.

[10] P. Bacigaluppi, R. Abgrall, and S. Tokareva. "A Posteriori" Limited High Order and Robust Residual Distribution Schemes for Transient Simulations of Fluid Flows in Gas Dynamics. *arXiv preprint arXiv:1902.07773*, 2019.

[11] S. Badia and R. Codina. Unified Stabilized Finite Element Formulations for the Stokes and the Darcy Problems. *SIAM Journal on Numerical Analysis*, 47, 01 2009.

[12] E. Burman. Consistent SUPG-method for transient transport problems: Stability and convergence. *Computer Methods in Applied Mechanics and Engineering*, 199:1114–1123, 03 2010.

[13] E. Burman. Weighted error estimates for transient transport problems discretized using continuous finite elements with interior penalty stabilization on the gradient jumps. *arXiv preprint arXiv:2104.06880*, 2021.

[14] E. Burman, A. Ern, and M. Fernández. Explicit Runge–Kutta Schemes and Finite Elements with Symmetric Stabilization for First-Order Linear PDE Systems. *SIAM Journal on Numerical Analysis*, 48, 01 2010.

[15] E. Burman and P. Hansbo. Edge stabilization for Galerkin approximations of convection–diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 193:1437–1453, 04 2004.

[16] E. Burman and P. Hansbo. The edge stabilization method for finite elements in CFD. In *Numerical mathematics and advanced applications*, pages 196–203. Springer, 2004.

[17] E. Burman, P. Hansbo, and M. G. Larson. A cut finite element method for a model of pressure in fractured media. *Numerische Mathematik*, 146(4):783–818, 2020.

[18] E. Burman, A. Quarteroni, and B. Stamm. Stabilization strategies for high order methods for transport dominated problems. *Bolletino dell'Unione Matematica Italiana*, 1, 02 2008.

[19] E. Burman, A. Quarteroni, and B. Stamm. Interior penalty continuous and discontinuous finite element approximations of hyperbolic equations. *Journal of Scientific Computing*, 43:293–312, 06 2010.

[20] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Ltd, 2008.

[21] J. C. Butcher. Numerical differential equation methods. In *Numerical Methods for Ordinary Differential Equations*, chapter 2, pages 55–142. John Wiley & Sons, Ltd, 2016.

[22] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Studies in mathematics and its applications 4. North-Holland, 1978.

[23] R. Codina. Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 190(13):1579–1599, 2000.

[24] R. Codina and J. Blasco. A finite element formulation for the Stokes problem allowing equal velocity-pressure interpolation. *Computer Methods in Applied Mechanics and Engineering*, 143(3):373–391, 1997.

[25] G. Cohen, P. Joly, J. Roberts, and N. Tordjman. Higher order triangular finite elements with mass lumping for the wave equation. *Siam Journal on Numerical Analysis*, 38, 01 2001.

[26] J. Douglas and T. Dupont. Interior penalty procedures for elliptic and parabolic galerkin methods. In *Computing methods in applied sciences*, pages 207–216. Springer, 1976.

[27] A. Dutt, L. Greengard, and V. Rokhlin. Spectral deferred correction methods for ordinary differential equations. *BIT Numerical Mathematics*, 40(2):241–266, 6 2000.

[28] A. G. Filippini, S. De Brye, V. Perrier, F. Marche, M. Ricchiuto, D. Lannes, and P. Bonneton. UHAINA : A parallel high performance unstructured near-shore wave model. In D. Levacher, M. Sanchez, X. Bertin, and I. Brenon, editors, *Journées Nationales Génie Côtier - Génie Civil*, volume 15 of *Journées Nationales Génie Côtier - Génie Civil (JNGCGC)*, pages 47–56, La Rochelle, France, May 2018. Editions Paralia.

[29] F. Giraldo and M. Taylor. A diagonal-mass-matrix triangular-spectral-element method based on cubature points. *J. Eng. Math.*, 56:307–322, 2006.

[30] J.-L. Guermond, R. Pasquetti, and B. Popov. Entropy viscosity method for nonlinear conservation laws. *Journal of Computational Physics*, 230(11):4248–4267, May 2011.

[31] T. Hughes and A. Brook. Streamline upwind Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. Engrg.*, 32:199–259, 1982.

[32] T. Hughes, G. Scovazzi, and T. Tezduyar. Stabilized methods for compressible flows. *J. Sci. Comp.*, 43:343–368, 2010.

[33] S. Jund and S. Salmon. Arbitrary high-order finite element schemes and high-order mass lumping. *International Journal of Applied Mathematics and Computer Science*, 17(3):375, 2007.

[34] D. Komatitsch, R. Martin, J. Tromp, M. Taylor, and B. Wingate. Wave propagation in 2-D elastic media using a spectral element method with triangles and quadrangles. *Journal of Computational Acoustics*, 9:703–718, 06 2001.

[35] D. Kuzmin. Entropy stabilization and property-preserving limiters for P1 discontinuous Galerkin discretizations of scalar hyperbolic problems. *Journal of Numerical Mathematics*, 29(4):307–322, 2021.

[36] D. Kuzmin and M. Quezada de Luna. Entropy conservation property and entropy stabilization of high-order continuous Galerkin approximations to scalar conservation laws. *Computers & Fluids*, 213:104742, 2020.

[37] D. Kuzmin and M. Quezada de Luna. Subcell flux limiting for high-order bernstein finite element discretizations of scalar hyperbolic conservation laws. *Journal of Computational Physics*, 411:109411, 2020.

[38] M. G. Larson and S. Zahedi. Stabilization of high order cut finite element methods on surfaces. *IMA Journal of Numerical Analysis*, 40(3):1702–1745, 04 2019.

[39] T. Liu, M. Sen, T. Hu, J. De Basabe, and L. Li. Dispersion analysis of the spectral element method using a triangular mesh. *Wave Motion*, 49:474–483, 06 2012.

[40] Y. Liu, J. Teng, T. Xu, and J. Badal. Higher-order triangular spectral element method with optimized cubature points for seismic wavefield modeling. *Journal of Computational Physics*, 336:458 – 480, 2017.

[41] J. Llobell, S. Minjeaud, and R. Pasquetti. High order CG schemes for KdV and Saint-Venant flows. In *Numerical Methods for Flows*, pages 341–352. Springer, 2020.

[42] S. Michel, D. Torlo, M. Ricchiuto, and R. Abgrall. Spectral Analysis of Continuous FEM for Hyperbolic PDEs: Influence of Approximation, Stabilization, and Time-Stepping. *Journal of Scientific Computing*, 89(2):31, 2021.

[43] S. Michel, D. Torlo, M. Ricchiuto, and R. Abgrall. Stability analysis of several FEM methods 2D: results. `https://gitlab.inria.fr/simichel/stability-analysis-of-several-fem-methods-in-2d.-results`, May 2021.

[44] J. Miller. On the location of zeros of certain classes of polynomials with applications to numerical analysis. *Journal of the Institute of Mathematics and its Applications*, 8, 12 1971.

[45] M. Minion. Semi-implicit spectral deferred correction methods for ordinary differential equations. *Communications in Mathematical Sciences*, 1, 11 2003.

[46] R. Moura, A. F. De Castro da Silva, E. Burman, and S. Sherwin. Eigenanalysis of gradient-jump penalty (GJP) stabilisation for CG. 02 2020.

[47] R. C. Moura, M. Aman, J. Peiró, and S. J. Sherwin. Spatial eigenanalysis of spectral/hp continuous Galerkin schemes and their stabilisation via DG-mimicking spectral vanishing viscosity for high Reynolds number flows. *Journal of Computational Physics*, 406:109112, 2020.

[48] P. Öffner and D. Torlo. Arbitrary high-order, conservative and positivity preserving Patankar-type deferred correction schemes. *Applied Numerical Mathematics*, 153:15 – 34, 2020.

[49] R. Pasquetti, J. L. Guermond, and B. Popov. Stabilized spectral element approximation of the saint venant system using the entropy viscosity technique. In R. M. Kirby, M. Berzins, and J. S. Hesthaven, editors, *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, pages 397–404, Cham, 2015. Springer International Publishing.

[50] R. Pasquetti and F. Rapetti. Cubature versus Fekete-Gauss nodes for spectral element methods on simplicial meshes. *Journal of Computational Physics*, 2017.

[51] R. Pasquetti and F. Rapetti. Cubature points based triangular spectral elements: An accuracy study. *Journal of Mathematical Study*, 51(1):15–25, 2018.

[52] M. Ricchiuto and A. Bollermann. Stabilized residual distribution for shallow water simulations. *Journal of Computational Physics*, 228(4):1071–1115, 2009.

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

[54] S. Ruuth. Global optimization of explicit strong-stability-preserving Runge-Kutta methods. *Math. Comp.*, 75:183–207, 2006.

[55] S. Sherwin and G. Karniadakis. A triangular spectral element method; applications to the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 123(1):189 – 229, 1995.

[56] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77:439–471, 1988.

[57] M. A. Taylor, B. A. Wingate, and R. E. Vincent. An algorithm for computing Fekete points in the triangle. *SIAM J. Numer. Anal.*, 38(5):1707–1720, Oct. 2000.

[58] N. Tordjman. *Éléments finis d'ordre élevé avec condensation de masse pour l'équation des ondes*. PhD thesis, Université Paris VI, 1995. Thése de doctorat dirigée par Cohen, Gary Chalom Mathématiques appliquées à l'ingénierie Paris 9 1995.