Spectral Analysis of Continuous FEM for Hyperbolic PDEs: Influence of Approximation, Stabilization, and Time-Stepping

Sixtine Michel, Davide Torlo, Mario Ricchiuto, Rémi Abgrall

Spectral analysis of continuous FEM for hyperbolic PDEs: influence of approximation, stabilization, and time-stepping

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

March 31, 2021

Abstract

We study continuous finite element dicretizations for one dimensional hyperbolic partial differential equations. The main contribution of the paper is to provide a fully discrete spectral analysis, which is used to suggest optimal values of the CFL number and of the stabilization parameters involved in different types of stabilization operators. In particular, we analyze the streamline-upwind Petrov-Galerkin (SUPG) stabilization technique, the continuous interior penalty (CIP) stabilization method and the local projection stabilization (LPS). Three different choices for the continuous finite element space are compared: Bernstein polynomials, Lagrangian polynomials on equispaced nodes, and Lagrangian polynomials on Gauss-Lobatto cubature nodes. For the last choice, we only consider inexact quadrature based on the formulas corresponding to the degrees of freedom of the element, which allows to obtain a fully diagonal mass matrix. We also compare different time stepping strategies, namely Runge-Kutta (RK), strong stability preserving RK (SSPRK) and deferred correction time integration methods. The latter allows to alleviate the computational cost as the mass matrix inversion is replaced by the high order correction iterations.

To understand the effects of these choices, both time-continuous and fully discrete Fourier analysis are performed. These allow to compare all the different combinations in terms of accuracy and stability, as well as to provide suggestions for optimal values discretization parameters involved. The results are thoroughly verified 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 CIP or LPS stabilization are the most promising combinations.

Keywords: Continuous Galerkin method, Spectral element method, Streamline Upwind Petrov–Galerkin, Local Projection Stabilization, Continuous Interior Penalty, Dispersion analysis, cubature nodes, Fekete nodes, Deferred Correction scheme

MSC: 65M60

1 Introduction

In this work we compare different numerical methods that can approximate the solution of the one dimensional hyperbolic conservation laws

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

where $\Omega \subset \mathbb{R}$ is an interval, $f : \mathbb{R}^D \to \mathbb{R}^D$ is the flux function and $u : \Omega \to \mathbb{R}^D$ is the unknown of the system of equations. For the spectral analysis of the numerical methods we will mainly focus on the particular case of a linear flux

$$f(u(x,t)) = au(x,t)\,, \quad a = \text{const}\,. \tag{2}$$

In this work, we compare different explicit high order accurate schemes based on the continuous Galerkin (CG) approach. 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 the

*Team CARDAMOM, Inria Bordeaux sud-ouest, - 200 av. de la vieille tour, 33405 Talence, France
†Institut für Mathematik, Winterthurstrasse 190, CH 8057 Zürich, Switzerland.

matrix multiplication must be iterated for all the time steps. 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]. There, to avoid the full mass matrix, a mass lumping is introduced, transforming the mass matrix into a diagonal one. The deferred correction (DeC) iterative time integration method alters the right–hand side in order to recover the original order of accuracy. Another approach consists of a careful choice of quadrature points and basis functions in order to automatically obtain a diagonal mass matrix. We denote such elements as *cubature* elements [29]. The classical use of Runge–Kutta methods will provide the high order accuracy also for the time discretization.

The second aspect we will focus on is the stabilization technique. We emphasize that without any special treatment on the boundaries, such as the ones in [4, 5], the CG methods are not stable for hyperbolic problems and there is the need of stabilization. In particular, this is always true when using periodic boundary conditions (BC). The CG discretizations with stabilization techniques can have dissipation levels that are comparable to the ones brought by discontinuous Galerkin (DG) with upwind numerical flux of the same order of accuracy, still remaining decently stable [32, 33]. The stabilization terms play an important role and we will compare three of them. The first is the streamline upwind Petrov–Galerkin (SUPG) stabilization [18, 13], which is strongly consistent, but it is also introducing new terms in the mass matrix which are necessary to retain the appropriate consistency order. This can only be alleviated when using DeC time stepping. The second approach is the so–called continuous interior penalty (CIP) method [16, 19, 14], which penalizes the jump of the derivative of the solution across cell boundaries. This stabilization does not affect the mass matrix and, therefore, can be easily combined with mass–matrix free methods. The last is the local projection stabilization [8], which penalizes the $\mathbb{L}^2$ projection of the gradient of the error within the elements. 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.

The goal of this work is to analyze the different methods and their combinations, and give suggestions concerning the most convenient choices in terms of accuracy, stability, and cost. To achieve this objective an important role is played by a spectral analysis which we perform both in the time-continuous and fully discrete cases. The analysis reveals the best parameters (stabilization and CFL coefficients) that can be stably used in practice.

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 introduce the different discretization methods, starting from the choice of the elements, then discussing the stabilization terms and finally presenting the different time integration methods. Sections 3 and 4 are dedicated to the Fourier stability analysis. In Section 5 we provide some elements concerning the extension of the stabilization methods discussed to nonlinear problems, and finally in Section 6 we show numerical results on linear and nonlinear problems. The paper is ended by a summary and overlook on future perspectives in Section 7.

2 Numerical Discretization

We are interested in the approximation of solutions of (1) on a tessellation of non overlapping celles, which we denote by Ω_h. We denote by K the generic cell of Ω_h, and more precicely $\Omega_h = \bigcup K$. We also introduce the set of internal element boundaries (cell faces in 2D and 3D, cell nodes in 1D) of Ω_h, which we denote by $\mathcal{F}_h$. h denotes the characteristic mesh size of Ω_h. The discrete solution is sought in a continuous finite element space $V_h^p = \{v_h \in \mathcal{C}^0(\Omega_h) : \quad v_{h\,|K} \in \mathbb{P}_p(K) \quad \forall K \in \Omega_h\}$. We are interested in particular nodal 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 $u_h(x) = \sum_{j\in\Omega_h} u_j\varphi_j(x)$.

The unstabilized approximation of (1) reads: find $u_h \in V_h^p$ such that for any $v_h \in W_h \subset \mathbb{L}_2(\Omega_h)$

$$\int_\Omega v_h \partial_t u_h dx - \int_\Omega \partial_x v_h f(u_h)\, dx + [v_h f(u_h)]_{\partial\Omega} = 0. \tag{3}$$

The main topic of this paper is the study of the linear stability of (3) and of several stabilized variants using Fourier's analysis. We will therefore assume periodic boundary conditions. We aim at

characterizing the schemes both in terms of their stability range and their accuracy in the fully discrete case, for different choices of the stabilization strategy and of the time stepping. The extensions of these discretization techniques to more dimensions is well known in literature, even if sometimes not uniquely defined. We believe that the one dimensional study can provide useful information also in that context.

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

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

having re-integrated by parts and used the continuity of the approximation, and the periodicity of the boundary conditions to pass to the strong form of the PDE, and with S being a bilinear 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

This method was introduced in [25] (see also [26, 13] 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 (3) with a test function belonging to the space

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

Here τ_K denotes a positive definite stabilization parameter with the dimensions of a time-step that we will assume to be constant for every element. Although other definitions are possible, here we will evaluate this parameter as

$$\tau_K = \delta \frac{h_K}{\|\partial_u f\|_K}$$

where h_K is the cell diameter and the denominator represents a reference value of the flux Jacobian norm on the element K.

The final stabilized variational formulation reads

$$\int_\Omega v_h \partial_t u_h \, dx + \int_\Omega v_h \partial_x f(u_h) \, dx + \underbrace{\sum_{K\in\Omega} \int_K \left(\partial_u f(u_h)\partial_x v_h\right)\tau_K \left(\partial_t u_h + \partial_x f(u_h)\right) \, dx}_{S(v_h,u_h)} = 0. \tag{6}$$

To characterize the accuracy of the method, we can use the consistency analysis discussed e.g. in [6, §3.1.1 and §3.2]. In particular, of a finite element polyomial 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

$$\begin{aligned}\epsilon(\psi_h) :=& \Big| \int_\Omega \psi_h \partial_t (u_h^e - u^e) \, dx - \int_\Omega \partial_x \psi_h (\partial_x f(u_h^e) - \partial_x f(u^e)) \, dx \\ &+ \sum_{K\in\Omega} \sum_{l,m\in K} \frac{\psi_l - \psi_m}{k+1} \int_K \left(\partial_u f(u_h)\partial_x \varphi_i\right)\tau_K \left(\partial_t (u_h^e - u^e) + \partial_x (f(u_h^e) - f(u^e))\right) \, dx \Big| \le C h^{p+1},\end{aligned} \tag{7}$$

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 (2), which is the main focus of the analysis of this paper, and for exact integration with $\tau_K = \tau$, a classical result is obtained in the time continuous case by testing with $v_h = u_h + \tau \, \partial_t u_h$ to obtain [13]

$$\int_{\Omega_h} \partial_t \left(\frac{u_h^2}{2} + \tau^2 \frac{(a\partial_x u_h)^2}{2} \right) + \int_{\Omega_h} a\partial_x \left(\frac{u_h^2}{2} + \tau^2 \frac{(\partial_t u_h)^2}{2} \right) = - \int_{\Omega_h} \tau (\partial_t u_h + a\partial_x u_h)^2. \tag{8}$$

With periodic boundary conditions this easily shows that the norm $|||u|||^2 := \int_{\Omega_h} \frac{u_h^2}{2} + \tau^2 \frac{(a\partial_x u_h)^2}{2} dx$ is non-increasing. The interested reader can refer to [13] for the analysis of some (implicit) fully discrete schemes.

2.1.2 Continuous Interior Penalty - CIP

An alternative, which maintains the structure of the mass matrix, is the continuous interior penalty (CIP) stabilization used in [16, 19, 14]. This method has been develop by E. Burman and P. Hansbo in [15], but it can be seen as a variation of the method originally proposed by Douglas and Dupont [21].

This method stabilizes convection-diffusion-reaction problems by adding a least-squares term based on the jump in the gradient of the discrete solution over element boundaries. With this simple concept we obtain stability for convection-reaction-diffusion problems also in the vanishing viscosity limit.

The method reads

$$\int_{\Omega_h} v_h \partial_t u_h \, dx + \int_{\Omega_h} v_h \partial_x f(u_h) \, dx + \underbrace{\sum_{\mathsf{f} \in \mathcal{F}_h} \int_{\mathsf{f}} \tau_{\mathsf{f}} [\partial_x v_h] \cdot [\partial_x u_h] \, d\Gamma}_{S(v_h, u_h)} = 0, \tag{9}$$

with $[\cdot]$ denoting the jump of a quantity across a face f, and where we recall that $\mathcal{F}_h$ is the collection of internal boundaries (points in 1D), and f are its elements. In one space dimension the last integral reduces to a point evaluation. Although other definitions are possible, we evaluate the scaling parameter in the stabilization as

$$\tau_{\mathsf{f}} = \delta \, h_{\mathsf{f}}^2 \|\partial_u f\|_{\mathsf{f}} \tag{10}$$

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

The advantage of this method is that the formulation remains symmetric, and that the mass matrix can be lumped for efficient time marching if the finite element space allows it. The drawback is a slight increase in the stencil associated to the use of the gradients in all neighboring elements. Note that for higher order approximations [17, 28] suggest the use of jumps in higher derivatives to improve the stability of the method. In this work, we only focus on the gradient jump stabilization. For orders up to 4 this seems to be enough to get $\mathbb{L}_2$ stability and allows the study in more detail the impact of the coefficient δ in the stabilization.

As before, we can easily characterize the accuracy of the method following e.g. [6, §3.1.1 and §3.2], and 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

$$\begin{aligned} \epsilon(\psi_h) := \Big| \int_{\Omega} \psi_h \partial_t (u_h^e - u^e) \, dx - \int_{\Omega} \partial_x \psi_h (\partial_x f(u_h^e) - \partial_x f(u^e) \, dx \\ + \sum_{\mathsf{f} \in \mathcal{F}_h} \int_{\mathsf{f}} \tau_{\mathsf{f}} [\partial_x \psi_h] \cdot [\partial_x (u_h^e - u^e)] \Big| \leq C h^{p+1}, \end{aligned} \tag{11}$$

with C a constant independent of h. The estimate is again a direct consequence of standard approximation results applied to $u_h^e - u^e$ and to its derivatives.

The symmetry of the stabilization makes is rather easy to derive a linear stability estimate. In particular, for a linear flux with periodic boundary conditions we can easily show that

$$\int_{\Omega_h} \partial_t \frac{u_h^2}{2} = - \sum_{\mathsf{f} \in \mathcal{F}_h} \int_{\mathsf{f}} \tau_{\mathsf{f}} [\partial_x u_h]^2 \tag{12}$$

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

2.1.3 Local Projection Stabilization - LPS

Another symmetric stabilization approach is the Local Projection Stabilization (LPS) method. Its original formulation was presented in [10] for Stokes equations. Then, the LPS was successfully extended to transport problems in [11] and applications of local projection methods to Oseen and Navier-Stokes

equations were studied in [12, 8]. The local projection method also aims at providing some control on the fluctuations of the gradient of the discrete solution. The method can be written as follows: find $u_h \in V_h^p$ such that $\forall v_h \in V_h^p$

$$\begin{cases} \int_{\Omega_h} v_h \partial_t u_h \, dx + \int_{\Omega_h} v_h \partial_x f(u_h) \, dx + \underbrace{\sum_{K \in \Omega_h} \int_K \tau_K \partial_x v_h (\partial_x u_h - w_h) \, dx}_{S(v_h, u_h)} = 0, \\ \int_{\Omega_h} v_h w_h \, dx - \int_{\Omega_h} v_h \partial_x u_h \, dx = 0. \end{cases} \tag{13}$$

For this method, the stabilization parameter is evaluated as

$$\tau_K = \delta h_K \|\partial_u f\|_K. \tag{14}$$

Compared to the CIP approach this method has the drawback of requiring the mass matrix inversion in the gradient $\mathbb{L}_2$ projection represented by the second equation in (13). So the possibility of simplifying this operator, and, more precisely, to lump the mass matrix, appear as essential elements for its efficient implementation.

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

$$\begin{aligned} \epsilon(\psi_h) := \Big| \int_\Omega \psi_h \partial_t (u_h^e - u^e) \, dx - \int_\Omega \partial_x \psi_h (\partial_x f(u_h^e) - \partial_x f(u^e)) \, dx \\ + \sum_{K \Omega_h} \int_K \partial_x \psi_h (\partial_x u_h^e - \partial_x u_h^e) + \sum_{K \Omega_h} \int_K \partial_x \psi_h (\partial_x u^e - w_h^e) \Big| \le C h^{p+1}, \end{aligned} \tag{15}$$

where the last term is readily estimated using

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

Finally, for a linear flux and taking $\tau_K = \tau$, as for the SUPG, we can test with $v_h = u_h$ in the first of (13), and $v_h = \tau w_h$ in the second and sum up the result to get (using the periodicity)

$$\int_{\Omega_h} \partial_t \frac{u_h^2}{2} = -\sum_K \int_K \tau (\partial_x u_h - w_h)^2, \tag{16}$$

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

2.2 Finite Element Spaces and Quadrature Rules

We describe the one-dimensional finite element spaces we consider in the Fourier analysis. References to the corresponding multi-dimensional extensions are suggested for completeness where appropriate.

In a one dimensional discretized space Ω_h an element K is a segment, i. e., $K = [x_i, x_{i+1}]$ for some i. 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 with $\{\varphi_1, \ldots, \varphi_N\}$ the basis functions of $\mathbb{P}^p(K)$, and their definitions amounts to describe the degrees of freedom, i.e., the dual basis. In one dimension, $N = p + 1$. We consider two families of polynomials:

1. Lagrange polynomials. They are uniquely defined by the interpolation points ξ_j with $\xi_1 = x_i < \ldots < \xi_j < \ldots < \xi_N = x_{i+1}$. We study two cases
   - Equidistant points: $\xi_j = x_i + j \frac{x_{i+1} - x_i}{p}$ for $j = 0, \ldots, p$,
   - Gauss–Lobatto points: the roots of Legendre polynomial of degree $p + 1$ mapped onto $[x_i, x_{i+1}]$.
2. Bernstein polynomials. Linearly mapping K onto $[0, 1]$ they are defined for $j = 0, \ldots, p$ by

$$B_j(x) = \binom{p}{j} x^{p-j} (1 - x)^j.$$

Bernstein polynomials verify the following properties

$$\sum_{j=0}^{p} B_j(x) \equiv 1, \qquad B_j(x) \geq 0 \quad \forall x \in [0,1].$$

Even if the degrees of freedom associated to this approximation have no physical meaning, we identify them geometrically with the Greville points $\xi_j = \frac{j}{p}$.

The use of different polynomial basis functions leads to different properties. Let us remark that the evaluation of integrals is done by Gaussian quadrature formulae, because of their efficiency. If Gauss points are used in the discretization of the polynomials, the same points will be used in the quadrature formula. Thanks to this, we see that for Lagrange polynomials defined on Gauss quadrature points

$$\int_{x_i}^{x_{i+1}} \varphi_l(x)\varphi_j(x)\, dx = (x_{i+1}-x_i)\omega_l \delta_l^j \quad \text{with } \omega_l := \frac{1}{(x_{i+1}-x_i)} \int_{x_i}^{x_{i+1}} \varphi_l^2(x)\, dx > 0.$$

This leads to a diagonal local mass matrix

$$\mathbb{M}^i_{l,j} = \left(\int_{x_i}^{x_{i+1}} \varphi_l(x)\varphi_j(x)\, dx \right).$$

This does not hold for Lagrange polynomials defined on equidistant points or the Bernstein polynomials.

Another important property that we need to effectively apply the DeC method of [3] is the positivity of the lumped mass matrix entries, i.e., $\mathbb{D}_{k,k} := \sum_{j=0}^{N} \int_{x_i}^{x_{i+1}} \varphi_j \varphi_k \, dx = \int_{x_i}^{x_{i+1}} \varphi_k \, dx > 0$. The positivity of these values is trivially verified for Bernstein polynomials and for Lagrange polynomials with matching quadrature formulae. In the case of equispaced points Lagrangian polynomials, the lowest degree ($p \leq 7$ in one dimension) they also verify the positivity of the lumped matrix. This is not true in the case of two dimensional problems and triangular meshes, where already for degree $p = 2$ we have nonpositive values in the diagonal of the lumped matrix. This mainly motivated the choice of Bernstein polynomials, as well as the Lagrange interpolation with the Gauss–Lobatto points.

In the following we will use the wording

- *basic* elements for Lagrangian polynomials on equispaced points with Gauss–Legendre quadrature;
- *cubature* elements for Lagrangian polynomials on on Gauss–Lobatto points and quadrature rule using the same points;
- *Bernstein* elements for Bernstein polynomials with Gauss–Legendre quadrature.

2.3 Time Integration

The finite element semi-discrete equations constitute a coupled system of ordinary differential equations which can be written as

$$\mathbb{M}\frac{dU}{dt} = \mathtt{r}(t) \tag{17}$$

where U is the collection of all the degrees of freedom, $\mathbb{M}$ and $\mathtt{r}$ are the global mass matrix and right-hand side term defined in the previous sections through the element definition and stabilization terms. We must 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 strategies: explicit Runge–Kutta (RK) methods and their strong stability preserving (SSP) variant; Deferred Correction, 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 can be described by the following one step procedure

$$\begin{aligned} U^{(0)} &:= U^n, \\ U^{(s)} &:= U^n + \Delta t \sum_{j=0}^{s-1} \alpha_j^s \mathbb{M}^{-1} \mathtt{r}(U^{(j)}) \quad s = 1, \dots, S, \\ U^{n+1} &:= U^n + \Delta t \sum_{s=0}^{S} \beta_s \mathbb{M}^{-1} \mathtt{r}(U^{(s)}). \end{aligned} \tag{18}$$

Here, we use the superscript n to indicate the timestep and the superscript in brackets (s) to denote the stage of the method. 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's tableau can be found in Appendix A in Table 8.

A particular case is that of SSPRK methods introduced in [37]. They are essentially convex combinations of forward Euler steps, and can be rewritten as follows

$$\begin{aligned} U^{(0)} &:= U^n, \\ U^{(s)} &:= \sum_{j=0}^{s-1} \left(\gamma_j^s U^{(j)} + \Delta t \mu_j^s \mathbb{M}^{-1} \mathbf{r}(U^{(j)}) \right) \quad s = 1, \dots, S, \\ U^{n+1} &:= U^{(S)}, \end{aligned} \tag{19}$$

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 [37], the third order SSPRK(4,3) presented in [35, Page 189], and the fourth order SSPRK(5,4) defined in [35, Table 3]. For complete reproducibility of the results, we put all their Butcher' tableaux in Appendix A in Table 9.

2.3.2 The *Deferred Correction* scheme

Deferred correction methods were originally introduced in [22] as explicit solvers of ODEs, but soon implicit [31] or positivity preserving [34] versions and extensions to PDE solvers [1] were studied. In [1, 7, 3] the method is also used to avoid the inversion of the mass matrix, applying a mass lumping and adding correction iterations to regain the order of convergence. This is only achievable when the lumped matrix have only positive values on its diagonal. Hence, the use of *Bernstein* polynomials is recommended in [1], but also the *cubature* elements can serve the purpose.

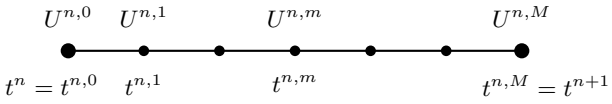

Figure 1: Subtimesteps inside the time step $[t^n, t^{n+1}]$

Consider a discretization of each timestep into M subtimesteps as in Figure 1. For each subtimestep the goal is to find the solution of the integral form of the semidiscretized ODE (17) as

$$\mathbb{M}\left(U^{n,m} - U^{n,0}\right) - \int_{t^{n,0}}^{t^{n,m}} \mathbf{r}(U(s))ds \approx \mathcal{L}^2(\underline{U})^m := \mathbb{M}\left(U^{n,m} - U^{n,0}\right) - \Delta t \sum_{z \in [\![0,M]\!]} \rho_z^m \mathbf{r}(U^{n,z}) = 0, \tag{20}$$

with $\underline{U} = \left(U^{n,0}, \dots, U^{n,M}\right)$ and having used high order quadrature with points $t^{n,0}, \dots, t^{n,M}$ and weights ρ_z^m for every different subinterval (see [1, 7, 3] for details). The algebraic system $\mathcal{L}^2(\underline{U}^*) = 0$ is in general implicit and nonlinear and may not be easy to solve. The DeC procedure approximates iteratively this solution by successive corrections relying on a a low order easy–to–invert operator $\mathcal{L}^1$. This operator is typically obtained using an explicit timestepping and a lumped mass matrix, i.e.,

$$\mathbb{M}\left(U^{n,m} - U^{n,0}\right) - \int_{t^{n,0}}^{t^{n,m}} \mathbf{r}(U(s))ds \approx \mathcal{L}^1(\underline{U})^m := \mathbb{D}\left(U^{n,m} - U^{n,0}\right) - \Delta t \beta^m \mathbf{r}(U^{n,0}) = 0. \tag{21}$$

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 A. Denoting with the superscript (k) index the iteration step, we describe the DeC algorithm as

$$U^{n,m,(0)} := U^n \qquad m = 0, \dots, M, \tag{22a}$$

$$U^{n,0,(k)} := U^n \qquad k = 0, \dots, K, \tag{22b}$$

$$\mathcal{L}^1(\underline{U}^{(k)}) = \mathcal{L}^1(\underline{U}^{(k-1)}) - \mathcal{L}^2(\underline{U}^{(k-1)}) \qquad k = 1, \dots, K, \tag{22c}$$

$$U^{n+1} := U^{n,M,(K)}. \tag{22d}$$

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 the low order operator, the method has for each iteration a cost equivalent essentially to the assembly of the right hand side, whatever the complexity of the mass matrix appearing in $\mathcal{L}^2$. The only requirement that is necessary for the DeC approach is the invertibility of the lumped mass matrix, which limits its application to equispaced Lagrange elements only to the degrees for which this is the case, and to other choices as the *Bernstein* and *cubature* elements introduced earlier.

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 (22c) as

$$U^{n,m,(k+1)} = U^{n,m,(k)} - \mathbb{D}^{-1}\mathbb{M}\left(U^{n,m,(k)} - U^{n,0,(k)}\right) + \sum_{j=0}^{M} \Delta t \rho_j^m \mathbb{D}^{-1} \mathtt{r}(U^{n,j,(k)}). \tag{23}$$

Comparing with (19), 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.

3 Fourier Analysis

The dispersion and the stability properties of numerical methods can be shown by means of a spectral analysis. We will focus on the linear case (2) with periodic boundary conditions:

$$\partial_t u + a\partial_x u = 0, \quad x \in [0, 1]. \tag{24}$$

The main idea is to investigate the semi and fully discrete evolution of periodic waves represented by the the ansatz

$$u = Ae^{i(kx-\xi t)} = Ae^{i(kx-\omega t)}e^{\epsilon t} \qquad \text{with} \quad \xi = \omega + i\epsilon, \quad i = \sqrt{-1}. \tag{25}$$

Here, ϵ denotes the damping rate, while the wavenumber is denoted by $k = 2\pi/L$ with L the wavelength. We recall that the phase velocity defined as

$$C = \frac{\omega}{k} \tag{26}$$

represents the celerity with which waves propagate in space, and it is in general a function of the wavenumber. Substituting (25) in the advection equation (24) leads to the well known result

$$C = a \quad \text{and} \quad \epsilon = 0. \tag{27}$$

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 different stabilization methods and time integration. This will also allow to investigate the parametric stability with respect to the time step (CFLnumber) 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. For completeness, the study is performed first in the semi-discrete time continuous case in Section 3.1. We the consider the fully discrete schemes in Section 3.2.

3.1 Preliminaries and time continuous analysis

The Fourier analysis for numerical schemes on the periodic domain is based on Parseval theorem.

Theorem 3.1 (Parseval)**.** *Let $\hat{u}(k) := \int_0^1 u(x)e^{-i2\pi kx}dx$ for $k \in \mathbb{Z}$ be the Fourier modes of the function u. The $\mathbb{L}_2$ norms of the function u and of the Fourier modes coincide, i.e.,*

$$\int_0^1 u^2(x)dx = \sum_{k\in\mathbb{Z}} |\hat{u}(k)|^2. \tag{28}$$

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 (25) we can write local equations coupling degrees of freedom belonging to neighbouring cells through a multiplication by the factor of $e^{i\theta}$ representing the shift in space along the oscillating solution. The dimensionless coefficient

$$\theta := k\Delta x \tag{29}$$

is a discrete reduced wave number which naturally appears all along the analysis. Formally replacing the ansatz in the scheme we end up with a dense algebraic problem of dimension p (the polynomial degree) reading in the time continuous case

$$(24) \text{ and } (25) \quad \Rightarrow \quad -i\xi\mathbb{M}\mathbf{U} + a\mathcal{K}_x\mathbf{U} = 0 \tag{30}$$

$$\text{with} \quad (\mathbb{M})_{ij} = \int_\Omega \phi_i\phi_j dx, \qquad (\mathcal{K}_x)_{ij} = \int_\Omega \phi_i\partial_x\phi_j dx + S(\phi_i,\phi_j), \tag{31}$$

with ϕ_j the finite element basis functions and $\mathbf{U}$ the array of all the degrees of freedom. Although system (30) is in general a global eigenvalue problem, we can reduce its complexity by exploiting more explicitly the ansatz (25). More exactly, we can introduce elemental vectors of unknowns $\widetilde{\mathbf{U}}_K$, which, for continuous finite elements, are a arrays of p degrees of freedom including only one of the two boundary nodes. Using the periodicity of the solution and denoting by $K \pm 1$ the neighboring elements, we have

$$\widetilde{\mathbf{U}}_{K\pm1} = e^{\pm\theta}\widetilde{\mathbf{U}}_K. \tag{32}$$

This allows to show that (30) is equivalent to a compact system (we drop the subscript $_K$ as they system is equivalent for all cells)

$$-i\xi\widetilde{\mathbb{M}}\widetilde{\mathbf{U}} + a\widetilde{\mathcal{K}}_x\widetilde{\mathbf{U}} = 0, \tag{33}$$

where the matrices $\widetilde{\mathbb{M}}$ and $\widetilde{\mathcal{K}}$ are readily obtained from the elemental discretization matrices by using (32).

As shown in [36] some particular cases can be easily studied analytically. For example for the semidiscretized $\mathbb{P}_1$ CG scheme without stabilization one easily finds that

$$\frac{\omega}{k} = a\frac{\sin(\theta)}{\theta}\frac{3}{2+\cos(\theta)} \quad \text{and} \quad \epsilon = 0. \tag{34}$$

As the degree of the approximation increases, so does the size of the eigenvalue problem. For the non stabilized CG $\mathbb{P}_2$ scheme we can still find an analytical solution associated to the quadratic equation (cf also [36]) reading

$$\frac{\omega_{1,2}}{k} = a\frac{4\sin(\theta) \pm 2\sqrt{40\sin^2(\frac{\theta}{2}) - \sin^2(\theta)}}{\theta(\cos(\theta)-3)}. \tag{35}$$

For more general cases, the study needs to be performed numerically.

Defining with $\lambda_i(\theta)$ the eigenvalues of (33), $\omega_i(\theta) = \mathrm{Im}(\lambda_i(\theta))$ and $\epsilon_i(\theta) = -\mathrm{Re}(\lambda_i(\theta))$ are the respective phase and damping coefficients of each mode of the solution. In practice, we solve numerically the eigenvalue problem (33) for $\theta = k\Delta x_p = \frac{2\pi}{N_x}$ varying in $[0,\pi]$, where N_x is the number of the nodes in each wavelength and $\Delta x_p = \Delta x/p$ is the average distance between degrees of freedom. However, to satisfy the Nyquist stability criterion, it is necessary to have $\Delta x_p \leq \frac{L}{2}$, with L the wavelength.

As an example, in Figure 2 we plot ω and ϵ and we see that CG scheme does not have diffusive terms, or, in other words, there is no damping ($\epsilon = 0$) in the CG scheme. For clarity of the pictures, we plot in Figure 2 only the principal eigenvalue of each system ($p = 1, 2, 3$), i. e., the one that minimizes $|\omega_i - ak|$. As expected, with $\mathbb{P}_1$ elements, the scheme is more dispersive than with $\mathbb{P}_2$ or $\mathbb{P}_3$ elements, while, for all of them, there is no dissipation, since the scheme is not stabilized and there is no time discretization.

We apply the same analysis to stabilized methods. The results obtained with SUPG, CIP and LPS stabilizations lead to an almost identical result shown in Figure 3 (reporting the LPS data). The interested reader can access all the other plots online [30]. 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.

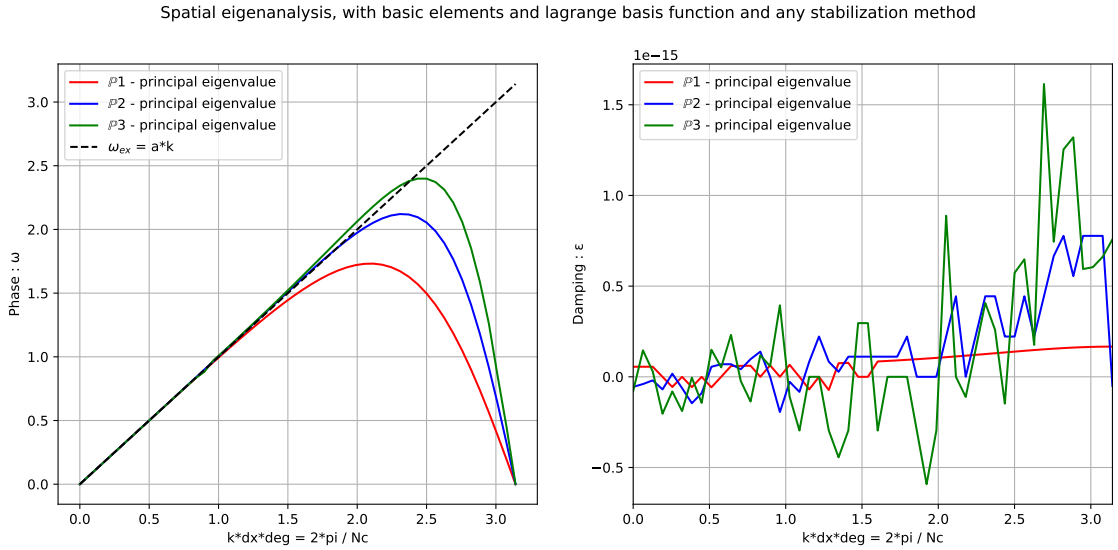

Figure 2: Phase ω (left) and amplification ϵ (right) with *basic* elements without stabilization for $\mathbb{P}_1, \mathbb{P}_2$ and $\mathbb{P}_3$.

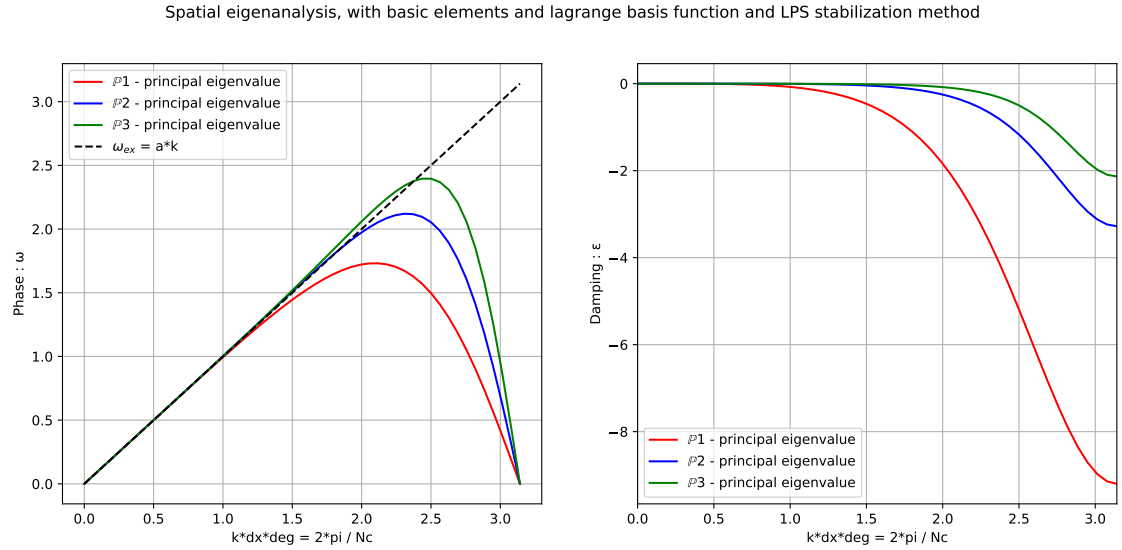

Figure 3: Phase ω (left) and amplification ϵ (right) with *basic* elements with LPS stabilization for $\mathbb{P}_1, \mathbb{P}_2$ and $\mathbb{P}_3$.

3.2 Fully discrete analysis

3.2.1 Methodology

We analyze now the fully discrete schemes obtained using the RK, SSPRK and DeC time marching methods presented in Section 2.3. Let us consider as an example the SSPRK schemes (19). If we define as $A := \mathbb{M}^{-1}\mathcal{K}_x$ we can write the schemes as follows

$$\begin{cases} \mathbf{U}^{(0)} := & \mathbf{U}^n \\ \mathbf{U}^{(s)} := & \sum_{j=0}^{s-1} \left(\gamma_{sj} \mathbf{U}^{(j)} + \Delta t \mu_{sj} A \mathbf{U}^{(j)} \right), \quad s \in [\![1, S]\!], \\ \mathbf{U}^{n+1} := & \mathbf{U}^{(S)}. \end{cases} \tag{36}$$

Expanding all the stages, we can obtain the following formulation:

$$\mathbf{U}^{n+1} = \mathbf{U}^{(0)} + \sum_{j=1}^{S} \nu_j \Delta t^j A^j \mathbf{U}^{(0)} = \left(\mathcal{I} + \sum_{j=1}^{S} \nu_j \Delta t^j A^j \right) \mathbf{U}^n, \tag{37}$$

where coefficients ν_j in (37) are obtained as combination of coefficient γ_{sj} and μ_{sj} in (36) 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 (32) one easily shows that the problem can be written in terms of the local $p \times p$ matrices $\widetilde{A} := a\widetilde{\mathbb{M}}^{-1}\widetilde{\mathcal{K}}_x$ and in particular that

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

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

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

For the DeC method we can proceed with the same analysis transforming also the other involved matrices into their Fourier equivalent ones. Using (23) 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 represents the evolution in one timestep of the Fourier modes for all the p 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. Furthermore, we notice that the matrix G depends not only on $\theta, \Delta x$ and Δt, but also on at the stabilization coefficients τ_K. Hence, the proposed analysis should contain an optimization process also along the stabilization parameter. With the notation of section §2, we will in particular set

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

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

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

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 we obtain a positive damping coefficient $\epsilon(\theta) > 10^{-12}$ for any value of the reduced wavenumber θ (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 the product between error and computational cost. In the following we describe the 3 strategies to find the best parameters couples (CFL,δ):

1. *maximize the CFL in the stable area*;
2. *minimize 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

$$\left[\frac{u(t)-u_{ex}(t)}{u_{ex}(t)}\right]^2 = \left[e^{\epsilon t - it(\omega-\omega_{ex})} - 1\right]^2 \tag{38}$$

$$= \left[e^{\epsilon t}\cos(t(\omega-\omega_{ex})) - 1\right]^2 + \left[e^{\epsilon t}\sin(t(\omega-\omega_{ex}))\right]^2 \tag{39}$$

$$= e^{2\epsilon t} - 2e^{\epsilon t}\cos(t(\omega-\omega_{ex})) + 1. \tag{40}$$

Here, we denote with ϵ and ω the damping and phase of the *principal* mode. 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

$$\left[\frac{u(t)-u_{ex}(t)}{u_{ex}(t)}\right]^2 \approx \underbrace{\left[e^{\epsilon t}-1\right]^2}_{\text{Damping error}} + \underbrace{e^{\epsilon t}t^2\left[\omega-\omega_{ex}\right]^2}_{\text{Dispersion error}}. \tag{41}$$

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,

$$\eta_u(\omega,\epsilon)^2 := \frac{3}{2\pi}\left[\int_0^{\frac{2\pi}{3}}(e^{\epsilon}-1)^2 dk + \int_0^{\frac{2\pi}{3}} e^{\epsilon}(\omega-\omega_{ex})^2 dk\right]. \tag{42}$$

Recalling that $\epsilon = \epsilon(k\Delta x, \mathrm{CFL}, \delta)$ and $\omega = \omega(k, \Delta x, \mathrm{CFL}, \delta)$ and $\omega_{ex} = ak$, 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. Finally, we seek the couple $(\mathrm{CFL}^*, \delta^*)$ allowing to solve

$$(\mathrm{CFL}^*,\delta^*) := \arg\max_{\mathrm{CFL}}\left\{\eta(\omega(\mathrm{CFL},\delta)),\epsilon(\mathrm{CFL},\delta)) < \mu \min_{(\mathrm{CFL},\delta)\text{stable}} \eta(\omega(\mathrm{CFL},\delta),\epsilon(\mathrm{CFL},\delta))\right\}. \tag{43}$$

3. *minimize the dispersion error η_ω while maximizing the CFL in the stable area.* In particular we set in this case

$$\eta_\omega^2(\omega) := \int_0^{\frac{2\pi}{3}}\left(\frac{\omega-\omega_{ex}}{\omega_{ex}}\right)^2 dk. \tag{44}$$

As before we choose the optimal parameters from (43).

For the second and third strategies, 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 1.3 providing a sufficient flexibility to obtain results of practical usefulness, which we verified in numerical computations as we will see later.

In the following we will compare all the methods with these error measures, in order to suggest the best possible schemes between the proposed ones.

4 Results of the fully discrete spectral analysis

The typical results reported in Figures 4 to 8 show in the plane (δ, CFL) the unstable (crossed) and stable regions, and with colored symbols the optimal points corresponding to the three strategies introduced earlier. In case of ambiguity, the point with maximum δ is marked in the figures. A summary of the results for all combinations of schemes is provided in Tables 1 to 3.

Before commenting these results we remark that some of the schemes are equivalent. For example without mass lumping *Bernstein* and *basic* elements are the same up to an orthogonal change of variable. This is not the case when using DeC due to the difference in lumped mass matrices. Similarly,

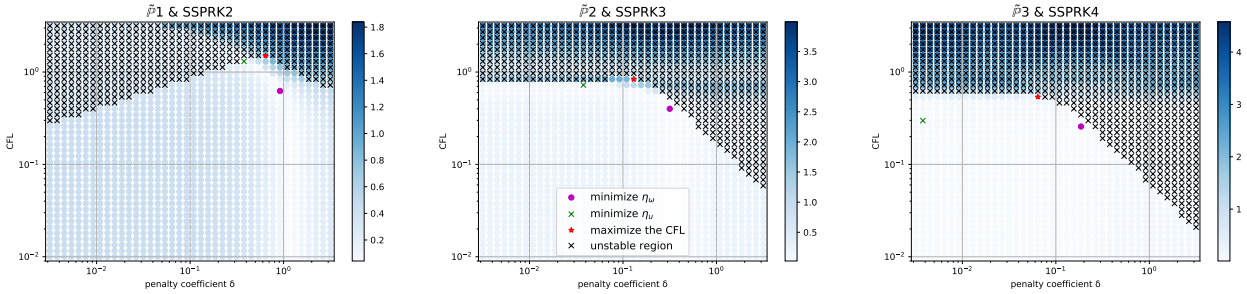

Figure 4: Computation of optimal parameters according to errors η_ω and η_u. (CFL, δ) plot of η_u (blue scale) and instability area (black crosses) for cubature elements SSPRK scheme with SUPG stabilization method. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$. The purple circle is the optimizer of η_u, the green cross is the optimizer of η_ω, the red star is the maximum stable CFL.

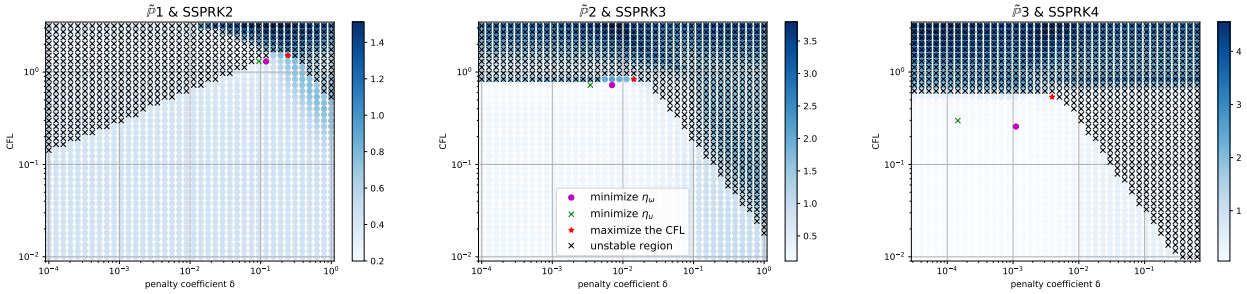

Figure 5: Computation of optimal parameters according to errors η_ω and η_u. (CFL, δ) plot of η_u (blue scale) and instability area (black crosses) for cubature elements SSPRK scheme with CIP stabilization method. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$. The purple circle is the optimizer of η_u, the green cross is the optimizer of η_ω, the red star is the maximum stable CFL.

the mass matrix used for *cubature* elements is already diagonal, which makes the DeC procedure entirely equivalent to the RK scheme with Butcher tableau corresponding to the quadrature weights of the DeC. Only for SUPG a difference is observed due to the contributions to the mass matrix of the stabilization.

Concerning the plots, it is interesting to remark the appearance of four different structures which have an impact on the practical usefulness of some of the combinations.

- The first kind of structures are associated to schemes presenting V-shaped stability regions. We can observe these on Figures 4 and 5, for $p = 1$. This shape requires a very careful choice of the stability parameter as small perturbations of δ may lead, for a given CFL, to an unstable behavior. Generally, lowering the CFL increases somewhat the robustness allowing more flexibility in the choice of δ. We highlight that this type of topology is common to all the second order schemes, as well as to all DeC schemes with *basic* and *Bernstein* elements for degree $p \geq 2$.
- Another structure typically observed is an L-shaped stability region as in Figures 4 and 5 for $p = 2, 3$. This shape is characterized by a CFL bound $\mathrm{CFL} \leq C_1$ and a one–sided bound on the stabilization coefficient $\delta \leq C_2 \mathrm{CFL}^{C_3}$, and it much more robust concerning the choice of the stability parameter as all values below a certain maximum are stable. Most of the schemes with $p \geq 2$, besides those listed in the first group, belong to this category.
- The third kind of structures involve "broom"- or "box"-shaped stability domains. In the first case we observe two clear bounds $\delta \geq C_1 \mathrm{CFL}^{C_2}$ and $\delta < C_3$ plus a small stable stripe with higher

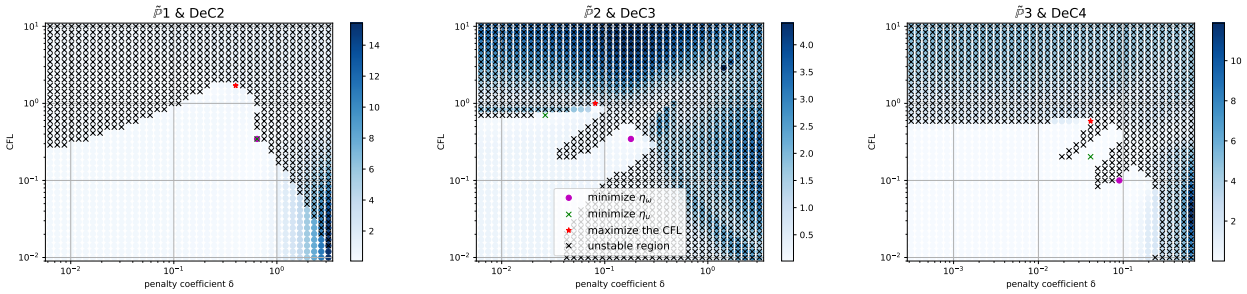

Figure 6: Computation of optimal parameters according to errors η_ω and η_u. (CFL, δ) plot of η_u (blue scale) and instability area (black crosses) for cubature elements DeC scheme with SUPG stabilization method. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$. The purple circle is the optimizer of η_u, the green cross is the optimizer of η_ω, the red star is the maximum stable CFL.

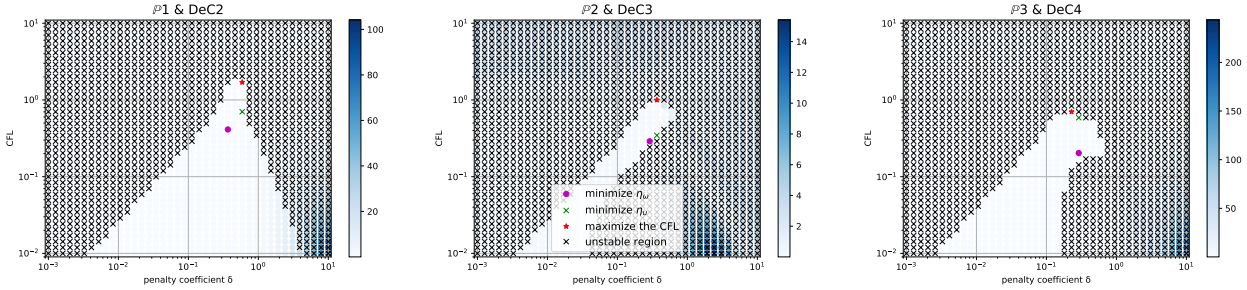

Figure 7: Computation of optimal parameters according to errors η_ω and η_u. (CFL, δ) plot of η_u (blue scale) and instability area (black crosses) for Bernstein elements DeC scheme with SUPG stabilization method. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$. The purple circle is the optimizer of η_u, the green cross is the optimizer of η_ω, the red star is the maximum stable CFL.

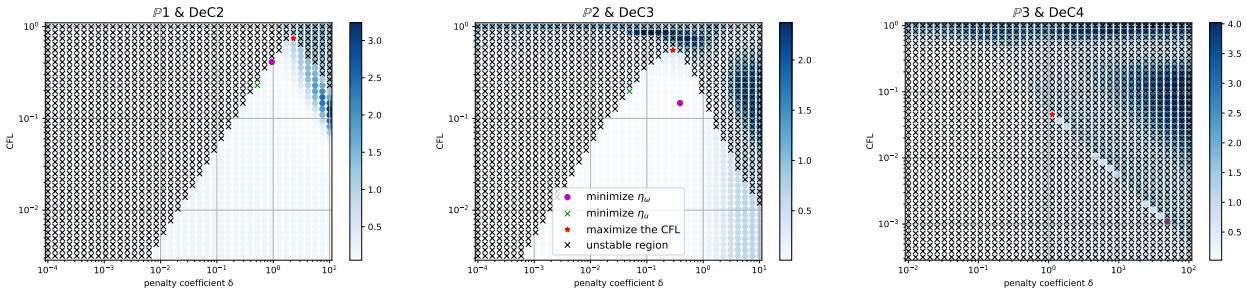

Figure 8: Computation of optimal parameters according to errors η_ω and η_u. (CFL, δ) plot of η_u (blue scale) and instability area (black crosses) for basic elements DeC scheme with LPS stabilization method. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$. The purple circle is the optimizer of η_u, the green cross is the optimizer of η_ω, the red star is the maximum stable CFL.

CFL $> (C_3/C_1)^{1/C_2}$ and $\delta > C_3$. This is for example visible in Figure 7. In the second case, see for example Figure 6, we also have two bounds of the type CFL $\geq C_1$ and $\delta < C_2$, with an additional stable stripe outside these bounds. The problem with this type of methods is that the optimal parameters, *viz.* those involving the highest CFL, are within a stripe which means that instability may be introduced by lowering the CFL[1] . For applications involving multiscale problems, or variable mesh sizes this is clearly unacceptable in practice. Schemes showing this sort of behaviors are all the SUPG schemes with DeC time stepping, and with $p \geq 2$, for which we indicate good values (CFL, δ) in Table 4.

- Finally, the DeC scheme with *basic* elements and $p = 3$ shows essentially everywhere instability for CIP and LPS stabilization. The study finds some very thin oblique stripes of stability, but they are not wide enough to find stable regions. See Figure 8 for an example.

Element &		No stabilization			SUPG		
Time scheme		$p=1$	$p=2$	$p=3$	$p=1$	$p=2$	$p=3$
Basic	RK	/	0.389	0.389	0.624 (0.464)	0.492 (0.07)	0.389 (0.027)
	SSPRK	/	0.492	0.389	0.889 (0.464)	0.554 (0.089)	0.438 (0.027)
	DeC	/	/	/	1.701 (0.588)	0.492 (0.229)[1]	0.492 (0.089)[1]
Cub.	RK	/	0.492	0.492	0.971 (0.767)	0.624 (0.13)	0.464 (0.064)
	SSPRK	/	0.624	0.492	1.512 (0.642)	0.838 (0.13)	0.538 (0.064)
	DeC	/	0.492	0.492	1.701 (0.398)	1.0 (0.081)[1]	0.588 (0.041)[1]
Bern.	RK	/	0.389	0.389	0.624 (0.464)	0.492 (0.07)	0.389 (0.027)
	SSPRK	/	0.492	0.389	0.889 (0.464)	0.554 (0.089)	0.438 (0.027)
	DeC	/	/	/	1.701 (0.588)	1.0 (0.367)[1]	0.702 (0.229)[1]

Element &		LPS			CIP		
Time scheme		$p=1$	$p=2$	$p=3$	$p=1$	$p=2$	$p=3$
Basic	RK	0.681 (0.767)	0.478 (0.077)	0.378 (0.032)	0.838 (0.094)	0.538 (5.54e-03)	0.4 (8.38e-04)
	SSPRK	1.093 (0.767)	0.605 (0.109)	0.425 (0.038)	1.125 (0.119)	0.624 (7.02e-03)	0.464 (6.61e-04)
	DeC	0.744 (2.29)	0.554 (0.289)	/	0.838 (0.289)	0.588 (0.02)	/
Cub.	RK	1.093 (0.702)	0.681 (0.143)	0.538 (0.049)	0.971 (0.191)	0.723 (0.011)	0.538 (1.84e-03)
	SSPRK	1.557 (1.0)	0.863 (0.17)	0.605 (0.049)	1.512 (0.242)	0.838 (0.014)	0.538 (3.93e-03)
	DeC	1.093 (0.702)	0.681 (0.143)	0.538 (0.049)	0.971 (0.191)	0.723 (0.011)	0.538 (1.84e-03)
Bern.	RK	0.681 (0.767)	0.478 (0.077)	0.378 (0.032)	0.838 (0.094)	0.538 (5.54e-03)	0.4 (8.38e-04)
	SSPRK	1.093 (0.767)	0.605 (0.109)	0.425 (0.038)	1.125 (0.119)	0.624 (7.02e-03)	0.464 (6.61e-04)
	DeC	0.744 (2.29)	0.052 (0.215)	0.109 (0.215)	0.838 (0.289)	0.059 (0.016)	0.119 (7.02e-03)

Table 1: Optimized CFL and penalty coefficient δ in parenthesis, only maximizing CFL

4.1 Dispersion and damping

In Figures 9 and 10 are represented the phase and the damping of the principal eigenvalue depending on $\theta = k\Delta x = \frac{2\pi}{N_x}$ for few schemes (*cubature* DeC LPS and *Bernstein* SSPRK CIP), using the best parameters (CFL, δ) found in the previous analysis with the optimization of η_u. As before, we notice that the mode for $p = 1$ is particularly dispersive. Nevertheless, the frequencies on which the scheme is dispersive are also much damped as we see in the right plots. For higher order methods, the phase ω of the principal mode is closer to the exact phase $\omega_{ex} = ak$ in the left figures. We observe that the principal mode of higher order methods is much more precise in terms of dispersion than the first order one, but also less damped in the low frequency area $\theta \geq \frac{2\pi}{3}$.

For completeness, a comparison of damping and phase coefficients for DeC and SSPRK for all the stabilization techniques and elements can be found in Appendix B. There we used the (CFL,δ) coefficients found by minimizing η_u in Table 2, and we try also to compare the obtained results. Nevertheless,

[1]These values do not allow to decrease the CFL

Element &		No stabilization			SUPG		
Time scheme		$p=1$	$p=2$	$p=3$	$p=1$	$p=2$	$p=3$
Basic	RK	/	0.151	0.191	0.389 (0.089)	0.17 (2.57e-03)	0.215 (8.38e-03)
	SSPRK	/	0.191	0.242	0.492 (0.089)	0.215 (2.57e-03)	0.273 (5.22e-03)
	DeC	/	/	/	0.702 (0.588)	0.143 (0.022)	0.024 (0.013)
Cub.	RK	/	0.492	0.242	0.971 (0.538)	0.624 (0.045)	0.222 (0.019)
	SSPRK	/	0.624	0.307	1.304 (0.378)	0.723 (0.038)	0.298 (3.78e-03)
	DeC	/	0.492	0.242	0.346 (0.642)	0.702 (0.026)	0.203 (0.041)
Bern.	RK	/	0.151	0.191	0.389 (0.089)	0.17 (2.57e-03)	0.215 (8.38e-03)
	SSPRK	/	0.191	0.242	0.492 (0.089)	0.215 (2.57e-03)	0.273 (5.22e-03)
	DeC	/	/	/	0.702 (0.588)	0.346 (0.367)[1]	0.588 (0.289)[1]

Element &		LPS			CIP		
Time scheme		$p=1$	$p=2$	$p=3$	$p=1$	$p=2$	$p=3$
Basic	RK	0.335 (0.077)	0.165 (3.78e-03)	0.209 (0.013)	0.4 (0.011)	0.165 (1.60e-04)	0.222 (2.03e-04)
	SSPRK	0.478 (0.077)	0.209 (3.78e-03)	0.265 (9.15e-03)	0.624 (0.011)	0.191 (2.03e-04)	0.257 (3.26e-04)
	DeC	0.229 (0.522)	0.197 (0.049)	/	0.346 (0.077)	0.203 (2.42e-03)	/
Cub.	RK	0.863 (0.492)	0.605 (0.041)	0.235 (0.012)	0.971 (0.119)	0.624 (3.46e-03)	0.257 (1.13e-04)
	SSPRK	1.23 (0.412)	0.767 (0.041)	0.298 (4.12e-03)	1.304 (0.094)	0.723 (3.46e-03)	0.298 (1.45e-04)
	DeC	0.863 (0.492)	0.605 (0.041)	0.235 (0.012)	0.971 (0.119)	0.624 (3.46e-03)	0.257 (1.13e-04)
Bern.	RK	0.335 (0.077)	0.165 (3.78e-03)	0.209 (0.013)	0.4 (0.011)	0.165 (1.60e-04)	0.222 (2.03e-04)
	SSPRK	0.478 (0.077)	0.209 (3.78e-03)	0.265 (9.15e-03)	0.624 (0.011)	0.191 (2.03e-04)	0.257 (3.26e-04)
	DeC	0.229 (0.522)	0.052 (0.215)	0.109 (0.215)	0.346 (0.077)	0.059 (0.016)	0.119 (7.02e-03)

Table 2: Optimized CFL and penalty coefficient δ in parenthesis, minimizing η_u

we must remark that the different CFLs used for different schemes do not allow a direct comparison.

The different strategies lead to different values of best CFL and δ. In general, the most reliable is the one that optimizes η_u. Looking at Table 2, we can compare the different elements, stabilization terms and time integration techniques and obtain some conclusions.

- In general SSPRK time integration methods allow to use higher CFL with respect to both classical RK methods and DeC.
- With *cubature* elements we can use larger CFLs conditions than with *basic* and *Bernstein* elements.
- Concerning efficiency, we do not observe any impact of the choice of the stabilization approach on the magnitude of the allowed CFL. Other factors are much more relevant in this respect. For example, for SUPG we need to stress the advantage of using DeC w.r.t. the possibility of avoiding the inversion of the non-diagonal mass matrix required by the full consistency of the method. For CIP the larger stencil and non-local data structure gives a small overhead, and, for LPS, the gradient projection favors clearly *cubature* elements for which this phase requires no matrix inversion.
- Some combinations produce very unstable schemes. As remarked also before, DeC with high order *basic* elements may have problems in the mass lumping, and we can see an example with the LPS and CIP stabilization.
- DeC with SUPG stabilization leads to stability regions that are not comprehending all the CFLs smaller than the one inside the region, for a fixed δ. This is very dangerous, for instance when doing mesh adaptation algorithms, hence, we marked with an asterisk in Tables 1 to 3 such schemes and we put in Table 4 reliable values of (CFL,δ).

5 A note on nonlinear stability

The stability analysis performed before holds only for linear problems. For nonlinear ones the original ansatz of supposing that the solutions can be decomposed orthogonally into waves that propagate at

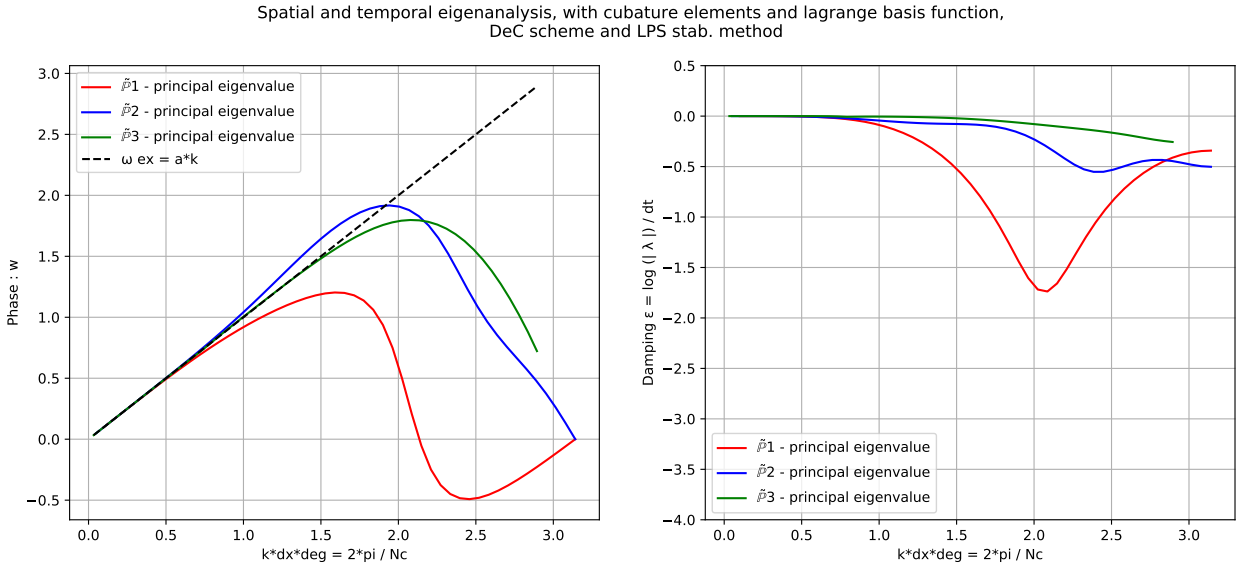

Figure 9: Comparison of dispersion in the fully discrete case, using coefficients from 2, *cubature* elements, DeC scheme and LPS stabilization method. $\mathbb{P}_1$ elements in red, $\mathbb{P}_2$ elements in blue and $\mathbb{P}_3$ elements in green. The phase ω of the principal eigenvalues is on the left and the damping ϵ_i on the right

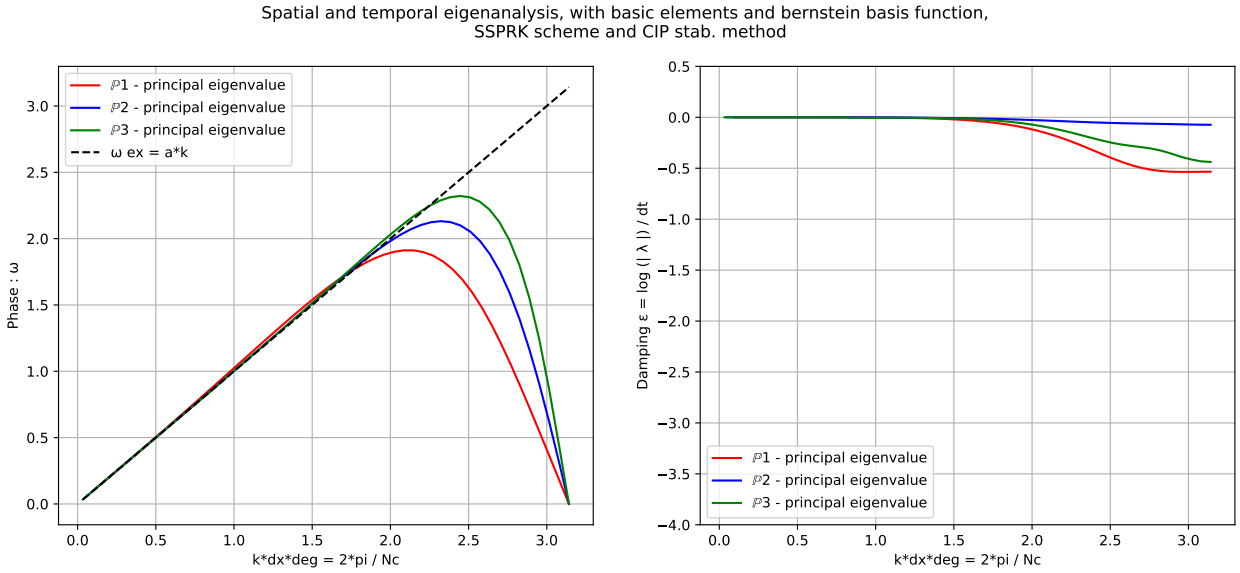

Figure 10: Comparison of dispersion in the fully discrete case, using coefficients from 2, *Bernstein* elements, SSPRK scheme and CIP stabilization method. $\mathbb{B}_1$ elements in red, $\mathbb{B}_2$ elements in blue and $\mathbb{B}_3$ elements in green. The phase ω of the principal eigenvalues is on the left and the damping ϵ_i on the right.

Element &		No stabilization			SUPG		
Time scheme		$p=1$	$p=2$	$p=3$	$p=1$	$p=2$	$p=3$
Basic	RK	/	0.191	0.307	0.059 (0.289)	0.191 (0.027)	0.307 (0.044)
	SSPRK	/	0.242	0.307	0.084 (0.289)	0.242 (0.027)	0.346 (0.035)
	DeC	/	/	/	0.412 (0.367)	0.242 (0.089)[1]	0.017 (0.113)[1]
Cub.	RK	/	0.492	0.389	0.538 (0.767)	0.298 (0.316)	0.165 (0.156)
	SSPRK	/	0.624	0.492	0.624 (0.915)	0.4 (0.316)	0.257 (0.186)
	DeC	/	0.492	0.389	0.346 (0.642)	0.346 (0.179)[1]	0.1 (0.09)[1]
Bern.	RK	/	0.191	0.307	0.059 (0.289)	0.191 (0.027)	0.307 (0.044)
	SSPRK	/	0.242	0.307	0.084 (0.289)	0.242 (0.027)	0.346 (0.035)
	DeC	/	/	/	0.412 (0.367)	0.289 (0.289)[1]	0.203 (0.289)[1]

Element &		LPS			CIP		
Time scheme		$p=1$	$p=2$	$p=3$	$p=1$	$p=2$	$p=3$
Basic	RK	0.478 (0.186)	0.13 (0.265)	0.116 (0.13)	0.464 (0.037)	0.123 (0.011)	0.165 (3.46e-03)
	SSPRK	0.605 (0.378)	0.165 (0.265)	0.335 (0.026)	0.624 (0.046)	0.143 (0.014)	0.346 (5.22e-04)
	DeC	0.412 (0.943)	0.147 (0.389)	/	0.588 (0.13)	0.143 (0.016)	/
Cub.	RK	0.971 (0.492)	0.538 (0.119)	0.425 (0.024)	0.971 (0.119)	0.538 (0.011)	0.4 (4.00e-04)
	SSPRK	1.23 (0.492)	0.681 (0.119)	0.478 (1.43e-03)	1.304 (0.119)	0.723 (7.02e-03)	0.257 (1.11e-03)
	DeC	0.971 (0.492)	0.538 (0.119)	0.425 (0.024)	0.971 (0.119)	0.538 (0.011)	0.4 (4.00e-04)
Bern.	RK	0.478 (0.186)	0.13 (0.265)	0.116 (0.13)	0.464 (0.037)	0.123 (0.011)	0.165 (3.46e-03)
	SSPRK	0.605 (0.378)	0.165 (0.265)	0.335 (0.026)	0.624 (0.046)	0.143 (0.014)	0.346 (5.22e-04)
	DeC	0.412 (0.943)	0.052 (0.215)	0.109 (0.215)	0.588 (0.13)	0.059 (0.016)	0.119 (7.02e-03)

Table 3: Optimized CFL and penalty coefficient δ in parenthesis, minimizing η_ω

DeC	SUPG	
Element	$p=2$	$p=3$
Basic	0.08 (0.025)	0.059 (0.035)
Cubature	0.346 (0.025)	0.242 (2.22 e-03)
Bernstein	0.03 (0.025)	0.1 (0.1)

Table 4: Optimized CFL and penalty coefficient δ in parenthesis, stable for all smaller CFLs

constant speed does not hold anymore. Nevertheless, the stabilization methods presented also introduces some nonlinear stabilization. To show it we will briefly consider their potential for dissipating entropy. In order to test so, we neglect the time discretization, the used elements and the quadrature and the discrete differentiation formulae.

Consider any convex smooth entropy $\rho(u)$, i.e., $\rho_{uu}(u) > 0$, the respective entropy variables $\nu := \rho_u(u)$ and the entropy flux $g(u)$ such that $\rho_u f_u = g_u$. In the following discussion, we consider the entropy variable $\nu_h = \rho_u(u)_h$ to be in the finite element space, while u_h will be defined as the projection onto the finite element space of the uniquely defined function $\nu \to u = u(\nu)$, as proposed in [2].

When substituting $v_h = \nu_h$, the Galerkin discretization of the conservation law becomes

$$\sum_K \int_K \nu_h \left(\partial_t u_h + \partial_x f(u_h)\right) dx = \sum_K \int_K \partial_t \rho_h + \partial_x g_h dx = \int_\Omega \partial_t \rho_h + [g_h]_{\partial K}\,, \tag{45}$$

which, according to the boundary conditions, gives us a measure of the variation of the entropy.

The CIP stabilization must be slightly modified for nonlinear equations with nontrivial entropies, so that it reads

$$s(v,u) := \sum_{K,\mathsf{f}\in K} \int_{\mathsf{f}} [\partial_x v^T] \rho_{uu}(u)^{-1} [\partial_x \nu(u)] d\Gamma, \tag{46}$$

where the inverse of the hessian of the entropy must be added for unit of measure reasons and it is

positive definite and invertible. So that when we substitute $v = \nu_h$ in the stabilization term, we obtain

$$s(\nu, u_h) = \sum_{K,\mathsf{f}\in K} \int_{\mathsf{f}} \underbrace{[\partial_x \nu_h^T] \rho_{uu}(u_h)^{-1} [\partial_x \nu_h]}_{>0} d\Gamma. \tag{47}$$

It would guarantee a decrease in the discrete total entropy. Moreover, this formulation coincide with (9) when we are dealing with the energy as entropy.

For the LPS we modify, similarly the formulation (13) into

$$\begin{cases} s(v,u) := \sum_K \tau_K \int_K \partial_x v^T \rho_{uu}(u)^{-1} (\partial_x \nu(u) - w) dx, \text{ with} \\ \int_K z^T (w - \partial_x \nu(u)), \quad \forall z \in V_h \end{cases} \tag{48}$$

As in the linear case, we can take $\tau_K = \tau$, and test with $v_h = \nu_h$ in the stabilization term and we substitute $z = \tau \rho_{uu}(u)^{-1,T} w$ in the previous equation and we sum this 0 contribution to the stabilization term, we obtain

$$\begin{aligned} s(\nu_h, u_h) = &\sum_K \tau \int_K \partial_x \nu_h^T \rho_{uu}(u_h)^{-1} (\partial_x \nu_h - w_h) + \rho_{uu}(u_h) w_h^T \rho_{uu}(u_h)^{-1} (w_h - \partial_x \nu_h) dx = \\ &\sum_K \tau \int_K (\partial_x \nu_h - w_h)^T \rho_{uu}(u_h)^{-1} (\partial_x \nu_h - w_h) dx \geq 0. \end{aligned} \tag{49}$$

As for the CIP we can say that the LPS stabilization reduces entropy. Anyway, this analysis does not guarantee that the fully discrete method will be entropy stable, as all the other discretizations (time, quadrature, differentiation and interpolation) are not taken into consideration.

For the SUPG stabilization, as the linear analysis of Section 2.1.1 shows, the spatial and temporal derivatives need to be properly combined. This can be done easily for space-time discretizations (see e.g. in [9]), context in which SUPG and least squares stabilization coincide. In simple cases with constant convexity entropy, namely the energy, one can bound other types of energy norm in time, but not the entropy itself. For explicit methods, and general convex entropies, the non-symmetric nature of the method requires ad-hoc analysis which we leave out of this paper. More elaborated analysis are possible with other types of stabilization, as the ones proposed in [2, 27, 24], and they will be the object of future research.

In the next sections, we perform also some nonlinear tests, where we use the coefficients we found in the stability analysis for the linear case, in order to understand if this information is also relevant for nonlinear problems.

6 Numerical Simulations

We perform numerical tests to check the validity of our theoretical findings. We will use elements of degree p, with p up to 3, with time integration schemes of the corresponding order to ensure an overall error of $\mathcal{O}(\Delta x^{p+1})$, under the CFL conditions presented earlier in Table 2. The integral formulae are performed with high order quadrature rules, for *cubature* elements they are associated with the definition points of the elements themselves, for *basic* and *Bernstein* we use Gauss–Legendre quadrature formulae with $p+1$ points per cell.

6.1 Linear advection equation

We start with the one dimensional initial value problem for the linear advection equation (24) on the domain $\Omega = [0, 2]$ using periodic boundary conditions:

$$\begin{cases} \partial_t u(x,t) + a \partial_x u(x,t) = 0 & (x,t) \in \Omega \times [0,5], \quad a \in \mathbb{R}, \\ u(x,0) = u_0(x), & \\ u(0,t) = u(2,t), & t \in [0,5], \end{cases} \tag{50}$$

where $u_0(x) = 0.1 \sin(\pi x)$. Clearly the exact solution is $u_{ex}(x,t) = u_0(x - at)$ for all $x \in \Omega$. We discretize the mesh with uniform intervals of length Δx. In particular, we will use different discretization scales to

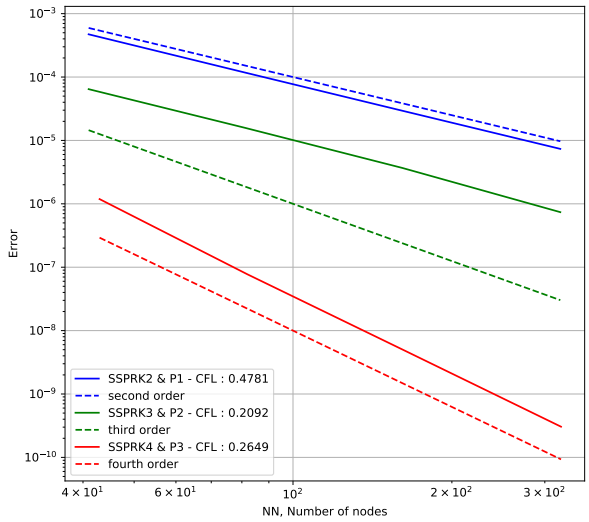

Figure 11: Error decay for linear advection with *basic* elements, LPS stabilization and SSPRK. $\mathbb{P}_1$, $\mathbb{P}_2$ and $\mathbb{P}_3$ elements are, respectively, in blue green and red.

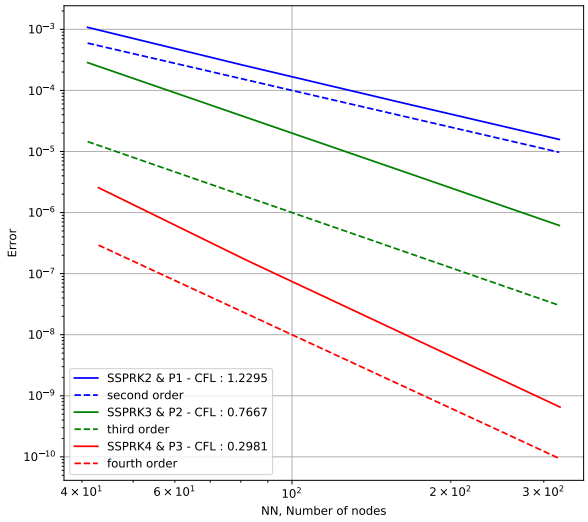

Figure 12: Error decay for linear advection with *cubature* elements, LPS stabilization and SSPRK. $\mathbb{P}_1$, $\mathbb{P}_2$ and $\mathbb{P}_3$ elements are, respectively, in blue green and red.

test the convergence: $\Delta x_1 = \{0.05, 0.025, 0.0125, 0.00625\}$ for $\mathbb{P}_1$ elements, $\Delta x_2 = 2\Delta x_1$ for $\mathbb{P}_2$ elements and $\Delta x_3 = 3\Delta x_1$ for $\mathbb{P}_3$ elements. This allows to guarantee the use ot the same number of degrees of freedom for different p. We will compare the errors obtained with SSPRK and DeC time integration method, with all the stabilization methods (SUPG, LPS and CIP) and with *basic*, *cubature* and *Bernstein* elements.

A representative result is provided as an example in Figures 11 and 12: it shows a comparison between *cubature* and *basic* elements with LPS stabilization and SSPRK time integration. As we can see, the two schemes have very similar error behavior, but the *basic* elements require stricter CFL conditions, see Table 2, and have larger computational costs because of the full mass matrix. A summary table with the order of accuracy reached by each simulations in Table 5. The plots and all the errors are available at the repository [30].

Element &		No stabilization			SUPG			LPS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Cub.	SSPRK	/	1.98	3.98	2.04	2.93	3.98	2.03	2.95	3.98	2.05	2.94	3.98
	DeC	/	1.98	3.98	2.0	2.88	3.97	2.03	2.95	3.98	2.12	2.96	3.98
Basic	SSPRK	/	3.84	3.97	2.0	2.08	3.98	2.0	2.14	3.98	2.0	2.07	3.97
	DeC	/	/	/	2.02	2.72	2.05	1.95	2.93	/	1.98	2.82	/
Bern.	SSPRK	/	3.84	3.97	2.0	2.08	3.98	2.0	2.14	3.98	2.0	2.07	3.97
	DeC	/	/	/	/	/	/	1.98	3.05	2.04	1.98	3.0	2.0

Table 5: Summary table of convergence orders, using coefficients obtained by minimizing η_u in Table 2

Looking at the table we can make the following observations. First of all, we remark that despite the weak stability obtained in the spectral analysis, in practice the absence of damping makes it difficult to obtain converging results with a fixed CFL and for all p. For this reason, in the following we will only focus on stabilized methods.

We observe otherwise that almost all the stabilized scheme provide the expected order of accuracy. When the order is correct there are minor differences in the errors. There are however few cases that

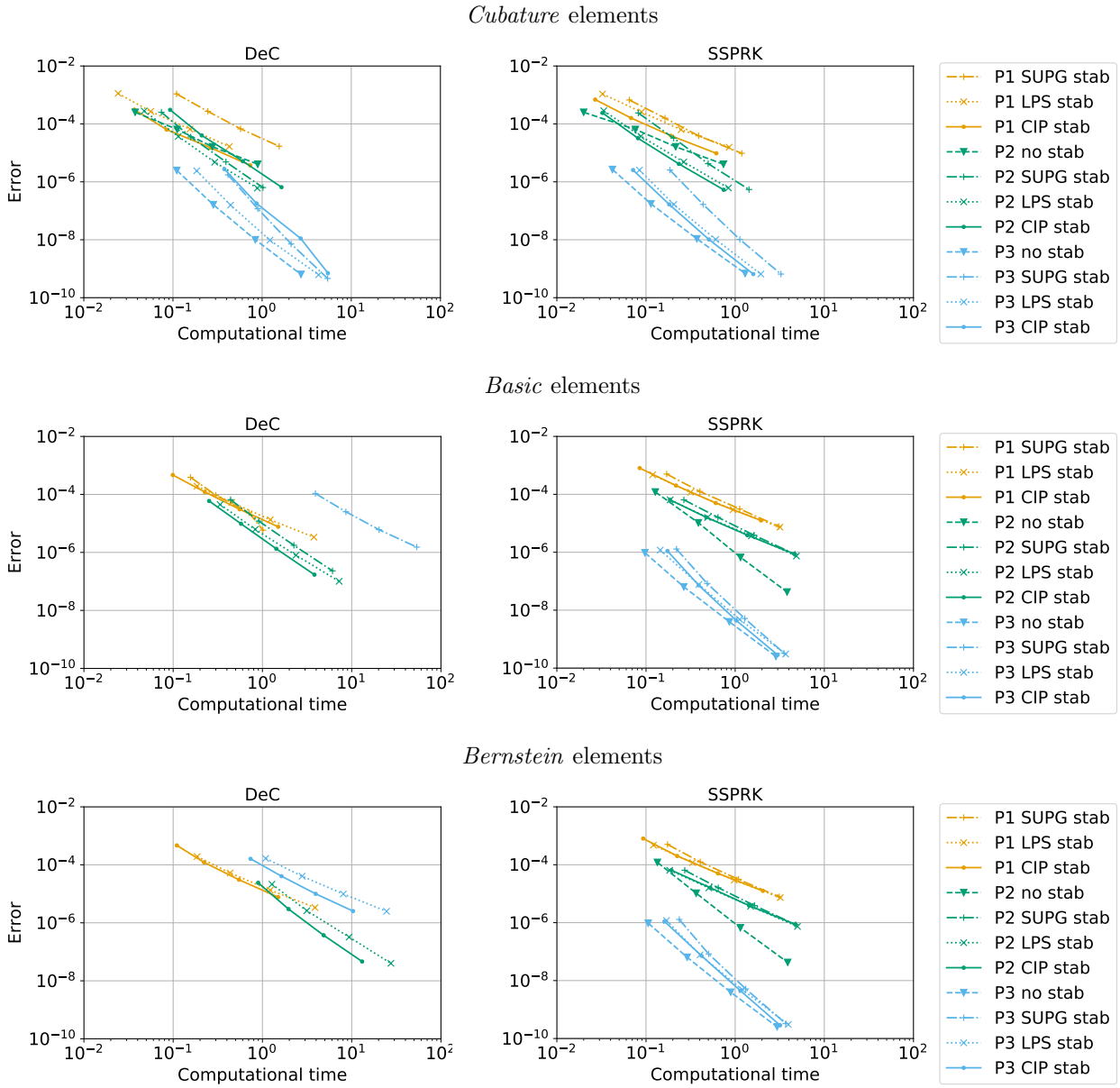

Figure 13: Error for linear advection problem (50) with respect to computational time for all elements and stabilization techniques: DeC on the left, SSPRK on the right

fail in doing so and deserve some comments. In particular, we notice the failure of DeC for *basic* $\mathbb{P}_3$ and *Bernstein* $\mathbb{B}_3$ polynomials and the SSPRK with *basic* and *Bernstein* $\mathbb{P}_2$ elements. While disappointing, this negative result is not completely new. Indeed, in [3, **?**] obtaining correct convergence with DeC for some orders required both increasing the number of substebs, thus making the method more expensive than the corresponding RK scheme, as well as including penalty terms on the jumps of higher order derivatives. Finally, note that this is in line with these methods falling in the family of "broom", "box", and thin striped shaped stability regions which we expect to be difficult to use in practice. Concerning the stabilization of high order derivatives this is also something a few authors advocate, see for instance the work by Burman, Hansbo and collaborators [17, 28]. While this mayor explains the behavior observed,

since we did not observe the need of including these terms for other cases than the DeC, we decided to focus on the simplest and most efficient approaches.

An interesting comparison is the one in Figure 13 where we plot the error of each method against computational time. Note that the simulations are all obtained using the CFL reported in Table 2. In general, we can state that the *cubature* elements obtain the best computational time as they are mass matrix free. On the other side, *Bernstein* elements are slightly more expensive than *basic* elements for DeC, because of the CFL restrictions that Table 2 requires.

Comparing time discretizations, we see that despite the inversion of the mass matrix, SSPRK converges more rapidly than DeC. We think this is related to several reasons. First of all, the DeC CFL conditions are stricter, and also DeC requires more stages. Even though not explicitly inverted, the mass matrix still needs to be assembled and multiplied to the solutions in the correction terms. Note however that the situation might radically change in the multidimensional case in which the mass matrix inversion in the SSPRK will provide a much larger overhead.

On the stabilization side, LPS and CIP behave very similarly (also their CFL do), but overall, the CIP is a little faster as it does not require the inversion of the mass matrix, for example, in DeC. As expected, the SUPG stabilization requires more computational time, even if it often has larger CFs conditions. This is even clearer when using *cubature* elements, where SUPG is the only case in which we still need to invert the mass matrix with RK time stepping.

6.2 Burgers' equation

We consider here application to a simple nonlinear problem to verify the applicability of the conditions obtained in the linear case. We test the numerical schemes on the solution of the Burgers' equation

$$\begin{cases} \partial_t u(x,t) + \partial_x \frac{u^2(x,t)}{2} = 0 & (x,t) \in \Omega \times [0, t_f], \\ u(x,0) = u_0(x), & x \in \Omega \\ u(x_D, t) = g(x_D, t), & x_D \in \partial\Omega, \end{cases} \tag{51}$$

where $\Omega = [0, 2]$ and $u_0(x) = -\tanh(4(x-1))$ and $g(x,t) = u_{ex}(x,t)$ is the boundary condition. The exact solution is obtained using the method of characteristics and reads $u_{ex}(x,t) = u_0(\chi)$ where

$$\chi = x - u_0(\chi)t \tag{52}$$

for all $(x,t) \in \Omega \times [0, t_f]$, solving the nonlinear equation (52) for χ at every point (x,t). To obtain the exact solution we employed the Broyden method implemented in SciPy library [39]. Note that the analytical solution shows a shock at time

$$t_s = -\frac{1}{\min\limits_{x\in\Omega} u_0'(x)} = \frac{1}{4}. \tag{53}$$

This knowledge allows tho set for this study $t_f = 0.5 t_s = 0.125$, at which the solution is still smooth and the convergence of the higher order approximations can be investigated. As before, in doing this we perform conformal refinement of the 1D grid, while paying attention to guarantee to use the same number of degrees of freedom for different p, and in particular taking: $\Delta x_2 = 2\Delta x_1$ for $\mathbb{P}_2$ elements and $\Delta x_3 = 3\Delta x_1$ for $\mathbb{P}_3$ elements.

Using the CFL and δ obtained in Table 2 we obtain the experimental order of convergence in Table 6.

Element &		No stabilization			LPS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Cub.	SSPRK	/	1.99	3.71	2.05	2.85	3.67	2.05	2.85	3.68
	DeC	/	1.99	3.71	2.06	2.85	3.57	2.06	2.85	3.69
Basic	SSPRK	/	1.99	3.82	2.07	2.56	3.66	2.06	2.48	3.66
	DeC	/	/	/	2.7	2.92	/	2.59	2.85	/
Bern.	SSPRK	/	1.99	3.82	2.07	2.56	3.66	2.06	2.48	3.66
	DeC	/	/	/	2.7	2.9	1.41	2.59	2.87	1.37

Table 6: Summary table of convergence order, using coefficients obtained in Table 2

The results are very similar to the ones obtained for the linear advection case. There is a small improvement in *basic* and *Bernstein* $\mathbb{P}_2$ SSPRK cases, while the DeC *basic* and *Bernstein* $\mathbb{P}_3$ cases are even worse than the linear advection ones. The DeC $\mathbb{P}_1$ *basic* and *Bernstein* cases show a super–convergent behavior. The interested reader will find the convergence plots for all the combinations on the repository [30]. Here we focus on the comparison between error and computational time, reported in Figure 14. Again for *cubature* elements it is clear the advantage in using high order methods, in

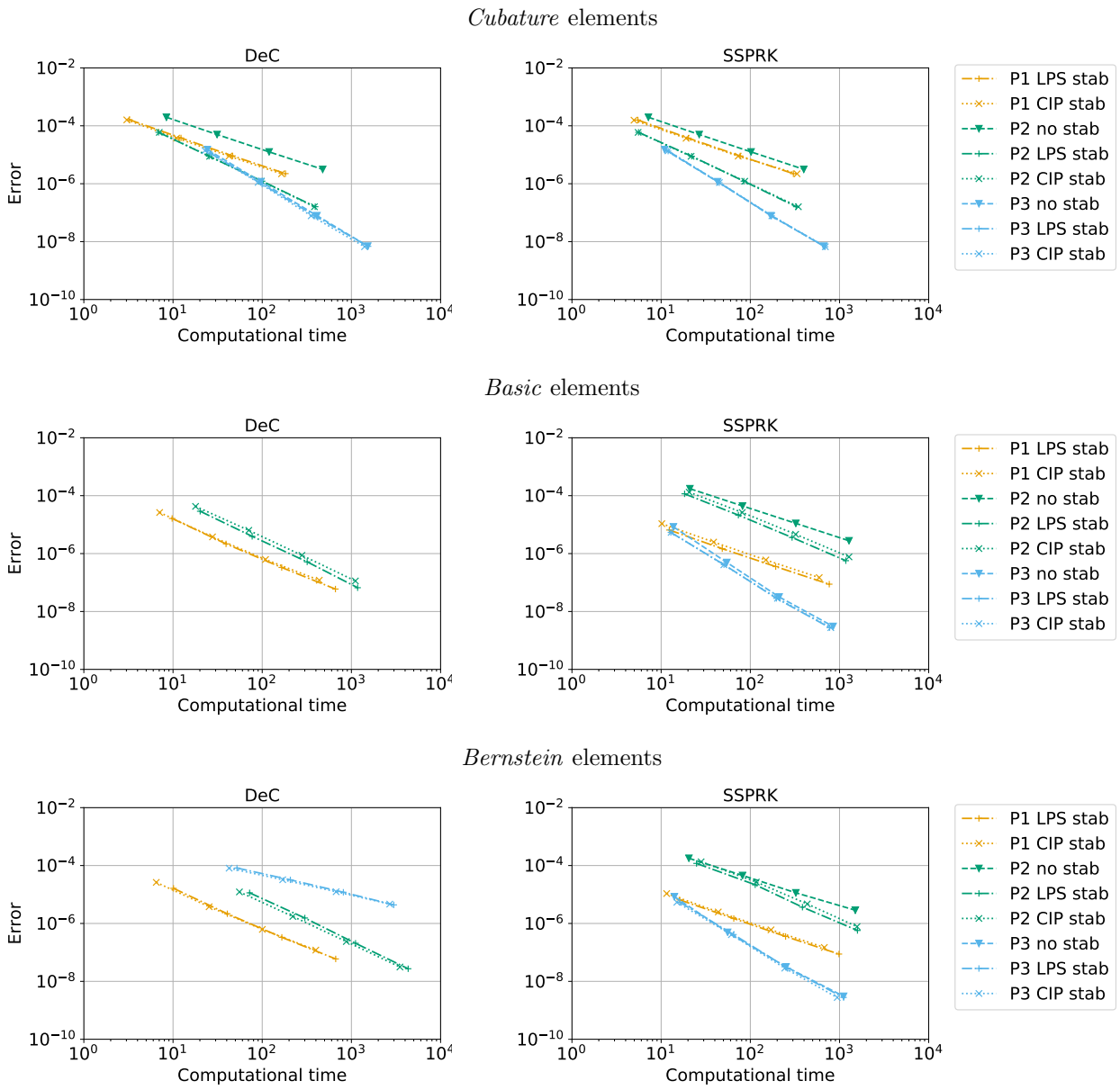

Figure 14: Error for Burgers' equation (51) with respect to computational time for all elements and stabilization techniques: DeC on the left, SSPRK on the right

particular for SSPRK methods, which has less stages than DeC. For this test, we only compare CIP and LPS and they systematically out-perform SUPG. For these two, the difference in computational time is very minimal for all element choices. This may change in the multidimensional case where the LPS may

be penalized on elements requiring the inversion of the full mass matrix.

For DeC *basic* and *Bernstein* $\mathbb{P}_1$ elements, the superconvergence of the second order schemes makes them the best in their category, see Table 6. For SSPRK the expected order of convergence of fourth order scheme shows how the high order accurate methods can provide the fastest and most precise solutions.

6.3 Shallow water equations

As a final application we consider the non linear shallow water equations:

$$\begin{cases} \partial_t h + \partial_x(hu) & = 0, \\ \partial_t(hu) + \partial_x(hu^2 + g\frac{h^2}{2}) + \Phi & = 0, \end{cases} \quad x \in \Omega,\ t \in [0,5]. \tag{54}$$

Here, h is the water elevation, u the velocity field, g the gravitational acceleration. We will solve the system on the domain $\Omega = [0, 200]$, and add the source term $\Phi = \Phi(x,t)$ in order to impose the solution to be equal to

$$\begin{cases} h_{ex}(x,t) = h_0 + \epsilon h_0 \text{sech}^2(\kappa(x - ct)), \\ u_{ex}(x,t) = c\left(1 - \frac{h_0}{h_{ex}(x,t)}\right), \\ \kappa = \sqrt{\frac{3\epsilon}{4h_0^2(1+\epsilon)}}, \quad c = \sqrt{gh_0(1+\epsilon)}. \end{cases} \tag{55}$$

Following the classical manufactured solution method, we set

$$\begin{aligned} \Phi(x,t) &= -\left[\partial_t\left(h_{ex}(x,t)u_{ex}(x,t)\right) + \partial_x\left(h_{ex}(x,t)u_{ex}^2(x,t) + g\frac{h_{ex}^2(x,t)}{2}\right)\right] \\ &= -\left[h_{ex}(\partial_t u_{ex} + u_{ex}\partial_x u_{ex} + g\partial_x h_{ex})\right]. \end{aligned}$$

For our study, we set $\epsilon = 1.2$, $h_0 = 1$ and the initial and Dirichlet boundary condition given by the exact solution at time 0 and at the borders of the domain.

We discretize the mesh with uniform intervals of length Δx, 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. In Table 7 we show the convergence orders for this shallow water problem with the CFL and δ coefficients found in Table 2.

Element &		No stabilization			LPS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Cub.	SSPRK	/	1.96	5.17	2.26	2.69	5.02	2.39	2.68	5.05
	DeC	/	1.97	5.17	2.28	2.65	4.79	2.7	2.66	5.07
Basic	SSPRK	/	1.98	5.54	1.94	2.31	4.93	1.95	2.29	4.98
	DeC	/	/	/	2.23	2.74	/	2.01	2.58	/
Bern.	SSPRK	/	1.97	2.44	1.94	2.07	2.19	1.95	2.09	2.21
	DeC	/	/	/	2.23	2.0	2.0	2.01	2.0	1.98

Table 7: Summary tab of convergence order, using coefficients obtained by minimizing η_u

The results obtained are similar to those of the other cases. The convergence rates are at least the expected ones with *cubature* elements while we still see problems with DeC and *basic* elements in the fourth order case, as well as with *Bernstein* polynomials for both $\mathbb{P}_2$ and $\mathbb{P}_3$. On the other hand, some superconvergence is measured in the $\mathbb{P}_3$ case with both *cubature* and *basic* elements. This creates an even larger bias in the error-cpu time plots, Figure 15, in favor of these higher polynomial degrees.

7 Conclusion

In summary, we propose a comparison of high order continuous Galerkin methods with stabilization techniques for hyperbolic problems. On the linear advection equation, we perform a Fourier analysis on the spatial discretization, then a von Neumann analysis on the space–time discretization given by each combination of stabilization, time discretization and finite elements. This provides reliable parameters

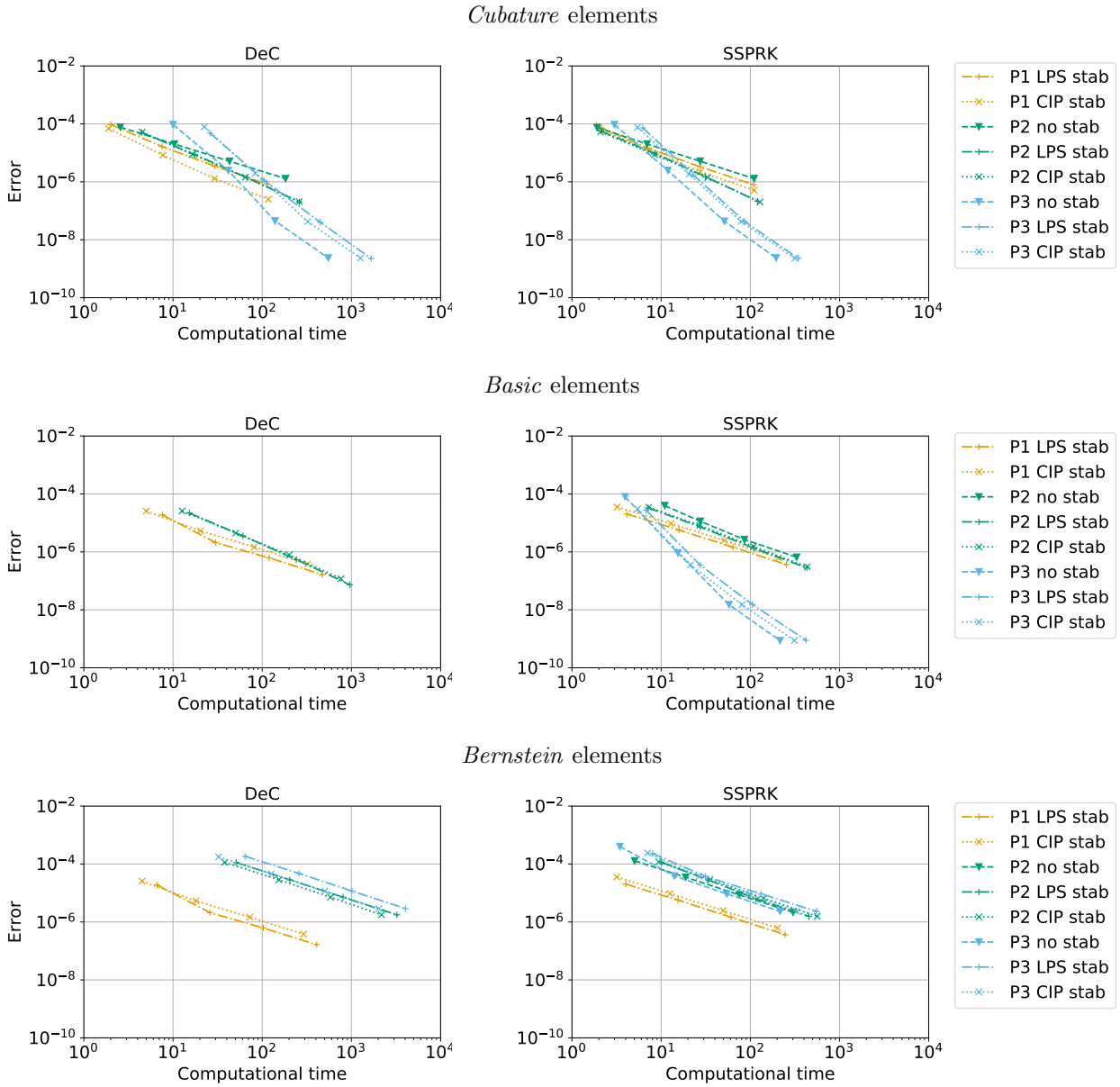

Figure 15: Error for Shallow Water equations (54) with respect to computational time for all elements and stabilization techniques: DeC on the left, SSPRK on the right

and CFL conditions for all the mentioned methods that can be used both in the linear advection case and in nonlinear problems, as the Burgers' and shallow water simulations showed.

The Fourier analysis is limited to one dimensional problems (or structured multidimensional meshes), so the main ongoing development is the verification of the properties of the methods studied in a multidimensional setting based on the approximation choices suggested e.g. in [38, 20, 23] and references therein.

Acknowledgment

This work was performed within the Ph.D. project of Sixtine Michel: "Evaluation of coastal and urban submersion risks", supported by INRIA and the BRGM, co-funded by in INRIA–Bordeaux Sud–Ouest and the Conseil Régional de la Nouvelle Aquitaine. Mario Ricchiuto and Davide Torlo have been supported by team CARDAMOM in INRIA–Bordeaux Sud–Ouest. Davide Torlo and Rémi Abgrall have been supported by the Swiss National Foundation grant No 200020_175784.

A Time schemes

In this appendix we introduce the time integration coefficients used in this work, to make the study fully reproducible. In Table 8 there are the RK coefficients, in Table 9 the SSPRK coefficients and in Table 10 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 8: Butcher Tableau of RK methods

SSPRK(3,2) by [37]

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 [35, 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 [35, 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 9: 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 10: DeC coefficients for equispaced subtimesteps.

B Fourier analysis, spatial and temporal eigenanalysis

In this appendix we present a summary of the fully discrete Fourier analysis of Section 3.2, comparing different time schemes (SSPRK and DeC), discretizations (*basic*, *cubature*, *Bernstein*), and stabilization methods (LPS, CIP, SUPG). We show the phase ω and the damping ϵ coefficients using the *best parameters* obtained by minimizing the relative error of the solution η_u for each scheme in Table 2. When the scheme was unstable we did not plot the mode. In Figure 16 one finds the phase and the damping for *basic* elements, in Figure 17 for *cubature* elements and in Figure 18 for *Bernstein* elements. We remark that for *cubature* elements in Figure 17, Δx_3 is scaled differently with respect to the other orders because the point distribution is not equispaced.

In general, we can observe that the phase error increases passing from full matrix SSPRK methods to diagonal one DeC. This is noticeable even more for *Bernstein* elements. *Cubature* elements, which are not effected by the mass lumping, do not show this behavior, and have a dispersion error which is greater than the other lumped methods, but smaller than the other full mass matrix methods. This step is also associated to a greater damping in the higher frequencies.

References

[1] R. Abgrall, *High order schemes for hyperbolic problems using globally continos approximation and avoiding mass matrices*, Journal of Scientific Computing, 73 (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 (2018), pp. 640 – 666.

[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 (2018), pp. 274–297.

[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 (2020), pp. 1573–7691.

[5] ——, *Analysis of the SBP-SAT Stabilization for Finite Element Methods Part II: Entropy Stability*, Commun. Appl. Math. Comput., (2021), pp. 2661–8893.

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

[7] R. Abgrall and D. Torlo, *High order asymptotic preserving deferred correction implicit-explicit schemes for kinetic models*, 2020.

[8] N. Ahmed, G. Matthies, L. Tobiska, and H. Xie, *Discontinuous Galerkin time stepping with local projection stabilization for transient convection–diffusion-reaction problems*, Computer Methods in Applied Mechanics and Engineering, 200 (2011), pp. 1747–1756.

[9] T. Barth, *Numerical methods for gasdynamic systems on unstructured meshes*, in An Introduction to Recent Developments in Theory and Numerics for Conservation Laws, Kröner, Ohlberger, and Rohde, eds., vol. 5 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, Heidelberg, 1998, pp. 195–285.

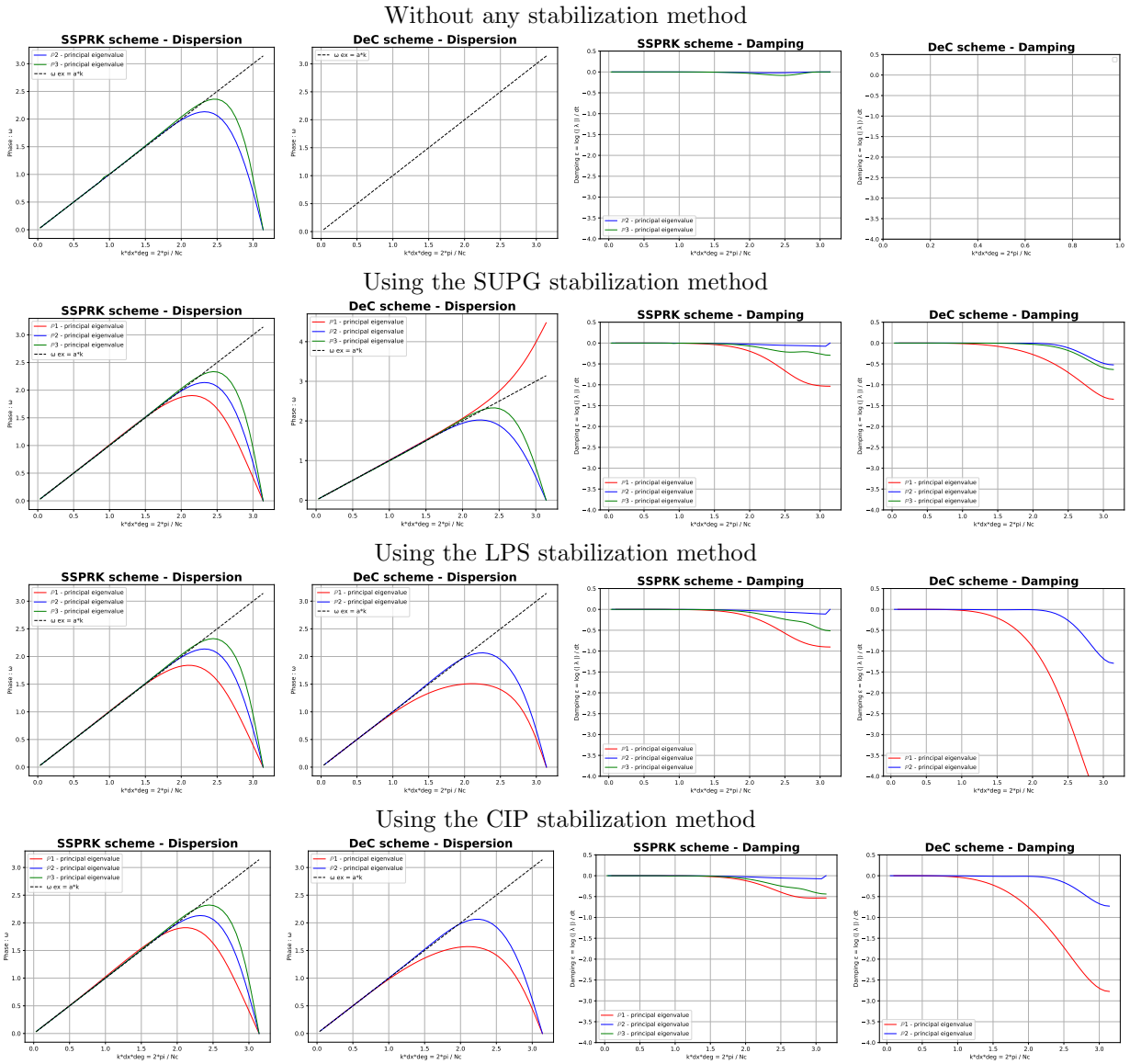

Figure 16: Dispersion and damping coefficients for *basic* elements, with DeC and SSPRK methods and all stabilization techniques

[10] R. Becker and M. Braack, *A finite element pressure gradient stabilization for the stokes equations based on local projections*, Calcolo, 38 (2001), pp. 173–199.

[11] ———, *A two-level stabilization scheme for the navier-stokes equations*, in Numerical mathematics and advanced applications, Springer, 2004, pp. 123–130.

[12] M. Braack and E. Burman, *Local projection stabilization for the oseen problem and its interpretation as a variational multiscale method*, SIAM Journal on Numerical Analysis, 43 (2006).

[13] E. Burman, *Consistent supg-method for transient transport problems: Stability and convergence*, Computer Methods in Applied Mechanics and Engineering - COMPUT METHOD APPL MECH ENG, 199 (2010), pp. 1114–1123.

[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

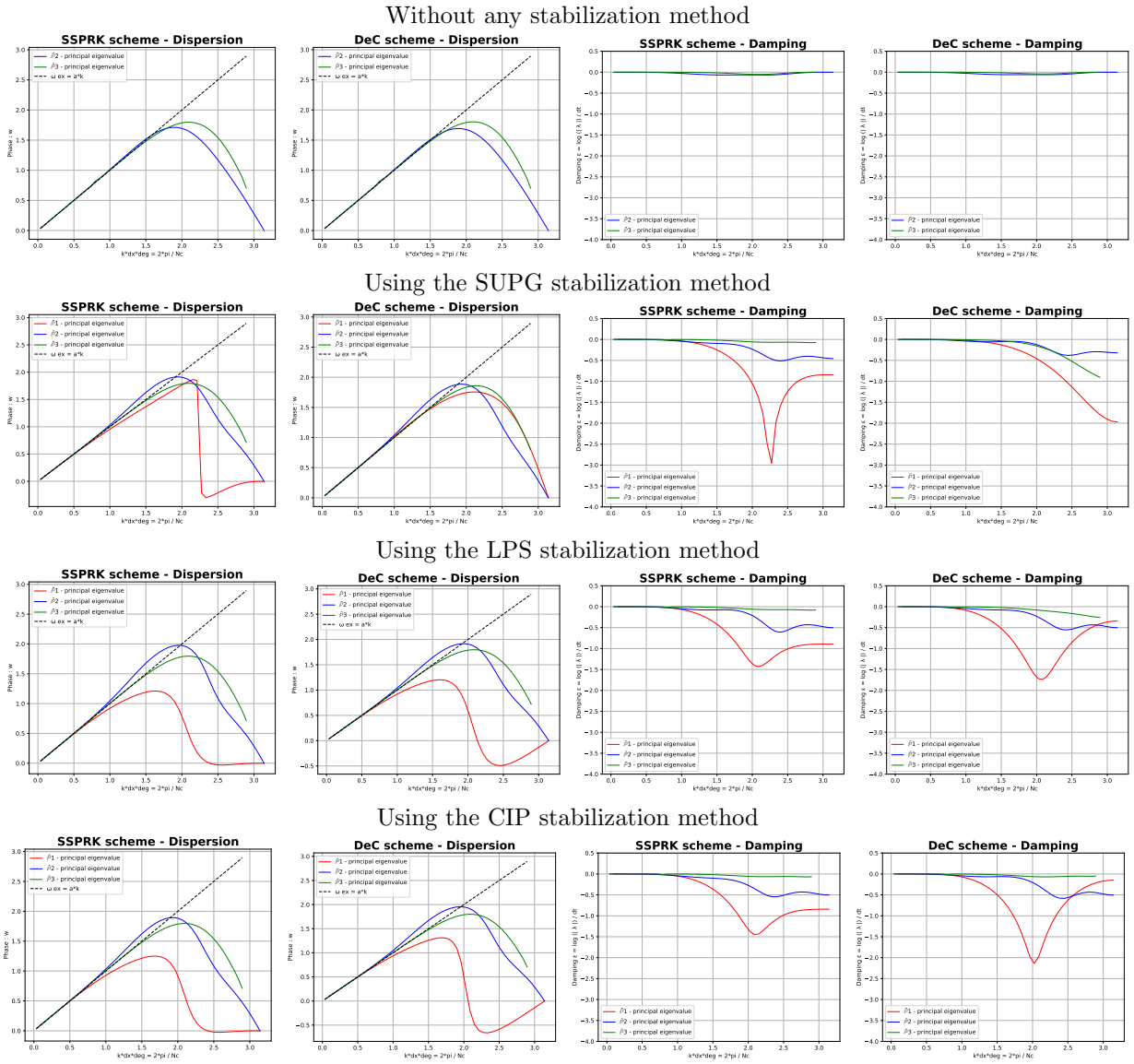

Figure 17: Dispersion and damping coefficients for *cubature* elements, with DeC and SSPRK methods and all stabilization techniques

Analysis, 48 (2010).

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

[16] ———, *The edge stabilization method for finite elements in cfd*, in Numerical mathematics and advanced applications, Springer, 2004, pp. 196–203.

[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 (2020), pp. 783–818.

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

[19] ———, *Interior penalty continuous and discontinuous finite element approximations of hyperbolic equations*, Journal of Scientific Computing, 43 (2010), pp. 293–312.

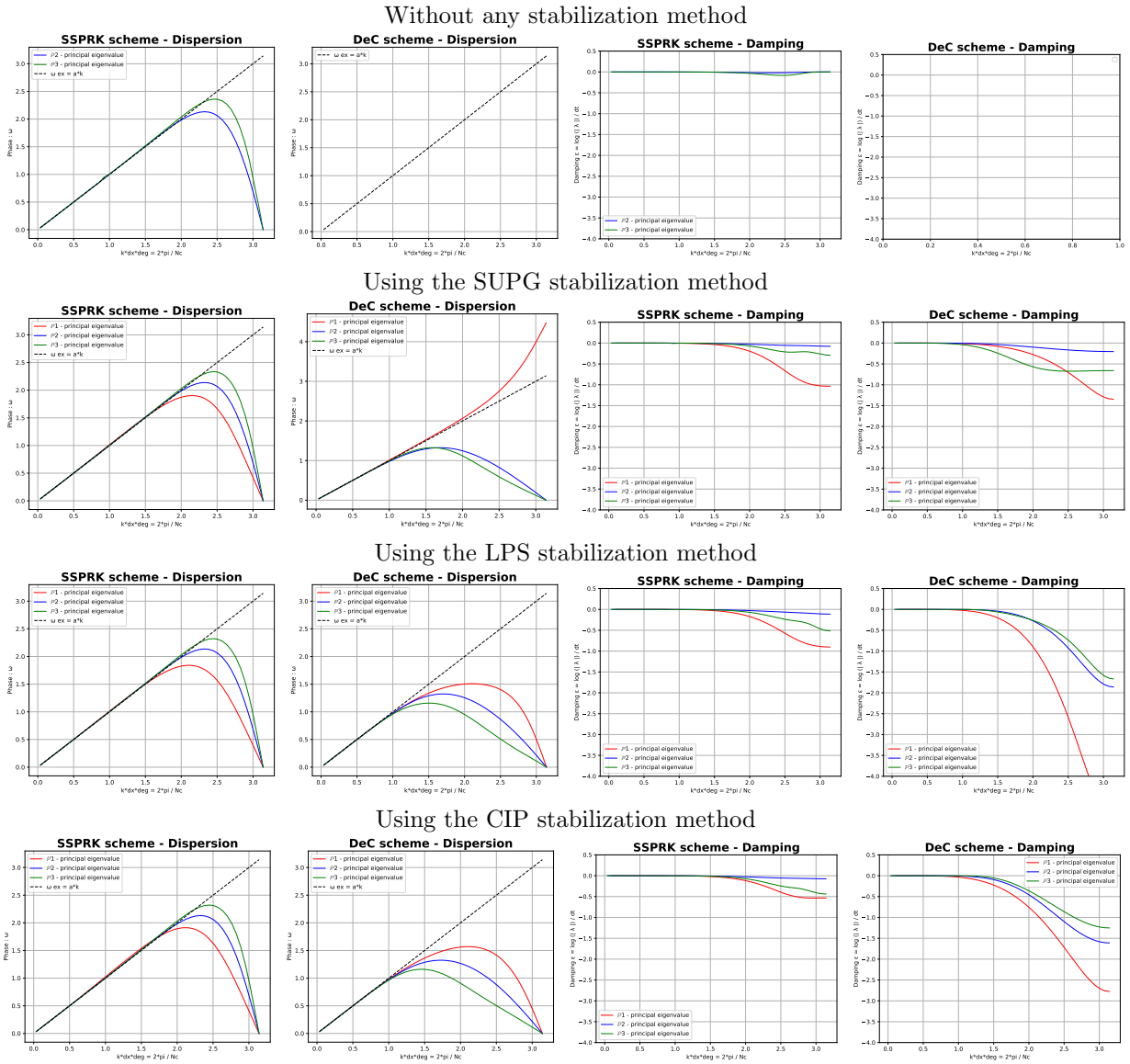

Figure 18: Dispersion and damping coefficients for *Bernstein* elements, with DeC and SSPRK methods and all stabilization techniques

[20] 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 (2001).

[21] J. Douglas and T. Dupont, *Interior Penalty Procedures for Elliptic and Parabolic Galerkin Method*, vol. 58, Springer, 08 2008, pp. 207–216.

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

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

[24] J.-L. Guermond, R. Pasquetti, and B. Popov, *Entropy viscosity method for nonlinear conservation laws*, Journal of Computational Physics, 230 (2011), pp. 4248 – 4267. Special issue High Order Methods for CFD Problems.

[25] 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 (1982), pp. 199–259.

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

[27] D. Kuzmin and M. Quezada de Luna, *Algebraic entropy fixes and convex limiting for continuous finite element discretizations of scalar hyperbolic conservation laws*, Computer Methods in Applied Mechanics and Engineering, 372 (2020), p. 113370.

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

[29] 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 (2017), pp. 458 – 480.

[30] S. Michel, D. Torlo, M. Ricchiuto, and R. Abgrall, *Stability analysis of several FEM methods: results and code.* `https://gitlab.inria.fr/dtorlo1/stability-analysis-of-several-fem-methods-results-and-code.git`, May 2021.

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

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

[33] 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 (2020), p. 109112.

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

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

[36] S. Sherwin, *Dispersion analysis of the continuous and discontinuous galerkin formulations*, Discontinuous Galerkin Methods, 11 (1999).

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

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

[39] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, Nature Methods, 17 (2020), pp. 261–272.