Finite Element Methods for Shallow Water Equations : Analysis, Modeling and Applications to Coastal Hydrodynamic

Sixtine Michel

▶ **To cite this version:**

Sixtine Michel. Finite Element Methods for Shallow Water Equations : Analysis, Modeling and Applications to Coastal Hydrodynamic. Modeling and Simulation. Université de Bordeaux, 2022. English. NNT : 2022BORD0050 . tel-03656234

HAL Id: tel-03656234

https://theses.hal.science/tel-03656234

Submitted on 2 May 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUE ET D'INFORMATIQUE

MATHÉMATIQUES APPLIQUÉES ET APPLICATION DES MATHÉMATIQUES

Par **Sixtine MICHEL**

FINITE ELEMENT METHODS FOR SHALLOW WATER EQUATIONS: ANALYSIS, MODELING AND APPLICATIONS TO COASTAL HYDRODYNAMIC

Sous la direction de : **Mario RICCHIUTO**

Soutenue le 4 Mars 2022

Membres du jury :

M. Rémi ABGRALL	Professeur	Universität Zürich	Président
Mme. Stéphanie SALMON	Professeur	Université de Reims	Rapporteur
M. Spencer J. SHERWIN	Professeur	Imperial College London	Rapporteur
M. Raphaël LOUBÈRE	Directeur de Recherche	CNRS	Examinateur
Mme. Lisl WEYNANS	Maitre de Conférence	Université de Bordeaux	Examinatrice
M. Vincent PERRIER	Chargé de Recherche	INRIA	Examinateur
M. Rodrigo PEDRERO	Docteur	BRGM	Invité
M. Mario RICCHIUTO	Directeur de Recherche	INRIA	Directeur de thèse

Finite Element Methods for Shallow Water Equations: Analysis, Modeling and Applications to Coastal Hydrodynamic

Abstract: This phd will be carried out as part of the Inria CARDAMOM team's activities concerning adaptive methods for coastal flows. The work will benefit from the team's interactions with the BRGM in terms of real applications, as well as exchanges with the University of Zurich on certain methodological aspects.
The main objective is to develop and compare high order and adaptative methods for the simulation of Shallow Water flows. More precisely, the objective is to obtain continuous finite element methods without mass matrix, stable and explicit in time with performances comparable to Galerkin discontinuous methods.
The implementation will be done in an object-oriented finite element library used by the INRIA CARDAMOM and CAGIRE teams, as well as at BRGM. The final validation will be done on real cases of interest for the *Région Nouvelle-Aquitaine* and in close collaboration with the BRGM Orléans with exchanges with the *Rivages Pro-Tech center* of SUEZ. Further exchanges are envisaged on certain aspects of the work, in particular with the University of Zurich, concerning residual distribution methods and the analysis of stabilized continuous methods. The main scientific contributions of this work will be:
- Multidimensional spectral analysis of continuous stabilized finite element methods.
- Comparison and optimization of these at different orders.
- The implementation of high order and well-balanced approaches near flood fronts.
- Numerical validation of methods on university cases as well as real cases (example of a case in *Nouvelle-Aquitaine*).
Keywords: Continuous Galerkin methods, Dispersion analysis, stabilization techniques, high order accuracy, nonstandard elements, mass lumping, Shallow Water equations.

Méthodes éléments finis pour la simulation d'écoulements en eaux peu profondes: Analyse, modélisation et applications à l'hydrodynamique côtière

Résumé : Cette thèse se fera dans le cadre des activités de l'équipe Inria CARDAMOM en matière de méthodes adaptatives pour les écoulements côtiers. Le travail bénéficiera d'interactions avec le BRGM en matière d'applications réelles, ainsi que d'échanges avec l'Université de Zurich sur certains aspects méthodologiques.
L'objectif principal de la thèse est de développer et comparer des approches d'ordres élevés pour la simulation d'écoulements en eaux peu profondes. Plus précisément, l'objectif est d'obtenir des méthodes d'éléments finis continues sans matrice de masse, stables et explicites en temps avec des performances comparables à des schémas de type Galerkin discontinus.
La mise en œuvre se fera dans une bibliothèque éléments finis orientée objets utilisée dans les équipes INRIA CARDAMOM et CAGIRE, ainsi qu'au BRGM. La validation finale se fera sur des cas réels d'intérêt pour la Région Nouvelle Aquitaine et en collaboration étroite avec le BRGM Orléans avec des échanges avec le centre Rivages Pro-Tech de SUEZ. D'autres échanges sont envisagés sur certains aspects du travail, en particulier avec l'Université de Zurich, concernant les méthodes aux résidus et l'analyse des méthodes continues dites stabilisées. Les principales contributions scientifiques de ce travail seront :
- L'analyse spectrale multidimensionnelle des méthodes numériques éléments finis continues stabilisées.
- Comparaison et optimisation de celles-ci à différents ordres.
- La mise en œuvre d'approches d'ordres élevés et well-balanced en proximité de fronts d'inondation.
- La validation numérique des méthodes sur des cas universitaires ainsi que des cas réels (exemple d'un cas en Nouvelle-Aquitaine).
Mots clés: Méthodes Galerkin Continues, Analyse de dispersion, techniques de stabilisation, méthodes d'ordre élevé, Élements non standards, *mass lumping*, Équations de Saint-Venant.

Team CARDAMOM (INRIA Bordeaux Sud-Ouest)
UMR 5152, Université de Bordeaux, 351 Cours de la Libération, 33400 Talence, France.

Acknowledgements

Firstly, of course, I would like to thank my supervisor Mario Ricchiuto. Thanks for these three amazing years. I always tried to follow your precious advice even if you often let me free to explore my ideas. You accepted to follow and support all of my initiatives and you always knew how to keep me on the good direction. These three years were a very rewarding scientific experience.

Then, a very special thank to Davide Torlo, my "unofficial" supervisor who also guided me during my research, conferences, etc. Thank you for your time and your generosity! We did a very huge work together, it was and it is still a real pleasure to work with you.

I would also like to thank Luca Arpaia, Andrea Fillipini, Vincent Perrier, Benjamin Lux, Christopher Poette and Rodrigo Pedreros for their precious help and contributions in the Aerosol / Uhaina project. Thanks also for your friendship which make the work easier to do.

Now, I would gratefully acknowledge all other people which contributed to make the PhD a wonderful story. A huge thank goes to all Cardamom mates, with whom I shared my life during these 3 years. You make the open-space a wonderful place to work. You are all amazing and I already miss "les vendredi gourmands" with the tiramisù competition. You are too many so I can not cite all of you, but I can at least cite my faithful PhD mates Elie, Giulia and Mirco with whom we share our stress, our happiness, our fears, all of our emotions.

Also, I want to thank other colleagues which participate to make these three years much more pleasant, in particular Corinne who learnt me the sewing and woke up my artistic side, and obviously Anne-Laure for her precious help.

I would also like to thank my love, my family and my closest friends who shared my life during all of my studies. In particular Juliette and Nathalie, my precious master mates.

Résumé en francais

Contexte et motivations

Du aux effets du changement climatique, les centres habités proches de la mer ou de cours d'eau sont exposés à des risques de submersion. Pour atténuer les conséquences de ces événements et mieux se préparer à réagir, il est nécessaire de mener des études d'évaluation des risques avec certaines exigences en terme de précision et de résolution. Il faut notamment prendre en compte les phénomènes pertinents tels que l'effet des vagues, les incertitudes qui les affectent, et la possibilité de décrire de manière flexible des phénomènes à différentes résolutions spatiales pour des petites et grandes échelles. Pour cela, la disponibilité de codes très performants et précis est essentielle. Ce projet fait suite aux recherches menées par l'équipe CARDAMOM du centre INRIA Bordeaux Sud-Ouest. En particulier, la thèse contribuera à terme au projet UHAINA[1], une plateforme opérationnelle open-source de simulation de l'impact des vagues sur la côte. Cette plateforme fonctionne avec la collaboration de cinq instituts, dont INRIA et le BRGM d'Orléans également impliqués dans le projet de co-financement soumis à la Région Nouvelle Aquitaine.

Dans ce contexte, cette thèse s'intéresse à des problématiques de recherche en amont: l'étude d'approches améliorées de l'hydrodynamique côtière. L'objectif de la thèse sera d'obtenir un modèle efficace/optimal du point de vue de la précision (pour un nombre donné d'inconnues de calcul), de la robustesse et de la rapidité de calcul. En effet, aujourd'hui nous disposons de données et de cartes haute résolution des grandes agglomérations urbaines. Si ces données permettent des prévisions précises (à l'échelle urbaine), les résolutions qui en résultent conduisent à des temps de calcul importants. Ce qui représente un frein lorsqu'on envisage leur application dans un contexte opérationnel. Il semble donc essentiel de rendre disponible des codes d'ordre élevé, adaptatifs, géométriquement flexibles et massivement parallèles pour ce type d'applications.

Pour modéliser la submersion et l'inondation, nous pouvons utiliser les équations dites "Shallow Water": un ensemble de lois de conservation, avec un caractère hyperbolique sous-jacent. Ces équations permettent d'approximer avec une précision surprenante le déferlement et la montée des vagues. De plus, elles ont un certain potentiel pour modéliser la propagation des vagues, en particulier pour les vagues longues (comme par exemple les vagues de tempête ou de tsunami). Ce modèle trouve également des applications en hydrologie et en météorologie (voir [31, 119, 120] et ses références).

Une caractéristique fondamentale des lois de conservation non-linéaires est que des discontinuités peuvent apparaître pendant la simulation, même à partir de données initiales lisses. Le principal enjeu numérique est de traiter ces discontinuités à la fois mathématiquement et informatiquement. Historiquement, les premières méthodes utilisées pour discrétiser un système hyperbolique étaient les méthodes de différences finies et de volumes finis [78], car elles permettent de décrire de manière précise les chocs. Cependant, même si elles ont

[1] https://gitlab.inria.fr/uhaina1/uhaina/-/wikis/home

été les premières introduites dans la littérature de par leur caractère intuitif, l'extension à des ordres de précisions élevés conduit à des coûts de calculs élevés, relatifs aux stockages de données (de structure) à considérer. Les méthodes de Galerkin ont alors été envisagées afin d'atteindre une convergence spatiale d'ordre élevé [61, 35] à des coûts de calcul moindre. Il est bien connu que la méthode standard des éléments finis de Galerkin n'est en général pas bien adaptée à la résolution des problèmes d'advection et d'advection-diffusion. Il est alors apparu deux possibilités pour améliorer la stabilité tout en gardant la précision. La première est la formulation Galerkin discontinue (DG) [36], qui considère des solutions approchées discontinues. La seconde utilise la formulation Galerkin continue (CG), en ajoutant un terme de stabilisation conçu de manière appropriée.

Méthodes numériques

Dans cette dernière approche (Galerkin continue stabilisée), l'une des techniques de stabilisation les plus populaires est le *Streamline-Upwind Petrov-Galerkin* (SUPG), introduit dans [65] (voir aussi [67, 21]). La formulation est fortement consistante dans le sens où l'on stabilise le résidu complet, et la stabilisation disparaît lorsqu'elle est appliquée à des solutions exactes de l'équation différentielle. Dans ce travail, nous considérerons également des techniques de stabilisation symétriques qui sont plus simples à mettre en œuvre par rapport aux méthodes de stabilisation basées sur les résidus. La première alternative est la stabilisation dite *Continuous Interior Penalty* (CIP) utilisée dans [25, 27, 23]. Cette méthode a été développée par E. Burman et P. Hansbo dans [24], mais elle peut aussi être vue comme une variante de la méthode initialement proposée par Douglas et Dupont [46]. La méthode stabilise la formulation de Galerkin en ajoutant un terme proportionnel au saut du gradient de la solution à travers les interfaces du maillage. Ce terme est calculé sur la solution au pas de temps précédent, il n'affecte donc pas la structure de la matrice de masse. La deuxième alternative est l'approche *Orthogonal Subscale Stabilization* (OSS). Introduite à l'origine sous le nom de *Pressure Gradient Projection* (PGP) dans [38] pour les équations de Stokes, elle a été étendu à la méthode OSS dans [37, 8] pour différents problèmes d'instabilités numériques, tels que les problèmes de convection–diffusion–réaction. La méthode stabilise la formulation de Galerkin en pénalisant les fluctuations de gradient.

Dans la résolution numérique d'équations aux dérivées partielles (EDP), l'intégration en temps joue également un rôle majeur. Les méthodes utilisées pour atteindre une convergence temporelle d'ordre élevé dans ce travail sont les méthodes générales *Runge-Kutta* (RK), ainsi que *Strong Stability Preserving Runge-Kutta* introduites dans [117] (SSPRK). De nombreuses variantes de SSPRK existent [117, 118, 112, 56, 30], et nous allons comparer certaines d'entre elles numériquement afin de sélectionner la plus stable. Ces méthodes explicites seront également comparées aux méthodes dites *Deferred Corrections* (DeC), qui ont été introduites à l'origine dans [47] en tant que solveurs explicites d'équations différentielles ordinaires (EDO), et par la suite en tant que solveurs implicites dans [90]. Puis sont apparues des versions et extensions préservant la positivité des solveurs d'EDP [100, 2]. Dans [2, 104, 107] la méthode est également étendue afin d'éviter l'inversion de la matrice de masse. L'idée est d'appliquer une condensation de la matrice de masse (*mass lumping*) et d'effectuer un processus itératif de correction pour atteindre l'ordre de convergence recherché. Ceci n'est réalisable que lorsque la matrice (locale et globale) contient uniquement des valeurs positives sur sa diagonale. L'utilisation des polynômes de Bernstein fut alors recommandée dans [2], mais d'autres choix sont également possibles comme par exemple

l'utilisation d'éléments dit de *Cubature* sous certaines conditions.

Ceci permet d'introduire les deux derniers aspects fondamentaux des méthodes éléments finis (MEF) : l'approximation polynomiale et la discrétisation spatiale. En pratique, et c'est ce qui sera utilisé dans cette thèse, des éléments triangulaires sont utilisés pour manipuler des géométries complexes (en 2D). L'approche classique consiste à définir les degrés de liberté placés uniformément dans le triangle et à définir les fonctions de base correspondantes (qui sont en général des fonctions de base de *Lagrange*). Suivant la procédure de discrétisation pour la MEF, la résolution d'un système hyperbolique consiste à résoudre un système en inversant une matrice appelée *la matrice de masse*. Le temps CPU de cette opération ne peut pas être négligé. Une façon de réduire ce coût est de recourir au principe du *mass-lumping*, éventuellement sans affecter la précision. Parmi les solutions possibles, on peut citer les éléments de *Cubature*, introduits par G. Cohen et P. Joly (2001) dans [39] qui sont une extension des polynômes de *Lagrange* dans le but d'optimiser l'erreur sous-jacente des formules de quadrature (tous les détails de ces éléments peuvent être trouvés dans [39, 55, 69]). Des techniques similaires ont été utilisées pour minimiser l'erreur d'interpolation en utilisant les points de *Fekete* et *Gauss–Lobatto* [70, 115, 122]. Cependant, les points de quadrature de *Gauss – Lobatto* ne sont connus que pour les domaines de type ligne en 1D et quadrangulaire en 2D, ce qui ne permet pas d'étendre leurs défintinitions à des domaines de type "produits non tensoriels" comme le triangle. Cependant, de part leurs définitions, ces derniers éléments ne permettent pas d'appliquer le principe du *mass-lumping*: ils ne garantissent ni les coefficients strictement positifs dans la diagonale, ni l'ordre de convergence. À noter que les points de *Fekete* (étant connus pour être les points *Gauss–Lobatto* en 1D [50] et dans le cube de dimension $d \geq 2$ [14])) sont une généralisation possible des points de *Gauss–Lobatto* pour le triangle. L'objectif de ces polynômes est d'utiliser les points de l'interpolation lagrangienne des polynômes comme points de quadrature. Un autre type de fonction de base sera également utilisé dans ce travail : les fonctions polynomiales de *Bernstein*. Ils vérifient des propriétés supplémentaires en plus de celle des points de Lagrange. En particulier, ces propriétés conduisent également au fait que la valeur en chaque point est une combinaison convexe des coefficients des polynômes. Par conséquent, il est facile de borner la valeur minimale et maximale de la fonction par le minimum et le maximum des coefficients. Cela a été utilisé dans différentes techniques pour préserver la positivité de la solution [7, 75].

Travail effectué

Dans cette thèse, nous décrirons, analyserons et optimiserons ces différents aspects de la résolution numérique, via la résolution de problèmes hyperboliques. En particulier, dans les chapitres 4 et 5, l'analyse sera effectuée selon la méthode de Fourier appelée aussi l'*analyse spectrale*. Plusieurs travaux montrent la méthode appliquée à un cas particulier (DG, CG et CIP, etc) [116, 115] mais aucun d'entre eux ne compare à la fois les discrétisations numériques, les méthodes d'intégration en temps et les techniques de stabilisation. De plus, il est plus commun de trouver dans la littérature des analyses semi-discrètes. Dans cette thèse, nous vous présenterons une étude partiellement et entièrement discrète (*semi discrete* et *fully discrete*) et tous ces résultats sont disponibles (en libre accès) [87, 86]. Via ces *analyses spectrales* dites de *von Neumann*, nous optimiserons les paramètres du schéma défini dans les techniques de stabilisation (notamment la CFL et le paramètre de stabilisation noté δ) pour toutes les combinaisons de schémas numériques énoncées précedemment. Un autre

point important est que tous les résultats seront soigneusement vérifiés numériquement à la fois sur des problèmes linéaires et non-linéaires (résolution des systèmes de Burger et Shallow Water) par l'évaluation de la stabilité et de l'ordre de convergence. Nous comparerons également les temps de calcul afin de converger vers la méthode la plus précise, permettant les meilleures performances en terme de temps de calcul. Il est important de noter que dans la littérature, il n'est pas commun de trouver une analyse spectrale bi-dimensionnelle, car à la différence de l'étude mono-dimensionnelle [88], l'étude bi-dimensionnelle fait intervenir un degré de liberté supplémentaire de la topologie du maillage, dont l'influence est également prise en compte. En particulier, dans cette thèse nous effectuerons pour l'analyse bi-dimensionnelle, une analyse de Fourier entièrement discrète en considérant deux configurations de maillage différentes et différents angles d'onde.
Nos conclusions finales concernant l'analyse de *von Neumann* et les tests numériques de convergence suggèrent que les éléments de *Cubature* combinés avec la stabilisation SSPRK et CIP ou OSS sont les combinaisons les plus prometteuses pour la résolution d'équations hyperboliques.

Ensuite dans le chapitre 6, nous proposerons une autre analyse comparative sur des cas tests faisant intervenir plusieurs complexités en hydrodynamique côtière, en résolvant les équations bi-dimensionnelles Shallow Water. Suite aux travaux effectués précedemment sur solutions lisses, nous évaluerons nos méthodes/approches numériques en présence de solutions non lisses et de topographies non constantes. En particulier, nous utiliserons entre autre une topographie discontinue, puis une solution initiale discontinue. D'une part nous étudierons le caractère dit *well-balanced* de la discrétisation en présence de bathymétrie (ie la capacité à préserver le lac au repos). Ensuite, pour améliorer la stabilité de la solution, nous ajouterons un terme dit à capture de choc. Une vérification de l'ordre de convergence sur solutions lisses, et une comparaison erreur/temps CPU seront aussi réalisées pour appuyer la pertinence de ces méthodes. Pour tout ce chapitre, nous choisirons de comparer les meilleures approches numériques Galerkin continues mises en avant dans les analyses précédentes. En particulier le CIP et l'OSS, combinées avec deux choix différents d'éléments finis continus: les polynômes lagrangiens sur les nœuds équidistants et les polynômes lagrangiens sur les nœuds de Cubature utilisant les schémas SSPRK d'intégration temporelle.
La principale contribution de cette comparaison sera de proposer une méthode entièrement explicite, sans matrice de masse, d'ordre élevé, *well-balanced* et à capture de choc pour résoudre les équations Shallow Water. Les résultats sont très prometteurs pour les applications d'ingénierie côtière dans les codes opérationnels. En effet, nous proposerons en particulier par l'utilisation des éléments de *Cubature*, une méthode numérique robuste en hydrodynamique côtière, permettant d'améliorer considérablement le temps de calcul comparé aux méthodes classiques.

Pour finir, dans le dernier chapitre 7 du manuscrit nous proposerons une mise en évidence de l'intérêt d'utiliser le *mass-lumping* dans un contexte opérationnel en soulignant le gain de temps de calcul. Les cas tests seront exécutés en utilisant une formulation Galerkin discontinue (pour une question d'implémentation), *well-balanced*, avec la même technique de capture de choc et les mêmes méthodes SSPRK que précedement, pour les éléments *Basic* et *Cubature*. Nous commencerons avec deux cas tests sphériques proposés dans [5]. Puis, nous effectuerons un benchmark de submersion de la Région Nouvelle-Aquitaine. Le premier cas test sphérique est un état stable (équilibre geostrophysique) qui n'utilise pas de technique à capture de choc. Pour ce cas test, nous réaliserons des tests de convergence et comparerons

les discrétisations numériques en termes de précision et de temps de calcul. Le second cas test sphérique correspond à l'ajout d'une perturbation qui parcourt le globe. La complexité de ce cas test est qu'il crée des champs de vorticité élevés pendant la simulation. Nous comparerons alors les champs de vorticité suivant la discrétisation spatiale choisie (avec ou sans *mass-lumping*). Nous terminerons par un cas test réel qui modélise la submersion provoquée par la tempête Xynthia (2010) aux *"Boucholeurs"* (Nouvelle-Aquitaine). La complexité du cas test est qu'il utilise des termes de viscosité entropique et doit aussi faire face au déplacement d'une interface sec/mouillé. Toutes les simulations sont réalisées dans le code Aerosol via la Plateforme Uhaina. Encore une fois, nous constaterons le gain en CPU notamment dans l'inversion de la matrice de masse. De plus, ces derniers tests mettront aussi en lumière différentes pistes d'améliorations dans le code Aerosol.

Contents

Abstract iii

Acknowledgements v

Résumé en francais vii

1 Introduction 1
- 1.1 Context and motivation . . . 1
- 1.2 State of the art . . . 2
- 1.3 Thesis accomplishments and outline . . . 5
  - 1.3.1 Scientific contributions . . . 5
  - 1.3.2 Outline of the manuscript . . . 6

2 Governing equations 9
- 2.1 Introduction to hyperbolic system . . . 9
- 2.2 Hyperbolic systems . . . 11
  - 2.2.1 Linear and non linear advection equations . . . 11
  - 2.2.2 Burgers' equations . . . 12
  - 2.2.3 Euler's equations . . . 13
  - 2.2.4 Shallow Water equations . . . 14
- 2.3 A notion of stability and stabilization methods . . . 14
  - 2.3.1 Entropy condition . . . 14
  - 2.3.2 Numerical stabilization . . . 16

3 Numerical methods 19
- 3.1 The Finite Element Methods . . . 20
  - 3.1.1 The Continuous Galerkin approach . . . 20
  - 3.1.2 The Discontinuous Galerkin approach . . . 21
  - 3.1.3 Approximation error . . . 22
- 3.2 Stabilization techniques . . . 23
  - 3.2.1 Streamline-Upwind/Petrov-Galerkin - SUPG . . . 23
    - Note on the SUPG technique applied to non scalar system . . . 24
  - 3.2.2 Continuous Interior Penalty - CIP . . . 25
  - 3.2.3 Orthogonal Subscale Stabilization - OSS . . . 26
- 3.3 Shock capturing technique . . . 27
- 3.4 Numerical Fluxes . . . 30
- 3.5 Finite Element Spaces and Quadrature Rules . . . 30
  - 3.5.1 The one-dimensional finite element spaces . . . 30
  - 3.5.2 The two-dimensional finite element spaces . . . 32
  - 3.5.3 Basic Lagrangian equispaced elements . . . 32
  - 3.5.4 Bernstein polynomials . . . 33

3.5.5 Cubature elements . . . 34
3.6 Time integration . . . 35
3.6.1 Explicit Runge–Kutta and Strong Stability Preserving Runge–Kutta schemes . . . 36
3.6.2 The *Deferred Correction* scheme . . . 36

4 Analysis of the one-dimensional formulation **39**
4.1 Introduction . . . 40
4.2 Basic spectral theory . . . 41
4.3 Fourier Analysis . . . 42
4.3.1 Preliminaries and time continuous analysis . . . 43
4.3.2 Space-continuous analysis for *Runge-Kutta* schemes . . . 47
4.3.3 Fully discrete analysis . . . 49
Methodology . . . 49
4.4 Results of the fully discrete spectral analysis . . . 51
4.4.1 Dispersion and damping . . . 54
4.5 A note on nonlinear stability . . . 58
4.6 Numerical Simulations . . . 60
4.6.1 Linear advection equation . . . 60
4.6.2 Application to Burgers' equation . . . 63
4.6.3 Application to Shallow Water equations . . . 66
4.7 Conclusion . . . 67

5 Extension to the two-dimensional formulation **69**
5.1 Introduction . . . 70
5.2 Fourier Analysis . . . 70
5.2.1 Preliminaries and time continuous analysis . . . 70
5.2.2 The eigenvalue system . . . 71
5.2.3 The fully discrete analysis . . . 73
5.2.4 Methodology . . . 74
5.2.5 Results of the fourier analysis using the *X* type mesh . . . 77
5.2.6 Comparison with a space-time split stability analysis . . . 79
5.2.7 Different mesh patterns . . . 81
5.2.8 Final results of the stability analysis . . . 82
5.2.9 Complementary analysis using viscosity terms . . . 85
Note on the stability of the method . . . 85
The von Neumann analysis . . . 86
5.3 Validation of the fourier analysis . . . 87
5.3.1 Linear advection equation test . . . 87
5.3.2 Shallow Water equations . . . 90
5.4 Numerical Simulations on arbitrary mesh . . . 92
5.4.1 Linear advection test . . . 93
5.4.2 Shallow Water equations . . . 95
5.4.3 Remark on the steady vortex case . . . 96
5.5 Conclusion . . . 96

6 Shallow Water equations: bore capturing and well balanced **99**
6.1 Introduction . . . 100
6.2 The well balanced formulation . . . 100
6.3 The lake at rest solution . . . 102
6.3.1 The lake at rest initial condition . . . 103
6.3.2 The Lake at rest with a perturbation . . . 104
6.4 The entropy viscosity technique . . . 105
6.5 The asymmetric break of dam on flat bathymetry . . . 107
6.5.1 Without shock capturing technique . . . 108
6.5.2 Using shock capturing technique . . . 109
6.6 Circular discontinuous perturbation on a lake at rest . . . 111
6.7 Conclusion . . . 115

7 Spherical and real benchmarks **117**
7.1 Introduction to the 3d/2d-covariant formulation . . . 118
7.1.1 Notations . . . 118
7.1.2 The Well Balanced property . . . 119
7.1.3 Fully diagonal mixed 3d/2d-covariant formulation . . . 120
7.2 Global atmospheric tests . . . 120
7.2.1 Steady-State . . . 120
7.2.2 Unstable Jet . . . 124
7.3 A real benchmark: Les Boucholeurs . . . 127
7.3.1 The datas . . . 127
7.3.2 The results . . . 128
7.4 Conclusion . . . 129

8 Conclusions and outlook **131**
8.1 The von Neuman numerical analysis . . . 131
8.2 Shallow water equations and applications . . . 132
8.3 Future investigations and developments . . . 133

A *Cubature* elements, definition and construction **135**
A.1 *Cubature* elements of degree 1 . . . 136
A.2 *Cubature* elements of degree 2 . . . 136
A.3 *Cubature* elements of degree 3 . . . 137

B Butcher tab **139**

C Backward Difference Formula **141**

D 1D dispersion curves **143**

E Fourier analysis, several 2D results **147**
E.1 Mesh types and degrees of freedom . . . 147
E.2 Fourier analysis results - Optimal Parameters . . . 147
E.3 Fourier analysis results - stability area . . . 149

F Shallow Water tests, additional results **157**
F.1 Perturbation of the lake at rest 157
F.1.1 With a smooth topography 157
F.1.2 With a non-smooth topography 158
F.2 Global atmospheric tests - Unstable jet 160

Bibliography **163**

List of Figures

1.1 Modeling of submersion - Nouvelle-Aquitaine: Arcachon and La-Teste-de-Buch [92] . . . 3

2.1 Solution u at time $t = 0$ in blue and $t = T$ in green at representative characteristic curve at $X(0) = x_1$ and $X(0) = x_2$. . . . 12
2.2 Rarefaction wave at left and shock at right. . . . 12
2.3 Definition of parameters. . . . 14
2.4 Lake at rest simulation [109]. Representation of the water elevation at $t_f = 1s$. At left: using stabilization technique, at right: without stabilization technique. . . . 16

3.1 Finite elements with 3 DOF at left, 6 DOF on the middle and 10 at right. . . . 20
3.2 Discontinuous Galerkin configuration at left - Continuous Galerkin confirguration at right. . . . 22
3.3 Lagrange polynomial basis functions defined on equidistant points, $p = 3$. 30
3.4 Bernstein polynomial basis functions defined on equidistant points, $p = 3$. 31
3.5 Basis functions of *Basic Lagrangian equispaced elements* for $p = 3$ and $\alpha_3 = 0$ . . . 33
3.6 Basis functions of *Basic Bernstein equispaced elements* for $p = 3$ and $\alpha_3 = 0$ 34
3.7 Comparison of the equispace repartition at left and the cubature repartition at right for elements of degree $p = 3$. . . . 34
3.8 Subtimesteps inside the time step $[t^n, t^{n+1}]$ . . . 36

4.1 Phase ω (left) and amplification ϵ (right) with *Basic* elements without stabilization for $\mathbb{P}_1, \mathbb{P}_2$ and $\mathbb{P}_3$. . . . 46
4.2 Phase ω (left) and amplification ϵ (right) with *Basic* elements with OSS stabilization for $\mathbb{P}_1, \mathbb{P}_2$ and $\mathbb{P}_3$. . . . 46
4.3 Dispersion analysis, SSPRK3 schemes . . . 47
4.4 Dispersion analysis, RK schemes . . . 48
4.5 Dispersion analysis, SSPRK4 schemes . . . 48
4.6 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 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. . . . 52
4.7 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 *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. . . . 52

4.8 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 *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. 53
4.9 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. 53
4.10 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 OSS 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. 54
4.11 Comparison of dispersion in the fully discrete case, using coefficients from 4.3, *Cubature* elements, DeC scheme and OSS 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 . . 56
4.12 Comparison of dispersion in the fully discrete case, using coefficients from 4.3, *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. 56
4.13 Error decay for linear advection with the OSS stabilization and SSPRK. $\mathbb{P}_1$, $\mathbb{P}_2$ and $\mathbb{P}_3$ elements are, respectively, in blue green and red. 61
4.14 Error for linear advection problem (4.42) with respect to computational time for all elements and stabilization techniques: DeC on the left, SSPRK on the right 62
4.15 Solution of linear advection equation with discontinuous initial condition using *Cubature* elements and SSPRK schemes: $\mathbb{P}_1$ at left, $\mathbb{P}_2$ at the center and $\mathbb{P}_3$ at right. 63
4.16 Error for Burgers' equation (4.44) with respect to computational time for all elements and stabilization techniques: DeC on the left, SSPRK on the right 65
4.17 Non linear instabilities for Burgers' equation (4.44) when $t_f > t_s$ using *Cubature* elements and SSPRK schemes: $\mathbb{P}_1$ at left, $\mathbb{P}_2$ at the center and $\mathbb{P}_3$ at right. 66
4.18 Error for Shallow Water equations (4.47) with respect to computational time for all elements and stabilization techniques: DeC on the left, SSPRK on the right 68

5.1 The *X* type triangular mesh. At left, the *Basic* finite element discretisation with $\mathbb{P}_2$ elements. At right, the grid configuration for $\tilde{\mathbb{P}}_2$ *Cubature* elements. The red square represents the periodic elementary unit that contains the degrees of freedom of interest for the Fourier analysis 72
5.2 Dispersion curves using *Cubature* $\tilde{\mathbb{P}}_2$ elements, the CIP stabilization technique, and a wave angle $\theta = 5\pi/4$. Phases ω (left) and amplifications ϵ (right). 73

5.3 Comparison of dispersion curves ω_i and damping coefficients ϵ_i, for *Cubature* $\tilde{\mathbb{P}}_2$ elements, with SSPRK time discretization and OSS stabilization: $\Phi = 0$ and $\Phi = 3\pi/16$. 76
5.4 Plot of $\log(\max_i \epsilon_i)$ for *Cubature* $\tilde{\mathbb{P}}_2$ elements, SSPRK time discretization and OSS stabilization. The blue and light blue region is the stable one. At the left only for $\Phi = 3\pi/16$, at the right we plot the maximum over all Φ. 76
5.5 Damping coefficients $\log(\max_i \epsilon_i)$ for $\mathbb{B}_3$ *Bernstein* elements and the DeC method with, from left to right, SUPG, OSS and CIP stabilization. 77
5.6 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ *Basic* elements with SSPRK scheme and OSS stabilization . . . 78
5.7 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ *Basic* elements with SSPRK scheme and CIP stabilization . . . 78
5.8 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ *Cubature* elements with DeC scheme and OSS stabilization . . 78
5.9 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ *Cubature* elements with DeC scheme and CIP stabilization . . . 78
5.10 Logarithm of the amplification coefficient $\log(\max_i(\varepsilon_i))$ for SUPG stabilization with $\tilde{\mathbb{P}}_3$ *Cubature* elements and the SSPRK method. Unstable region in yellow, optimal (5.15) parameters in red 79
5.11 Eigenvalues of $\widetilde{A}$ using *Cubature* discretization and the SUPG stabilization (varying k) and stability area of the SSPRK method. In red the stable eigenvalues, in blue the unstable ones. 80
5.12 The T type triangular mesh with degrees of freedom in blue and periodic unit in the red square for the Fourier analysis 81
5.13 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ *Basic* elements with SSPRK scheme and CIP stabilization . . . 82
5.14 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ *Cubature* elements with SSPRK scheme and OSS stabilization 83
5.15 Maximum logarithm of the amplification coefficient $\log(\max_i(\varepsilon_i))$ for $\tilde{\mathbb{P}}_3$ *Cubature* elements on the X and T meshes 83
5.16 Logarithm of the amplification coefficient $\log(\max_i(\varepsilon_i))$ for $\tilde{\mathbb{P}}_3$ *Cubature* elements on the X mesh 83
5.17 T mesh - Von Neumann analysis using an additional viscosity term (see (5.18)). *Cubature* $\tilde{\mathbb{P}}_3$ elements with SSPRK and OSS. Comparison of different μ. 87
5.18 Linear advection simulation on the X mesh 88
5.19 Error decay for linear advection problem with different elements and OSS stabilization and SSPRK time discretization: $\mathbb{P}_1$ in blue, $\mathbb{P}_2$ in green and $\mathbb{P}_3$ in red 88
5.20 Error for linear advection problem (5.21) with respect to computational time for SSPRK time discretization, comparing *Basic* and *Cubature* elements and all stabilization techniques 89

5.21 Initial water elevation and velocity field (red arrows) - Travelling vortex test case, $\Omega = [0,2] \times [0,1]$. 91
5.22 Error for Shallow Water system (5.22) with respect to computational time for SSPRK method with *Cubature* (left) and *Basic* (right) elements and CIP and OSS stabilizations. 91
5.23 Unstructured mesh on $\Omega = [0,2] \times [0,1]$. 92
5.24 Error for linear advection problem (5.21) with respect to computational time for all elements and stabilization techniques 94
5.25 Error for Shallow Water problem (5.22) with respect to computational time for all elements and stabilization techniques 95

6.1 Lake at rest solution with a smooth topography, $t_f = 1s$ using Basic $\mathbb{P}_1$ discretization with the OSS stabilization technique and the SSPRK(3,2) scheme. Amplification factor of water instabilities = 100. 103
6.2 Lake at rest solution with a discontinuous topography, $t_f = 0.02s$ using Basic $\mathbb{P}_1$ discretization with the OSS stabilization technique and the SSPRK(3,2) scheme. 103
6.3 Propagation of the perturbation eq. (6.15) at different time step using *Basic* $\mathbb{P}_1$ elements with the OSS stabilization technique and the SSPRK(3,2) scheme. 104
6.4 2D view of the free surface elevation at $t = 0.48s$ with the smooth topography eq. (6.13). All simulation are performed using *Cubature* elements, the OSS stabilization technique, and SSPRK schemes. 105
6.5 Initial water elevation and velocity field (red arrows) - Travelling vortex test case, $\Omega = [0,2] \times [0,1]$. 106
6.6 Error for Shallow Water problem (5.22) with respect to computational time for all elements, the linear stabilization techniques and the additional viscosity term. 106
6.7 Asymmetric dam break. At left, visualisation of the domain Ω_h and initial conditions. At right: 3d view of the initial state. 108
6.8 Asymmetric dam break. At left, 3d view of the final state at $t = 7.2$. At right: 2d view from the top of the final state, contour line of water elevations 108
6.9 2D view of the free surface elevation at $t = 7.2s$, *Basic* elements with the OSS stabilization technique. 109
6.10 2D view of the free surface elevation at $t = 7.s$, *Cubature* elements with the OSS stabilization technique. 109
6.11 Asymmetric dam break: water height at time $t = 7.2s$. Data extracted along the line $y = 132m$. Use of the OSS stabilization technique. 110
6.12 Asymmetric dam break: water height at time $t = 7.2s$. Data from [109], extracted along the line $y = 132m$ 111
6.13 Asymmetric dam break: water height at time $t = 7.2s$. Data extracted along the line $y = 132m$. Use of the *Basic* discretization, the OSS and entropy stabilization technique. 111
6.14 Asymmetric dam break: water height at time $t = 7.2s$. Data extracted along the line $y = 132m$. Use of the *Basic* discretization, the CIP and entropy stabilization technique. 112

6.15 Asymmetric dam break: water height at time $t = 7.2s$. Data extracted along the line $y = 132m$. Use of the *Cubature* discretization, the OSS and entropy stabilization technique. . . . 112
6.16 Asymmetric dam break: water height at time $t = 7.2s$. Data extracted along the line $y = 132m$. Use of the *Cubature* discretization, the CIP and entropy stabilization technique. . . . 113
6.17 Perturbation of a lake at rest, initial solution. . . . 113
6.18 Final state of the perturbation eq. (6.17) using *Basic* $\mathbb{P}_1$ elements with the OSS stabilization technique and the SSPRK(3,2) scheme. . . . 113
6.19 2D view of the free surface elevation at $t = 0.18s$ with the oscillating topography eq. (6.16). All simulation are performed using SSPRK schemes. . . . 114

7.1 Initial state of the Global atmospheric test at left, representation of velocity field at right. The green scale represents the momentum magnitude. . . . 121
7.2 Global steady state simulation, convergence rates. . . . 121
7.3 Global steady state simulation, error with respect to computational time. . . . 122
7.4 Global steady state simulation, L_1 error with respect to computational time. Comparison of the use of two different solvers. . . . 123
7.5 Initial state of the Unstable Jet test. . . . 125
7.6 Unstable Jet - 2D view from the northen pole of the vorticity field contour levels (from $-1.1e-4$ to $1.5e-4$) at $t_f = 6$ days. $h_K = 223km$ . . . 125
7.7 Approximated solution of the vorticity field after 6 days. Sources from J. Galewsky et al. [51]. . . . 126
7.8 Representation of the test case: Les Boucholeurs (Nouvelle-Aquitaine). . . . 127
7.9 Snapshot views of the marine submersion in the Boucholeurs sector during the Xynthia storm at different hours: at left 00:20 am ($t_s \approx 14h$), on the middle 03:50 am ($t_s \approx 18h$), at right 05:30 am ($t_s \approx 20h$). Source Brgm, model MARS-2DH. . . . 128
7.10 Les Boucholeurs simulation, Xynthia storm (2010) - 2D view from the top of the water elevation $\eta = h + b$ at different time steps. . . . 129

A.1 Comparison of two element of degree three: at left the classical one $\mathbb{P}_3$, at right the *Cubature* one $\tilde{\mathbb{P}}_3$. . . . 135

D.1 Dispersion and damping coefficients for *Basic* elements, with DeC and SSPRK methods and all stabilization techniques . . . 144
D.2 Dispersion and damping coefficients for *Cubature* elements, with DeC and SSPRK methods and all stabilization techniques . . . 145
D.3 Dispersion and damping coefficients for *Bernstein* elements, with DeC and SSPRK methods and all stabilization techniques . . . 146

E.1 Degrees of freedom and periodic unit for different mesh patterns and elements of degree 3 . . . 148
E.2 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1, \mathbb{P}_2, \mathbb{P}_3$ *Basic* elements with SSPRK scheme and OSS stabilization. . . . 150
E.3 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1, \mathbb{P}_2, \mathbb{P}_3$ *Basic* elements with SSPRK scheme and CIP stabilization. . . . 150

E.4 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1, \mathbb{P}_2, \mathbb{P}_3$ *Basic* elements with SSPRK scheme and SUPG stabilization. . 151
E.5 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1, \tilde{\mathbb{P}}_2, \tilde{\mathbb{P}}_3$ *Cubature* elements with SSPRK scheme and OSS stabilization. 151
E.6 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1, \tilde{\mathbb{P}}_2, \tilde{\mathbb{P}}_3$ *Cubature* elements with SSPRK scheme and CIP stabilization. . 152
E.7 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1, \tilde{\mathbb{P}}_2, \tilde{\mathbb{P}}_3$ *Cubature* elements with SSPRK scheme and SUPG stabilization. 152
E.8 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1, \tilde{\mathbb{P}}_2, \tilde{\mathbb{P}}_3$ *Cubature* elements with DeC scheme and OSS stabilization. . . 153
E.9 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1, \tilde{\mathbb{P}}_2, \tilde{\mathbb{P}}_3$ *Cubature* elements with DeC scheme and CIP stabilization. . . 153
E.10 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1, \tilde{\mathbb{P}}_2, \tilde{\mathbb{P}}_3$ *Cubature* elements with DeC scheme and SUPG stabilization. . 154
E.11 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1, \mathbb{P}_2, \mathbb{P}_3$ *Bernstein* elements with DeC scheme and OSS stabilization. . . 154
E.12 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1, \mathbb{P}_2, \mathbb{P}_3$ *Bernstein* elements with DeC scheme and CIP stabilization. . . 155
E.13 $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1, \mathbb{P}_2, \mathbb{P}_3$ *Bernstein* elements with DeC scheme and SUPG stabilization. . 155

F.1 2D view of the free surface elevation at $t = 0.48s$ with the smooth topography eq. (6.13). All simulation are performed using *Basic* elements, the OSS stabilization technique, and SSPRK schemes. 157
F.2 2D view of the free surface elevation at $t = 0.48s$ with the smooth topography eq. (6.13). All simulation are performed using *Basic* elements, the CIP stabilization technique, and SSPRK schemes. 158
F.3 2D view of the free surface elevation at $t = 0.48s$ with the smooth topography eq. (6.13). All simulation are performed using *Cubature* elements, the CIP stabilization technique, and SSPRK schemes. 158
F.4 2D view of the free surface elevation at $t = 0.48s$ with the non-smooth topography eq. (6.14). Both discretization use the OSS stabilization technique and SSPRK schemes. 159
F.5 2D view of the free surface elevation at $t = 0.48s$ with the non-smooth topography eq. (6.14). Both discretization use the CIP stabilization technique and SSPRK schemes. 159
F.6 Approximated solution of the vorticity field after 4 days. Sources from J. Galewsky et al. [51]. 160

F.7 Unstable Jet - 2D view from the northen pole of the vorticity field contour levels (from $-1.1e-4$ to $1.1e-4$) at $t_f = 4$ days. $h_K = 223km$ 161

List of Tables

4.1 Summary table of number of modes per systems. 46
4.2 Optimized CFL and penalty coefficient δ in parenthesis, only maximizing CFL. The sign / means unconditionally unstable.
* These values do not allow to decrease the CFL. 55
4.3 Optimized CFL and penalty coefficient δ in parenthesis, minimizing η_u. The sign / means unconditionally unstable.
* These values do not allow to decrease the CFL. 57
4.4 Optimized CFL and penalty coefficient δ in parenthesis, minimizing η_ω. The sign / means unconditionally unstable.
* These values do not allow to decrease the CFL. 58
4.5 Optimized CFL and penalty coefficient δ in parenthesis, stable for all smaller CFLs 58
4.6 Summary table of convergence orders, using coefficients obtained by minimizing η_u in table 4.3.
** These values are computed using parameters from the minimization of η_ω in table 4.4 61
4.7 Summary table of convergence order, using coefficients obtained in table 4.3. The sign / means unstable. 64
4.8 Summary tab of convergence order, using coefficients obtained by minimizing η_u. The sign / means unstable. 67

5.1 X mesh: Summary table of number of modes per systems. 73
5.2 X mesh: Optimized CFL and penalty coefficient δ in parenthesis, minimizing η_u.
"/" means that the fourier analysis shown that the scheme is unstable.
* These values are not reliable, see section 5.2.6. 79
5.3 Number of modes in the periodic unit for different elements in the T mesh . 81
5.4 Optimized CFL and penalty coefficient δ in parenthesis, combining the two mesh configurations. The values denoted by * are not the optimal one, but they lay in a safer region, see Section 5.2.6. The values marked by ** cannot be used on the T mesh. "/" means that it is unstable for every parameter . . 84
5.5 Convergence order for all schemes on linear advection test, using coefficients obtained in table E.1.
"/" means that the Fourier analysis showed that the scheme is unstable. . . . 89
5.6 Convergence order on Shallow Water, using coefficients obtained in E.1.
"/" means that the fourier analysis shown that the scheme is unstable. 92
5.7 Convergence order for linear advection on unstructured mesh, using coefficients obtained in table 5.4.
** These values are found using only the X mesh (see fig. 5.15).
"/" means that the scheme is clearly unstable. 93

5.8 Convergence order of methods using *Cubature* $\tilde{\mathbb{P}}_3$ elements and viscosity term (5.18) with tuned parameters . . . 94
5.9 Convergence order on Shallow Water for unstructured mesh, using coefficients obtained in table 5.4.
** These values are found using only the X mesh (see fig. 5.15).
"/" means that the scheme is clearly unstable. . . . 95
5.10 Summary tab of convergence order, steady vortex, $t_f = 0.1s$.
"/" means that the scheme is clearly unstable. . . . 96
5.11 Summary tab of convergence order, unsteady vortex, $t_f = 0.1s$.
"/" means that the scheme is clearly unstable. . . . 96

6.1 Convergence order for Shallow Water problem (5.22) on unstructured mesh, $t_f = 0.5s$. . . . 106
6.2 CFL, penalty coefficient δ and viscosity term c_E in parenthesis used for convergence tests: CFL (δ, c_E) . . . 107

7.1 Global steady state. Summary tab of numerical results. At left: the numerical schemes, at right: results. . . . 122
7.2 Global steady state. Summary tab of cpu-time repartition for the mesh $h_k = 223km$, in parenthesis the pourcentage). . . . 123
7.3 Unstable Jet. Summary tab of cpu-time repartition for the mesh $h_k = 223km$, in parenthesis the percentage. $t_f = 6$ days. . . . 126
7.4 Summary tab of numerical results. At left: the numerical schemes, at right: results. . . . 128

B.1 Butcher Tableau of RK methods . . . 139
B.2 DeC coefficients for equispaced subtimesteps. . . . 139
B.3 Butcher Tableau of SSPRK methods . . . 140

E.1 X mesh: Optimized CFL and penalty coefficient δ in parenthesis. The symbol "/" means that the fourier analysis for the scheme results always in instability. The values denoted by * are not the optimal one, but they lay in a safer region, see Section 5.2.6. . . . 147
E.2 T mesh: Optimized CFL and penalty coefficient δ in parenthesis. The symbol "/" means that the fourier analysis for the scheme results always in instability. . . . 149

F.1 Unstable Jet. Summary tab of cpu-time repartition for the mesh $h_k = 223km$, in parenthesis the pourcentage. $t_f = 4$ days. . . . 160

Chapter 1

Introduction

Chapter Abstract
Due to the effect of climate change during this century, the rising sea levels and storms are threatening more and more urban areas near the coast. Urban areas in the vicinity of the sea, of the ocean, as well as of rivers are exposed to increasing submersion risks. To mitigate the consequences of these events and to be better prepare to react, it is necessary to carry out risk assessment studies with accuracy and resolution requirements beyond traditional methods: in particular, one needs to take into account the relevant phenomena (such as wave effect), the uncertainties affecting them, and have the possibility to flexibly describe phenomena at different large and small spatial resolutions.

Outline

1.1 Context and motivation **1**
1.2 State of the art **2**
1.3 Thesis accomplishments and outline **5**
1.3.1 Scientific contributions 5
1.3.2 Outline of the manuscript 6

1.1 Context and motivation

Inhabited centers near the sea or rivers subject to flooding are exposed to the risk of submersion. These risks are linked either to the rise in sea level during strong tides or storms, or to catastrophic events such as tsunamis. Coastal management guidelines (e.g. the DGPR2014 Report [45]) make it necessary to forecast risks that go beyond the most traditional methods of submersion modeling concerning: physical phenomena (e.g. effect of waves), the required parameters (dynamics of submersion, current speeds), the taking into account of uncertainties, the spatial resolution which can go as far as interactions with the building and the structures.

For the French territory, we have high resolution MNT and MNE type data (digital terrain and elevation model), or maps of large urban agglomerations via platforms such as OpenStreetMap (www.openstreetmap.org, for France data also available on www.data.gouv.fr) or as the OpenData portal for Bordeaux Métropole (data.bordeaux-metropole.fr). While these data allow city-scale forecasts [77], the necessary resolutions lead to computation times so

long that they cannot be used in an operational setting.

The availability of codes with very high performance and precision is essential. This project follows on from the research carried out by the CARDAMOM[1] team at the INRIA Bordeaux Sud-Ouest center. In particular, the thesis will ultimately contribute to the UHAINA project[2] an open source operational platform for the simulation of the impact of waves at the coast. UHAINA is a medium / long term action undertaken by the CARDAMOM team, the EPOC laboratory (UMR 5805, Dr. P. Bonneton), CAGIRE team (V. Perrier), the Institute of Mathematics of Bordeaux (UMR 5251, Dr. D. Lannes) and the Montpellier Institute Alexander Grothendieck (UMR 5149, Dr. F. Marche). The objective of this action is to provide French actors involved in coastal risk assessment with a forecasting tool using the most modern digital models and techniques, as well as a high performance implementation based on libraries developed at INRIA as Aerosol[3] for the hydrodynamic core, and PaMPA[4] and SCOTCH[5] for parallelism.

In this context, this thesis focuses on upstream research issues: the study of improved approaches to coastal hydrodynamics. The objective of the thesis will be to obtain an efficient/optimal model from the point of view of precision (for a given number of computational unknowns), robustness and speed of computation. The implementation in the hydrodynamic core of UHAINA will make it possible to benefit from the performance of this kernel and to guarantee, once these techniques are mature enough, a transfer to the users of this platform. I will benefit from the interaction of the CARDAMOM team with BRGM, also involved in the co-financing project submitted to the Nouvelle Aquitaine Region.

1.2 State of the art

Nowadays, we have high-resolution data and maps of big urban agglomerations 1.1. If these data allow forecasts on an urban scale, the resulting resolutions lead to significant computing times when considering their application in an operational context. This makes the availability of high order, adaptive, geometrically flexible, massively parallel codes an essential building block for applications. To model submersion and inundation, we can use the so-called Shallow Water equations: a set of depth averaged balance law, with an underlying hyperbolic character. These equations allow to account with surprising accuracy for wave breaking, and wave runup, and have some potential to model wave propagation, especially for long waves (as e.g. storm or tsunami waves). This model finds also applications in hydrology, and meteorology (see [31, 119, 120] and references therein).
To resolve them with high accuracy, we need to design a solver allowing to handle propagation with great accuracy, to deal with discontinuous features, to manage inundation and flooding in a complex environment with as much geometrical flexibility as possible. The aims of my work is to find a such solver capable of being sufficiently accurate and having a good computation times.

[1] https://team.inria.fr/cardamom/
[2] https://gitlab.inria.fr/uhaina1/uhaina/-/wikis/home
[3] https://team.inria.fr/cardamom/aerosol/
[4] https://project.inria.fr/pampa/
[5] http://www.labri.fr/perso/pelegrin/scotch/

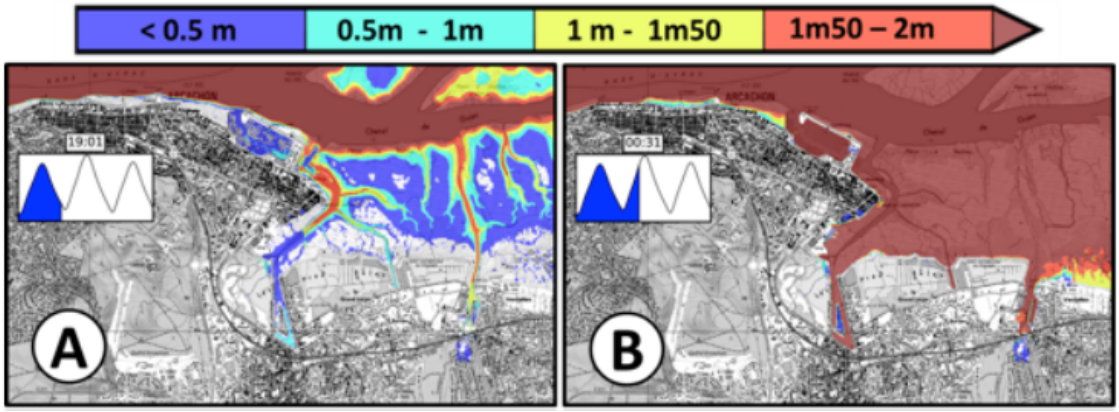


FIGURE 1.1: Modeling of submersion - Nouvelle-Aquitaine: Arcachon and La-Teste-de-Buch [92]

The coastal hydrodynamic is not the only application of fluid dynamics problem. More generally, in aerospace several engineering application can be mentioned: gas dynamics, structure/fluid interaction, etc. A fundamental characteristic of nonlinear conservation laws is that discontinuities can appear during simulation even from smooth initial data. The main issue is to deal with these discontinuities both mathematically and computationally. Historically, method used to discretize hyperbolic system are finite difference and finite volume methods [78], because they can describe shocks relatively well. However, even is the low order of accuracy is intuitive to create, the extension to higher order is not straightforward. Galerkin methods have also been considered in order to achieve high order spatial convergence [61, 35]. It is well known that the standard Galerkin finite element method is in general not well suited for the solution of advection and advection–diffusion problems. There are two possibilities to enhance stability while keeping accuracy. The first one is the Discontinuous Galerkin [36] formulation, which considers discontinuous approximated solutions. The second one uses the Continuous Galerkin formulation, adding an appropriately designed stabilization term.

One of the most popular stabilization techniques is the streamline-upwind Petrov-Galerkin (SUPG), introduced in [65] (see also [67, 21]). The formulation is strongly consistent in the sense that the stabilization contains the full residual and vanishes when applied to exact solutions of the PDE. In this work we will also consider symmetric stabilization techniques which are somewhat simpler to implement compared to residual based stabilization methods. The first alternative, is the continuous interior penalty (CIP) stabilization used in [25, 27, 23]. This method has been developed by E. Burman and P. Hansbo in [24], but it can be seen as a variation of the method originally proposed by Douglas and Dupont [46]. The method stabilizes the Galerkin formulation by adding a least-square term proportional to the jump of the gradient of the derivatives of the solution across the cell interfaces. The CIP introduces high order viscosity to the formulation, allowing the solution to tend to the vanishing viscosity limit. This term is computed on the solution at the previous time step, so it does not affect the structure of the LHS matrix. The second alternative is the Orthogonal Subscale Stabilization (OSS) approach. Originally introduced as Pressure Gradient Projection (PGP) in [38] for Stokes equations, it was extended to the OSS method in [37, 8] for different problems with numerical instabilities, such as convection–diffusion–reaction problems. The method stabilizes the Galerkin formulation by penalizing gradient fluctuations.

Time integration also plays a major role. The methods used to achieve high order temporal convergence in this work are the general *Runge-Kutta* ones (RK), as well as *Strong Stability Preserving Runge-Kutta* introduced in [117] (SSPRK). Numerous SSPRK variant exist [117, 118, 112, 56, 30], we will compare some of them numerically in order to select the most stable one. These explicit methods will be also compared to the *Deferred Correction* methods (DeC), which were originally introduced in [47] as explicit solvers of ODEs, but soon implicit [90] or positivity preserving [100] versions and extensions to PDE solvers [2] have appeared. In [2, 104, 107] 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. The use of Bernstein polynomials is recommended in [2], but other choices are also possible as e.g. *Cubature* elements under some conditions.

This allows to introduce the last fundamental aspect of finite element methods : the polynomial approximation. In practice, in this thesis, triangular elements are used to handle complex geometries (in 2D). The classical approach consists in defining degree of freedom place uniformly in the triangle and define corresponding basis functions (which are in general Lagrange basis functions). Following the FEM discretization procedure, the resolution of a hyperbolic system consists in solving a system by inverting a matrix called *the mass matrix*. The cpu-time of this operation cannot be neglected. A way to reduce this cost is to resort to the *mass-lumping* principle, possibly without affecting the accuracy. Among the possible solutions, we can cite *Cubature* elements, introduced by G. Cohen and P. Joly in 2001 [39] which are an extension of Lagrange polynomials with the goal of optimizing the underlying quadrature formulas error (all the details of such elements can be found in [39, 55, 69]). Similar techniques have been used to minimize the interpolation error using *Fekete* and *Gauss–Lobatto* points [70, 115, 122]. However, *Gauss–Lobatto* quadrature points are only known for tensor-product domains such as the line and square, making it unclear how to extend a Gauss–Lobatto numerical method to non-tensor-product domains like the triangle. Since *Fekete* points are known to be the *Gauss–Lobatto* points on the line [50] and in the d-dimensional cube [14], Fekete points are one possible generalization *of Gauss–Lobatto* points for the triangle. The objective of these polynomials is to use the points of the Lagrangian interpolation of the polynomials as quadrature points. Another type of basis function will be also used in this work: Bernstein polynomial functions. They verify additional properties besides the one for Lagrangian points. They form a partition of unity, the basis functions are non-negative in any point of the triangle, and, hence, their integrals are positive. These properties lead also to the fact that the value at each point is a convex combination of the coefficients of the polynomials, hence, it is easy to bound the minimum and maximum value of the function by the minimum and maximum of the coefficients. This has been used in different techniques to preserve positivity of the solution [7, 75].

In this PhD, I will describe, analyze and optimize these methods which solve hyperbolic problems. The analysis will be performed in the Fourier way called the *Fourier* or the *Spectral analysis*. Several works show the method applied to one specific case (DG, CG and CIP, etc) [116, 115] but none of those compare in a same time numerical discretizations, time integration methods and stabilization techniques. Moreover, in the fully discrete study, we will optimize scheme parameters defined in stabilization techniques. Another point is that all results are also used in numerical test using linear and non-linear systems to validate our study. We will also compare the time of computation in order to converge to the most

accurate method, allow performance in terms of time of computation.

1.3 Thesis accomplishments and outline

1.3.1 Scientific contributions

The thesis has contributed to different scientific projects. In particular, by the implementation of high order methods raised in an object-oriented finite element library used in INRIA teams CARDAMOM and CAGIRE, as well as at BRGM[6]. The validation of these methods is done on academics cases from Aerosol as well as large scale cases (example of a spherical case, comparison of the results obtained with the work of L. Arpaia et al. [5]) and of real interest for the Nouvelle Aquitaine region (example of Boucholeurs - 17340) in close collaboration with the BRGM d'Orléans via the Uhaina platform.

In this context, I worked in particular with Benjamin Lux, Mario Ricchiuto, Héloïse Beaugendre and Vincent Perrier in the implementation of the *continuous Galerkin* formulation in Aerosol. As well as the implementation of *Cubature* finite element discretization and stabilization method. The aims of this work are related to the optimization of the time of computation. The first aspect is to compare for a same test case, using a *continuous* or *discontinuous Galerkin* approach, different spatial discretizations in terms of accuracy and time of computation. The first spatial discretization used high order quadrature formula to compute the entire matricial system which is not straight to solve. The second discretization is based on *Cubature* quadrature points, and allows to obtain a matricial system "easy to solve" in the sense that it is mass matrix inversion free. The second aspect of this work is the impletementation of a stabilized Continuous Galerkin method in Aerosol. This method allows in particular to reduce considerably the number of degree of freedom and so the length of the system to solve. In this direction, and combined with the use of *Cubature* elements, we expect to optimize the time of computation in Aerosol.

Other scientific contributions have been made, notably in collaboration with the University of Zurich concerning the mathematical analysis of the methods used. A first monodimensional analysis work has been published in the Journal of Scientific Computing [88], and the extension for the two-dimensional case will soon be submitted for this same journal. In the context of these two publications, numerous spectral analysis results are available as open source in [87, 86]. The main contributions of this work are:
1/ Study continuous finite element discretizations for one and two-dimensional hyperbolic partial differential equations.
2/ Provide a multidimensional 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.
3/ Compare three different choices for the continuous finite element space.
4/ Compare different time stepping strategies, namely Runge-Kutta (RK), strong stability preserving RK (SSPRK) and deferred correction time integration methods.
5/ To understand the effects of these choices, we compare all the different combinations in

[6]Bureau de recherches géologiques et minières https://www.brgm.fr/fr

terms of accuracy and stability. The results are verified numerically both on linear and non-linear problems, and error-CPU time curves are provided and compared.

In the continuity of this work, we also proposed an extension of these numerical methods applied to coastal hydrodynamic. I.e. taking into account the topography, and shock formation during the wave propagation. This last work is also in preparation as a third publication.

Still in the context of these three publications, I developed from scratch a full Finite Element code that I called *Parasol*, as well as a graphic interface able to reproduce the mono-dimensional spectral analysis using all combination of scheme. This code is available for use by the CARDAMOM team in order to run easily spectral analysis and academic numerical test.

To summarize, this thesis has contributed scientifically through three different publications:

- **Spectral analysis of continuous FEM for hyperbolic PDEs: influence of approximation, stabilization, and time-stepping.** published in the Journal of Scientific Computing [88] (see chapter 4),
- The two dimensional extension, **Spectral analysis of high order continuous FEM for hyperbolic PDEs on triangular meshes: influence of approximation, stabilization, and time-stepping**, which is in preparation and will be submit very soon in the same journal (see chapter 5),
- A extension to Shallow Water equations and applications : **A high order mass-matrix free, stabilized and well-balanced continuous FEM for Shallow water equations**, which is also in preparation and will be submit soon (see chapter 6).

Additionally, through two different talks:

- **A high-order stabilized finite element method to solve advection equation**, during a research visit at the *Institute of Mathematics, Univ. of Zurich*, in January 2020.
- **Fourier analysis of continuous FEM for hyperbolic PDEs: influence of approximation and stabilization terms**, during the *ICOSAHOM* conference, in July 2021.

1.3.2 Outline of the manuscript

The manuscript is organized as follows: this chapter introduce the context and the motivation of the study, a synthesis of what has been done in the research areas of our studies (state of the art) and thesis accomplishment. In the second and the third chapter, we introduce governing equations (see chapter 2) and numerical methods used in this work (see chapter 3). The two next chapters are the study of the linear stability of scalar hyperbolic equations and of several stabilized variants using Fourier's analysis for the mono-dimensional case (see chapter 4) and the two-dimensional case (see chapter 5). 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. For these both chapters, we then perform numerical tests to illustrate our results and conclude about the most efficient numerical scheme. To finish, the sixth and the seventh chapter present a comparison of our numerical methods in a realistic context. We propose in particular a *continuous*

Galerkin well-balanced and shock capturing formulation in chapter 6. And then, we show the benefit of using mass-lumping in the *discontinuous Galerkin* context in chapter 7.We finally conclude by a synthesis of the work done during the PhD, the improvements and scientific contributions, and finally perspectives in chapter 8.

Chapter 2

Governing equations

Chapter Abstract
In this chapter, we introduce some generalities concerning the partial differential equations used in this work, and in particular hyperbolic equations through several examples. The particularity of these equations is that they evolve discontinuous solutions in time and in space, even for smooth initial and boundary data. We will mainly follow [113, 79].

Outline

2.1 Introduction to hyperbolic system 9
2.2 Hyperbolic systems 11
2.2.1 Linear and non linear advection equations 11
2.2.2 Burgers' equations 12
2.2.3 Euler's equations 13
2.2.4 Shallow Water equations 14
2.3 A notion of stability and stabilization methods 14
2.3.1 Entropy condition 14
2.3.2 Numerical stabilization 16

2.1 Introduction to hyperbolic system

There are three categories of partial differential equations:

- Elliptic equations, as for example the *Poisson* equation:

$$-\Delta u = f$$

where $u(x)$ is the unknown, $x \in \Omega \subset \mathbb{R}^n$ and f is given.

- Parabolic equations, which often model transient evolution irreversible phenomena associated with diffusion processes. The heat equation is a prototype:

$$\frac{\partial u}{\partial t} - \Delta u = f$$

where $u(x,t)$ is the unknown, $x \in \Omega \subset \mathbb{R}^n$, $t \geqslant 0$, and f is given.

➢ Hyperbolic equations which model time dependent transport phenomena as for example wave propagation. We identify two prototypes for this class of PDE:

– The advection equation

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$$

where $u(x,t)$ is the unknown, $x \in \Omega \subset \mathbb{R}^n$, $t \geqslant 0$.

– The wave equation

$$\frac{\partial^2 u}{\partial^2 t} - \Delta u = f$$

where $u(x,t)$ is the unknown, $x \in \Omega \subset \mathbb{R}^n$, $t \geqslant 0$ and f is given.

In this manuscript, we are interested on hyperbolic balance equations. Hyperbolic balance laws appear in the description of many physical processes. Indeed considering a domain $\Omega \in \mathbb{R}^n$ and a quantity of interest $\mathbf{U}$: a pressure, a concentration, or a density for example, we can describe the evolution of $\mathbf{U}$ in time by: *The temporal rate of change of* $\mathbf{U}$ *in any fixed sub-domain* $\omega \subset \Omega$ *is equal to the total amount of* $\mathbf{U}$ *produced or destroyed inside* ω *and the flux of* $\mathbf{U}$ *across the boundary* $\partial\omega$ [113]. In other word, it can be defined mathematically by eq. (2.1)

$$\frac{d}{dt}\int_w \mathbf{U}dx = -\underbrace{\int_{\partial\omega} f\cdot\mathbf{n}d\sigma(x)}_{\text{flux}} + \underbrace{\int_w \mathbf{S}dx}_{\text{source}} \tag{2.1}$$

where $\mathbf{n}$ is the unit outward normal, f the flux and $\mathbf{S}$ the source term. To obtain the hyperbolic equation define above, we use the integration by part

$$eq.\ (2.1) \quad \Leftrightarrow \quad \frac{d}{dt}\int_w \mathbf{U}dx + \int_\omega \nabla\cdot f\cdot\nu dx = \int_w \mathbf{S}dx \tag{2.2}$$

then, using an infinitesimal ω, we obtain the *balanced law*

$$\mathbf{U}_t + \nabla\cdot f = \mathbf{S} \qquad \forall(x,t)\in(\Omega,\mathbb{R}_+) \tag{2.3}$$

The system 2.3 is referred as *conservation laws* when $\mathbf{S} = 0$, i.e. when the change of $\mathbf{U}$ comes only from the quantity entering and leaving the domain of interest. For example, the scalar transport equation can be defined as follow: $\mathbf{U} = U \in \mathbb{R}$ a density of car in a road traffic, $\mathbf{a}(x,t) \in \mathbb{R}^2$ a velocity field at all points on the road. The flux in this case is $f = \mathbf{a}U$. And so, the corresponding conservation law 2.3 takes the form

$$U_t + \nabla\cdot(\mathbf{a}(x,t)U) = 0 \tag{2.4}$$

Considering a space of dimension d, $f = (F_1,\cdots,F_d)$, the hyperbolicity definition come from the interpretation of the Jacobian of the flux $J_d(U)$ defined as

$$J_d(\mathbf{U}) := \partial_{\mathbf{U}} f_d(\mathbf{U}) = \left(\frac{\partial F_{di}}{\partial U_j}(\mathbf{U})\right)_{i,j=1,\cdots,D}, \quad \forall d = 1,\cdots,D. \tag{2.5}$$

The system is called hyperbolic if for any $\mathbf{U}$ in the state space and any $\mathbf{w} = (\omega_1, \cdots, \omega_D) \in \mathbb{R}^d$, the matrix $J(\mathbf{U}, \mathbf{w})$ defined by

$$J(\mathbf{U}, \mathbf{w}) := \sum_{d=1}^{D} \omega_d J_d(\mathbf{U}) \tag{2.6}$$

has S real eigenvalues and S corresponding linearly independent eigenvectors. If the eigenvalues are all distinct, we call the system strictly hyperbolic.

2.2 Hyperbolic systems

In this section, we briefly introduce some hyperbolic systems.

2.2.1 Linear and non linear advection equations

The classical linear advection equation reads

$$u_t + \mathbf{a} \cdot \nabla u = 0, \quad u \in \mathbb{R}, \mathbf{a} \in \mathbb{R}^2 \tag{2.7}$$

with u the solution which correspond to an entity (a density for example). To understand properties of this equation, we consider the mono-dimensional Cauchy problem: find $u : \mathbb{R}^+ \times \mathbb{R} \longrightarrow \mathbb{R}$ such as

$$\begin{cases} \partial_t u(t,x) + a\partial_x u(t,x) = 0 & \text{with } a \text{ a velocity field} \quad \in \mathbb{R} \\ u(0,x) = u_0(x) & \text{and } u_0 \text{ the initial data} \end{cases} \tag{2.8}$$

Depending on the definition of a, the solution of the problem can take different form. If a is constant, it exists a unique solution which is $u_{ex}(t,x) = u_0(x - at)$. We call *Characteristic curves* of $\partial_t u(t,x) + a\partial_x u(t,x) = 0$ solutions of

$$\begin{cases} X'(t) &= a \\ X(0) &= x_0 \end{cases} \tag{2.9}$$

i.e. if a is constant, characteristic curves are describe by $X(t) = x_0 + at$.
NB: it is easy to prove that the solution u_{ex} is constant along any characteristic curve (see fig. 2.1).

In the linear advection case, characteristic curves are parallel lines of slope a. We now look at the non-linear conservation law:

$$\begin{cases} \partial_t u(t,x) + \partial_x f(u(t,x)) = 0 & \text{with } f = a(u) \text{ the numerical flux } \in \mathbb{R} \\ u(0,x) = u_0(x) & \text{and } u_0 \text{ the initial data} \end{cases}$$

Equivalently, $a(u) = J(u)$ in eq. (2.6), and we can mention two different types of fluxes: if $a(u) \nearrow$ when $u \nearrow$ (i.e. the flux f is convex), the characteristics form a bundle of converging lines and we call this effect a *shock*, and then if $a(u) \searrow$ if $u \nearrow$ (i.e. the flux f is concave), the characteristics form a bundle of divergent lines and we call this effect a *rarefaction*. These two different fluxes are reported in fig. 2.2.

To define a solution of this problem, we have to solve the *Riemann* problem, by using the

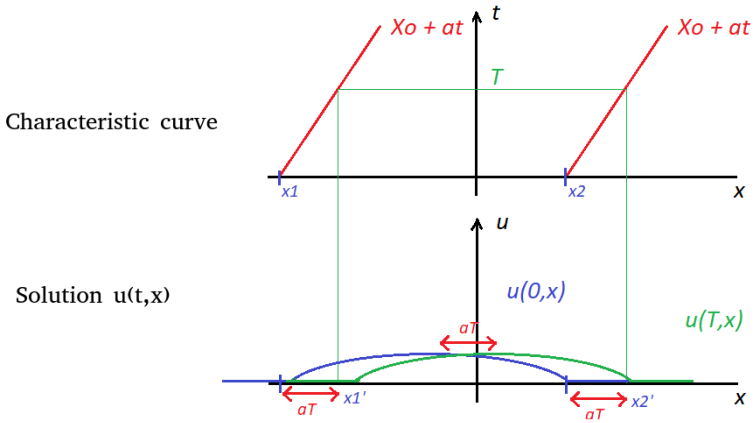


FIGURE 2.1: Solution u at time $t = 0$ in blue and $t = T$ in green at representative characteristic curve at $X(0) = x_1$ and $X(0) = x_2$.

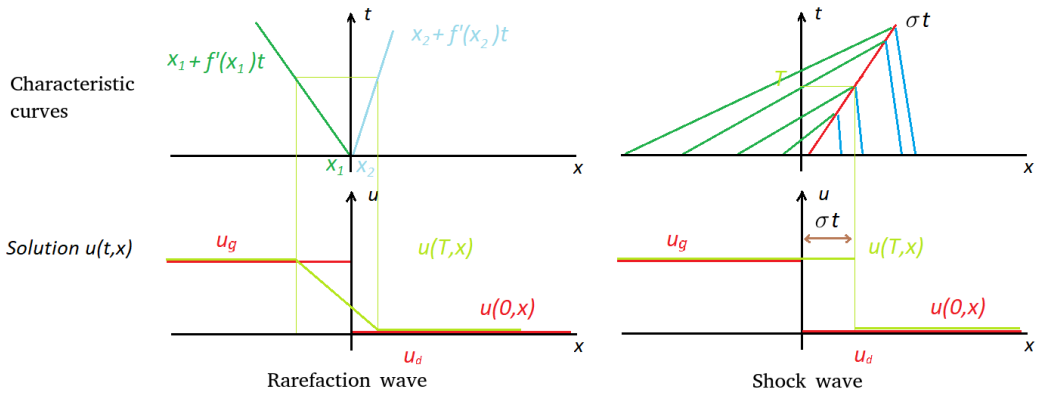


FIGURE 2.2: Rarefaction wave at left and shock at right.

Rankine-Hugoniot condition [124, 79]. This condition respects the property of conservation.

2.2.2 Burgers' equations

We briefly introduce another example of nonlinear transport equation called Burger's equation:

$$u_t + \mathrm{div}(f(u)) = 0 \qquad \forall (x,t) \in (\Omega, \mathbb{R}_+) \tag{2.10}$$

where $f(u)$ is a nonlinear function of u such as $f''(u) > 0 \, \forall u$ i.e. f is a convex function, or $f''(u) < 0 \, \forall u$ i.e. f is a concave function.
A example and probably the most use *Burgers'* equation that we will use in our numerical tests is

$$u_t + \nabla \cdot \frac{u^2}{2} = 0 \tag{2.11}$$

The Buger's equation 2.11 can be written in the same form as the advection velocity with $a = a(u) = u$. The corresponding characteristics are

$$x'(t) = u(x(t), t). \tag{2.12}$$

For convex initial data, even smooth, a shock appears since characteristics cross (see fig. 2.2) in a given time. It can be shown that the shock appears at $T_s = \frac{-1}{min u_0'(x)}$. We can note that

for concave initial data, a rarefaction phenomenon appears.

2.2.3 Euler's equations

The Euler equations are a system of conservation laws governing the dynamics of a compressible fluid. They can be written as follows :

- Let ρ be the density of a fluid (in $kg \cdot m^{-3}$). The conservation of mass is reflected locally by the so-called continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \tag{2.13}$$

  Where $\vec{v}$ denotes the Eulerian velocity of a fluid particle ($m \cdot s^{-1}$).

- The conservation of momentum is described by:

$$\frac{\partial (\rho \vec{v})}{\partial t} + \nabla \cdot (\rho \vec{v} \otimes \vec{v}) = -\nabla p + F \tag{2.14}$$

  Where p denotes the hydrostatic pressure (Pa) and F the external forces exerted in the fluid.

- And the conservation of total energy is characterized by

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho E \vec{v}) = -\nabla \cdot (p \vec{v}) + F \cdot \vec{v} \tag{2.15}$$

  Where E denotes the total energy per unit mass ($J \cdot kg^{-1}$). It is expressed as a function of the internal energy per unit of mass e: $E = e + \frac{1}{2}v$. Moreover E and p the pressure are coupled by $p = (\gamma - 1)(\rho E - \frac{1}{2}\rho v^2)$.

This system with three equations is called the Navier-Stokes system. It can be also read as

$$\begin{pmatrix} \rho \\ \rho \vec{v} \\ \rho E \end{pmatrix}_t + \nabla \cdot \begin{pmatrix} \rho \vec{v} \\ \rho \vec{v} \otimes \vec{v} + p I_d \\ \rho E \vec{v} + \vec{v} p I_d \end{pmatrix} = \begin{pmatrix} 0 \\ F \\ F \cdot \vec{v} \end{pmatrix} \tag{2.16}$$

with I_d the $d \times d$ identity matrix.

However, these equations include the effects of fluid viscosity in F and the resulting flux function depends not only on the state variables but also on their gradients, so the equations are not of the form 2.3 and are not hyperbolic.
Euler's equation comes from Navier-Stokes equations considering viscous effects known (and equal to zero). They are used to defining gas dynamics. Euler's equations in the conservative form are given by

$$\begin{pmatrix} \rho \\ \rho \vec{v} \\ \rho E \end{pmatrix}_t + \nabla \cdot \begin{pmatrix} \rho \vec{v} \\ \rho \vec{v} \otimes \vec{v} + p I_d \\ \rho E \vec{v} + \vec{v} p I_d \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{2.17}$$

2.2.4 Shallow Water equations

In the case of tsunamis and for coastal hydrodynamic models, it is considered that the wavelength of the waves is very large compared to the depth ($\lambda \gg d$). Using this hypothesis, we can define Shallow Water equations also called Saint-Venant equations, which describe the propagation of waves. They can be derived from the incompressible Navier Stokes equations 2.16 (neglecting the vertical acceleration) (see [53] and references therein).
In Cartesian coordinates, neglecting the effects of bottom friction, the Shallow Water equations read

$$\begin{pmatrix} h \\ hu \\ hv \end{pmatrix}_t + \begin{pmatrix} hu \\ \rho hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix}_x + \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}_y = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{2.18}$$

where $v = (u, v)^T$ now represents a depth averaged horizontal flow. And where g is acceleration due to gravity and ρ is the fluid density. The first equation is derived from mass conservation eq. (2.13), the second two from momentum conservation eq. (2.14).

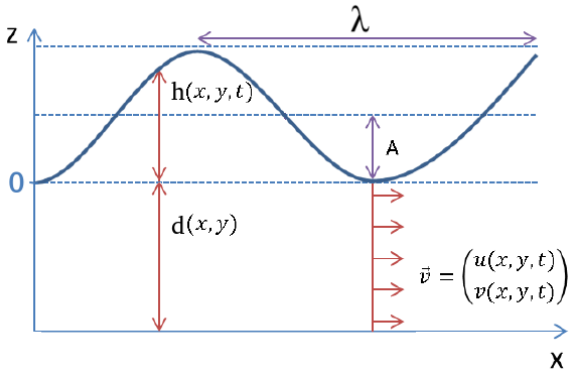


FIGURE 2.3: Definition of parameters.

2.3 A notion of stability and stabilization methods

2.3.1 Entropy condition

Following the approach of the entropy condition describe in [79, sec. 3.8], we define an entropy function $\eta(u)$. The idea of the entropy condition is that the entropy function $\eta(u)$ respects a conservation law for smooth solutions, and becomes an inequality for discontinuous solutions. To illustrate this condition, let us define $\eta(u)$, which satisfies the following conservation law:

$$\eta_t(u) + \nabla \cdot \phi(u) = 0 \qquad \forall (x,t) \in (\Omega, \mathbb{R}_+) \tag{2.19}$$

with $\phi(u)$ an entropy flux. By taking the one-dimensional cartesian space, for smooth solution u, the relation eq. (2.19) is equivalent to

$$\eta'(u)\partial_t u + \phi'(u)\partial_x u = 0 \qquad \forall (x,t) \in (\Omega, \mathbb{R}_+) \tag{2.20}$$

Considering now the initial transport equation $u_t + f(u)_x = 0 \Leftrightarrow u_t + f'(u)\partial_x u = 0$ and multiplying the relation by $\eta'(u)$, we obtain

$$\eta'(u)\partial_t u + \eta'(u)f'(u)\partial_x u = 0 \quad \Leftrightarrow \quad \phi'(u)\partial_x u = \eta'(u)f'(u)\partial_x u \tag{2.21}$$

$$\Rightarrow \quad \phi'(u) = \eta'(u)f'(u) \tag{2.22}$$

Now, the eq. (2.22) reads $\nabla\phi(u) = f'(u)\nabla\eta(u)$ (this system of m equations for the two variables η and ϕ may have no solution if $m > 2$). We also imposed the entropy function being convex, i.e. $\eta''(u) > 0$, for reasons that will be seen below.

As we said, the entropy $\eta(u)$ is conserved for smooth solutions. However, for discontinuous solutions, this property is not valid. In order to select the physically relevant solution, we introduce the so–called *vanishing viscosity* approximations. We explore the entropy behaviour for the vanishing viscosity weak solution. Considering a small $\epsilon > 0$, the viscous equation is

$$\partial_t u + \partial_x f(u) = \epsilon\partial_{xx} u \tag{2.23}$$

Multiplying eq. (2.23) by $\eta'(u)$, we obtain

$$\eta'(u)\partial_t u + \eta'(u)f'(u)\partial_x u = \epsilon\eta'(u)\partial_{xx} u \tag{2.24}$$

$$\Leftrightarrow \quad \eta_t(u) + \phi_x(u) = \epsilon\eta'(u)\partial_{xx} u \tag{2.25}$$

$$\Leftrightarrow \quad \eta_t(u) + \phi_x(u) = \epsilon\partial_x(\eta'(u)\partial_x u) - \epsilon\eta''(u)\partial_{xx} u^2 \tag{2.26}$$

To simplify, we denote by $\eta_t = \partial_t\eta$ and $u_x = \partial_x u$. Then, integrating this equation over $\Omega \times T = [x_1, x_2] \times [t_1, t_2]$ gives

$$\int_{t_1}^{t_2}\int_{x_1}^{x_2} \eta_t(u) + \phi_x(u)dxdt = \underbrace{\epsilon\int_{t_1}^{t_2}[(\eta'(u(x_2,t))u_x(x_2,t)) - (\eta'(u(x_1,t))u_x(x_1,t))]dt}_{\xrightarrow[\epsilon\to 0]{} 0} \tag{2.27}$$

$$\underbrace{-\int_{t_1}^{t_2}\int_{x_1}^{x_2}\epsilon\eta''(u)u_x^2 dxdt}_{\geq 0} \tag{2.28}$$

The last inequality is obtained knowing that $\epsilon > 0$, $u_x^2 > 0$ and $\eta'' > 0$. We finally obtain the following inequality: the vanishing viscosity weak solution satisfies

$$\int_{t_1}^{t_2}\int_{x_1}^{x_2} \eta_t(u) + \phi_x(u)dxdt \leq 0 \tag{2.29}$$

Consequently, the total integral of η is not necessarily conserved, but can only decrease.
In gas dynamics, there is a physical quantity called entropy, which is known to be constant along particle paths in a reversible process. However, an irreversible process increases the entropy of the system and so do not satisfy the conservation of the entropy.

Theorem 2.3.1 (The entropy condition) *The function $u(x,t)$ is the entropy solution of eq. (2.3) if, for all convex entropy functions and corresponding entropy fluxes, the inequality*

$$\eta(u)_t + \phi(u)_x \leq 0 \tag{2.30}$$

is satisfied in the weak sense.

This property is a necessary condition of stability of numerical methods. As an example, for the linear advection equation 2.7, a possible entropy pair (η, ϕ) is given by

$$\begin{cases} \eta(u) &= u^2/2 \\ \phi(u) &= \mathbf{a}u^2/2 \end{cases} \tag{2.31}$$

And $\eta(u) = u^2/2$ correspond to the kinetic energy of the system.

2.3.2 Numerical stabilization

Embedding stability in the discrete equations is very important in both smooth regions and in proximity of discontinuities. We can find several approaches in the literature in different settings as finite differences, finite volumes and discontinuous Galerkin methods which are based on the resolution of the *Riemann* problem. They are historically developed from one-dimensional flows (see [12, 52, 57, 97] and references therein). Although some examples of multidimensional formulations exist [99, 97, 98, 130, 44, 63, 96].

Other examples existing in literature involve stabilized finite element techniques (see [59, 68, 66, 64, 19, 20, 21] and references therein) and residual distribution schemes (see [15, 84, 110, 43, 3, 4] and references therein). In this manuscript, we mainly follow the continuous stabilized finite element approach. In particular, in the next section 3.2, we introduce different stabilization terms, that we then explore and compare by some numerical analysis and tests. As an example, in figure 2.4, we model a lake at rest with a non-constant topography (in brown). The water elevation is represented with the scale on the right, and the time of the simulation is $t_f = 1s$. We can observe that if we do not use a stabilized technique, we obtain a smeared water elevation. Some oscillations appear during the simulation.

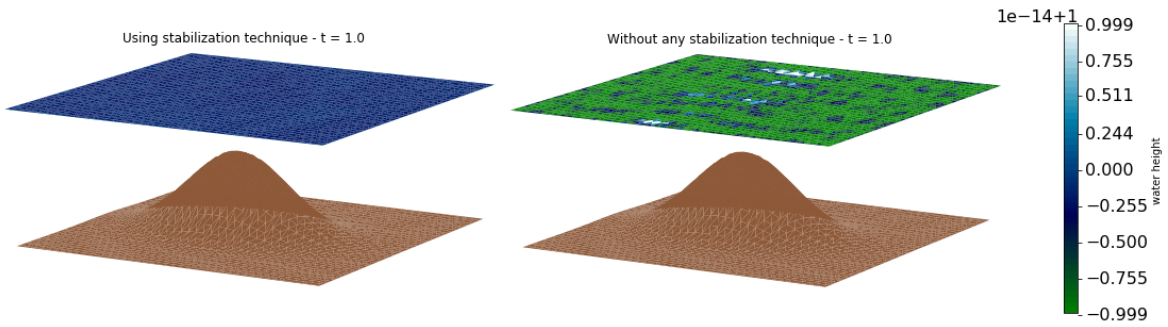


FIGURE 2.4: Lake at rest simulation [109]. Representation of the water elevation at $t_f = 1s$. At left: using stabilization technique, at right: without stabilization technique.

Another issue is the stabilization in correspondence of shocks. In this case too several approaches exist. In the context of continuous finite elements, this requires in general the

definition of a first order regularization allowing to provide an oscillation free solution. Example of approaches to achieve this are given in (see [58, 3, 72, 73] and references therein).

Chapter 3

Numerical methods

Chapter Abstract
In this chapter, we describe the *Finite Element Method* methods used in this manuscript to solve hyperbolic systems. We firstly introduce the Galerkin method in section 3.1. Then, stabilization techniques in section 3.2, section 3.3 and section 3.4 used to dump instabilities and smooth discontinuities. And finally, we introduce different spatial discretizations in section 3.5 and time integration techniques in section 3.6 which have all their advantages and disadvantages.

Outline

3.1 The Finite Element Methods 20
3.1.1 The Continuous Galerkin approach 20
3.1.2 The Discontinuous Galerkin approach 21
3.1.3 Approximation error 22
3.2 Stabilization techniques 23
3.2.1 Streamline-Upwind/Petrov-Galerkin - SUPG 23
3.2.2 Continuous Interior Penalty - CIP 25
3.2.3 Orthogonal Subscale Stabilization - OSS 26
3.3 Shock capturing technique 27
3.4 Numerical Fluxes 30
3.5 Finite Element Spaces and Quadrature Rules 30
3.5.1 The one-dimensional finite element spaces 30
3.5.2 The two-dimensional finite element spaces 32
3.5.3 Basic Lagrangian equispaced elements 32
3.5.4 Bernstein polynomials 33
3.5.5 Cubature elements 34
3.6 Time integration 35
3.6.1 Explicit Runge–Kutta and Strong Stability Preserving Runge–Kutta schemes 36
3.6.2 The *Deferred Correction* scheme 36

3.1 The Finite Element Methods

We are interested in the approximation of solutions of the following conservation law

$$\partial_t u(x,t) + \nabla \cdot f(u(x,t)) = 0 \quad x \in \Omega \subset \mathbb{R},\, t \in \mathbb{R}^+, \tag{3.1}$$

on a domain Ω of finite dimension. The resulting finite element space is built with elementary structures K_e such as triangles, rectangles or polygons. The finite element method or Galerkin method uses a piecewise continuous polynomial approximation (continuous in each K_e). We denote by $\Omega_h = \bigcup_e K_e$, and we also introduce the set of internal element boundaries (cell faces in 2 and 3 dimensional domains, cell nodes in 1 dimensional one) by $\mathcal{F}_h$. h is the characteristic mesh size of Ω_h, as for example the largest element diameter.

3.1.1 The Continuous Galerkin approach

From here on, we will simplify the notation, focusing on the scalar equation case $D = 1$, but the description can be easily generalized. When necessary, we will explicitely recall some details related to this generalization.
We introduce $H^1(\Omega)$ the Hilbert space defined by

$$H^1(\Omega) := \{u \in \mathbb{L}_2(\Omega); \exists v \in \mathbb{L}_2(\Omega) \text{ such as } \forall x,y \in \Omega, u(y) = u(x) + \int_x^y v(t)dt\} \tag{3.2}$$

Then $H_0^1(\Omega)$ a closed subset of $H^1(\Omega)$. Considering $\mathcal{V} = H_0^1(\Omega)$, the discrete solution is sought in a continuous finite element space $V_h^p = \{v_h \in \mathcal{C}^0(\overline{\Omega_h}) : \quad v_h|_K \in \mathbb{P}_p(K) \quad \forall K \in \Omega_h\}$ a subset of $\mathcal{V}$ of finite dimension and made of linear functions. Considering a finite element $K \in \Omega_h$, we introduce the notion of *degree of freedom* (DOF) (see fig. 3.1) which are nodes in elements. By abuse of notation, we write $(j)_{j \in K}$ the set of DOF in K.

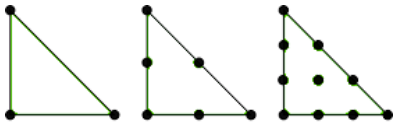


FIGURE 3.1: Finite elements with 3 DOF at left, 6 DOF on the middle and 10 at right.

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 $V_h^p = \text{span}\left\{\varphi_j\right\}_{j \in \Omega_h}$ and we can write $u_h(x) = \sum_{j \in \Omega_h} u_j \varphi_j(x)$, where, with an abuse of notation, with $j \in \Omega_h$ we mean the set of degrees of freedom with support in Ω_h. With a similar meaning, we will also use the notation $j \in K$ to mean the degrees of freedom with support on the cell K.

The variational formulation of the unstabilized approximation of eq. (3.1) reads: find $u_h \in V_h^p$ such that for any $v_h \in W_h \subset \mathbb{L}_2(\Omega_h) := \{v : \Omega_h \to \mathbb{R} : \int_{\Omega_h} |v|^2 < \infty\}$. The choice of W_h will be based on V_h, but it might take different forms for different stabilizations.

$$\int_\Omega v_h \partial_t u_h dx - \int_\Omega \partial_x v_h f(u_h)\, dx + [v_h f(u_h)]_{\partial\Omega} = 0. \tag{3.3}$$

As already said, we will consider several stabilized variants of eq. (3.3) which can be all written in the generic form: find $u_h \in V_h^p$ that satisfies

$$\int_\Omega v_h(\partial_t u_h + \partial_x f(u_h))dx + S(v_h, u_h) = 0, \quad \forall v_h \in V_h^p \tag{3.4}$$

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 details in the following section 3.2.
The two-dimensional unstabilized CG variational formulation reads:

$$\int_{\Omega_h} v_h \partial_t u_h dx - \int_{\Omega_h} \nabla \cdot v_h f(u_h)\ dx + \int_{\partial\Omega_h} v_h f(u_h) \cdot \mathbf{n} d\Gamma = 0, \tag{3.5}$$

where $\mathbf{n}$ is the normal to the boundary facing outward the domain. This formulation can be also re-integrated by parts, and using the continuity of the solution, we obtain

$$\int_{\Omega_h} v_h \partial_t u_h dx + \int_{\Omega_h} v_h \nabla \cdot f(u_h)\ dx + \int_{\partial\Omega_h} v_h (g_n - f(u_h) \cdot \mathbf{n}) d\Gamma = 0, \tag{3.6}$$

with g_n is the boundary flux.

NB: the general space of basis functions used in the finite element method is the set of Lagrange polynomial functions. The particularity of these functions is that considering the set degree of freedom (DOF) T_K of K, and $(\varphi_i)_{i \in T_K}$ the set of basis function relative to the DOF i,

$$\forall x_j \in T_K, \quad \varphi_i(x_j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{else.} \end{cases} \tag{3.7}$$

A complete definition of this set will be given later.

3.1.2 The Discontinuous Galerkin approach

The Discontinuous Galerkin (DG) method comes from the Finite Element Method considering that the solution u is piecewise continuous in $K \in \Omega_h$ but discontinuous over Ω. As it is well known, DG method gives very satisfying results to solve hyperbolic equations. The solution of system using DG methods will be useful in order to compare our results with the stabilized CG methods. This will be mainly used in the chapter 7 for complex and realistic cases.

The discrete solution is now sought in a discontinuous finite element space $V_h^p = \{v_h \in \mathcal{C}^0(\overline{\Omega_h}) : v_h|_K \in \mathbb{P}_p(K), \forall K \in \Omega_h\}$. The solution u is piecewise continuous in K. $V_h^p = \{v_h \in \mathcal{C}^0(\overline{K}) : v_h|_K \in \mathbb{P}_p(K), \forall K \in \Omega_h\}$. The DG approximation reads: find $u_h \in V_h^p$ such that for any $v_h \in W_h \subset \mathbb{L}_2(\Omega_h) := \{v : \Omega_h \to \mathbb{R} : \int_{\Omega_h} |v|^2 < \infty\}$

$$\sum_{K \in \Omega_h} \int_K v_h \partial_t u_h dx - \int_K \nabla \cdot v_h f(u_h)\ dx + \int_{\partial K} v_h f^*(u_h^+, u_h^-) \cdot \mathbf{n} d\Gamma = 0, \tag{3.8}$$

where $\mathbf{n}$ is the normal to the boundary facing outward of the domain and $f^*(u_h^+, u_h^-)$ the numerical flux through element's edges. An example of formulation for Euler equations is

given in [60, sec 6.6]. A CFL condition is also given in [60, sec 4.8, 4.7]. For our study, we use by default CFL $= 1/(2p+1)$, with p the degree of basis function space.
Numerical fluxes used for our studies are developed in section 3.4.

In practice, the DG method is used due to the locality of its approximation, which translates in a mass matrix which has a simple block diagonal structure. However, due to the discontinuity of the solution, the DG method need a high number of DOF over the domain and the system to solve require a high time of computation because of its size. For this reason, we decide to reduce the number of DOF by using Continuous Galerkin (CG) method (see fig. 3.2), add a stabilization term to treat/smooth discontinuities 3.2.

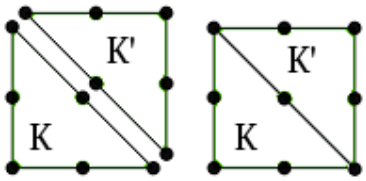


FIGURE 3.2: Discontinuous Galerkin configuration at left - Continuous Galerkin confirguration at right.

3.1.3 Approximation error

In a general framework, we study the error $\| u_{ex} - u_h \|$ in $H^1(\Omega)$, where u_{ex} is the exact solution of eq. (3.1) and u_h the approximated solution of eq. (3.5). As before, Ω_h denotes a tessellation of non-overlapping cells of Ω. We write the approximation error as:

$$\forall x \in K, \quad \| u_{ex}(x) - u_h(x) \| . \tag{3.9}$$

We control the approximation error by the interpolation error with $x \in K$

$$\forall x \in K, \quad \| u_{ex}(x) - u_h(x) \| \leq \| u_{ex}(x) - \sum_{i \in T_K} \varphi_i(x) u_i \| \tag{3.10}$$

where T_K denotes the set of degree of freedoms in K, $(\varphi_i)_{i \in T_K}$ the set of basis function of $\mathbb{P}_p(K)$ and u_i the approximated solution evaluated at the DOF i.

Then, knowing that $\mathbb{P}_p(K)$ is composed by Lagrange polynomial basis functions 3.7, we can use the error estimation of Lagrange interpolation, we obtain the bound

$$\forall x \in K, \quad \| u_{ex}(x) - u_h(x) \|_{\mathbb{L}_2} \leq h^{p+1} C, \qquad C \text{ a constant.} \tag{3.11}$$

The proof of this inequality can be easely found in [48] for example. In conclusion, using polynomial basis function in $\mathbb{P}_p(K)$ leads to an error approximation maximizes by $h_{max}^{p+1} C$, $C \in \mathbb{R}_+$.

3.2 Stabilization techniques

3.2.1 Streamline-Upwind/Petrov-Galerkin - SUPG

The SUPG method was introduced in [65] (see also [67, 21] 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 (3.3) with a test function belonging to the space

$$W_h := \{w_h : \quad w_h = v_h + \tau_K \nabla_u f(u_h) \cdot \nabla v_h; \quad v_h \in V_h^p\}, \tag{3.12}$$

or equivalently, using the notation eq. (2.6)

$$W_h := \{w_h : \quad w_h = v_h + \tau_K J(u_h, \nabla v_h); \quad v_h \in V_h^p\}, \tag{3.13}$$

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

$$\tau_K = \delta h_K (J_K)^{-1} \tag{3.14}$$

where h_K is the cell diameter, J_K represents a reference norm of the flux Jacobian norm on the element K and δ denotes a tunable stabilization parameter constant in space and time. E.g. δ is chosen as $1/2$ in [21]. In the scalar case, $J_K = ||\nabla_u f(u)||_K$.

The final stabilized variational formulation of eq. (3.4) reads

$$\int_\Omega v_h \partial_t u_h \, dx + \int_\Omega v_h \nabla \cdot f(u_h) \, dx + \underbrace{\sum_{K\in\Omega} \int_K \left(\nabla_u f(u_h) \cdot \nabla v_h\right) \tau_K \left(\partial_t u_h + \nabla \cdot f(u_h)\right) \, dx}_{S(v_h, u_h)} = 0. \tag{3.15}$$

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

To characterize the accuracy of the method, we can use the consistency analysis discussed e.g. in [3, sec. 3.1.1, sec. 3.2]. In particular, with a finite element polynomial approximation of degree p we can easily show that given a smooth exact solution $u^e(t,x)$, replacing formally u_h by the projection of u^e on the finite element space, we can write

$$\begin{aligned}\epsilon(\psi_h) := \Big| & \int_\Omega \psi_h \partial_t (u_h^e - u^e) \, dx - \int_\Omega \nabla \psi_h \cdot (\nabla f(u_h^e) - \nabla f(u^e)) \, dx \\ & + \sum_{K\in\Omega} \sum_{l,m\in K} \frac{\psi_l - \psi_m}{k+1} \int_K (\nabla_u f(u_h) \cdot \nabla \varphi_i) \tau_K \left(\partial_t (u_h^e - u^e) + \nabla \cdot (f(u_h^e) - f(u^e))\right) \, dx \Big| \\ & \leq C h^{p+1},\end{aligned} \tag{3.16}$$

with C a constant independent of h, for all functions ψ of class at least $\mathcal{C}^1(\overline{\Omega})$, 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 eq. (3.1), and for exact integration with $\tau_K = \tau$, a classical result is obtained for homogeneous boundary conditions and in the time continuous case by testing with $v_h = u_h + \tau\,\partial_t u_h$ to obtain [21]

$$\int_{\Omega_h} \partial_t \left(\frac{u_h^2}{2} + \tau^2 \frac{(\mathbf{a}\cdot\nabla u_h)^2}{2} \right) + \int_{\Omega_h} \mathbf{a}\cdot\nabla \left(\frac{u_h^2}{2} + \tau^2 \frac{(\partial_t u_h)^2}{2} \right) = -\int_{\Omega_h} \tau(\partial_t u_h + \mathbf{a}\cdot\nabla u_h)^2. \tag{3.17}$$

For periodic, or homogeneous boundary conditions this easily shows that the norm $|||u|||^2 := \int_{\Omega_h} \frac{u_h^2}{2} + \tau^2 \frac{(\mathbf{a}\cdot\nabla u_h)^2}{2} dx$ is non-increasing. The interested reader can refer to [21] for the analysis of some (implicit) fully discrete schemes.

Note on the SUPG technique applied to non scalar system

The extension of the SUPG method to a non scalar problem is not obvious. Here we describe the method used for the numerical simulation.
We define the following non scalar system of dimension D:

$$\begin{cases} \partial_t U + \nabla\cdot\mathcal{F}(U) = \mathbf{S}(U) \\ \mathcal{F} = (F_1, F_2) \end{cases} \tag{3.18}$$

with $U \in \mathbb{R}^D$, $\mathcal{F}(U) \in \mathbb{R}^{2\times D}$ and $\mathbf{S}(U) \in \mathbb{R}^D$. Equation (3.18) can also be written in its quasi-linear form

$$\partial_t U + \nabla_U \mathcal{F}(U)\cdot\nabla U = \mathbf{S}(U), \tag{3.19}$$

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

Following the definition of the SUPG method and [111, sec. 5] we define a positive definite stabilization matrix $\boldsymbol{\tau}_\mathbf{K} \in \mathbb{R}^{D\times D}$constant for every element. Although other definitions are possible, here we will evaluate this parameter as in [111]

$$\boldsymbol{\tau}_\mathbf{K} = \delta h_K \left(\sum_{j\in S_K} \left| \nabla_U \mathcal{F}(\bar{U}_K)\cdot n_j \right| \right)^{-1}, \tag{3.20}$$

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

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

$$\int_\Omega v_h \left(\partial_t U_h + \nabla\cdot\mathcal{F}(U_h) - \mathbf{S}(U_h)\right)_i +$$

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

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

3.2.2 Continuous Interior Penalty - CIP

An alternative, which maintains the sparse symmetric structure of the Galerkin matrix, is the continuous interior penalty (CIP) stabilization used in [25, 27, 23]. This method has been develop by E. Burman and P. Hansbo in [24], but it can be seen as a variation of the method originally proposed by Douglas and Dupont [46].

The method stabilizes the Galerkin formulation by adding a least-square term proportional to the jump of the gradient of the derivatives of the solution across the cell interfaces. The CIP introduces high order viscosity to the formulation, allowing the solution to tend to the vanishing viscosity limit. This term is computed on the solution at the previous time step, so it does not affect the structure of the LHS matrix.

The method reads

$$\int_{\Omega_h} v_h\partial_t u_h\,dx+\int_{\Omega_h} v_h\nabla\cdot f(u_h)\,dx+\underbrace{\sum_{\mathsf{f}\in\mathcal{F}_h}\int_{\mathsf{f}}\tau_{\mathsf{f}}[n_f\cdot\nabla v_h]\cdot[n_f\cdot\nabla u_h]\,d\Gamma}_{S(v_h,u_h)}=0, \quad (3.21)$$

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

$$\tau_{\mathsf{f}}=\delta\,h_{\mathsf{f}}^2\|\nabla_u f\|_{\mathsf{f}} \quad (3.22)$$

where $\|\nabla_u f\|_{\mathsf{f}}$ a reference value of the norm of the flux Jacobian on f, h_{f} a characteristic size of the mesh neighboring f and δ denotes a tunable stabilization parameter >0 [28], and constant in space and time. E.g. δ is chosen as 1 in [27].

As stated above, a clear advantage of CIP is that it does not modify the mass matrix, allowing to obtain efficient schemes if a mass lumping strategy can be devised. On the other side, the stencil of the scheme increases as the jump of a degree of freedom interacts with cells which are not next to the degree of freedom itself (up to 2 cells distance). Note that for higher order approximations [26, 76] suggest the use of jumps in higher derivatives to improve the stability of the method. However, here we consider the jump in the first derivatives in order to be able to apply the stability analysis and to study the influence of δ on the stability of the method. Some results might be definitely improved adding these stabilizations on higher derivatives.

The accuracy of CIP can be assessed with a consistency analysis as discussed in [3, sec. 3.1.1, sec. 3.2]. This consists in, formally substituting u_h by the projection onto the finite element polynomial of degree p space of u^e, a given smooth exact solution $u^e(t,x)$, we can show that for all functions ψ of class at least $\mathcal{C}^1(\Omega)$, of which ψ_h denotes the finite

element projection, we have the truncation error estimate

$$\epsilon(\psi_h) := \Big| \int_\Omega \psi_h \partial_t (u_h^e - u^e)\, dx - \int_\Omega \nabla\psi_h \cdot (f(u_h^e) - f(u^e))\, dx + \sum_{\mathsf{f}\in\mathcal{F}_h} \int_{\mathsf{f}} \tau_{\mathsf{f}} [n_f \cdot \nabla v_h] \cdot [n_f \cdot \nabla (u_h^e - u^e)] \Big| \leq C h^{p+1}, \tag{3.23}$$

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

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

$$\int_{\Omega_h} \partial_t \frac{u_h^2}{2} = -\sum_{\mathsf{f}\in\mathcal{F}_h} \int_{\mathsf{f}} \tau_{\mathsf{f}} [n_f \cdot \nabla u_h]^2, \tag{3.24}$$

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

Note that for higher than second order it may be relevant to consider additional penalty terms based on higher derivatives (see e.g. [26, 22, 107]). We did not do this in this work.

3.2.3 Orthogonal Subscale Stabilization - OSS

Another symmetric stabilization approach is the Orthogonal Subscale Stabilization (OSS) method. Originally introduced as Pressure Gradient Projection (PGP) in [38] for Stokes equations, it was extended to the OSS method in [37, 8] for different problems with numerical instabilities, such as convection–diffusion–reaction problems. This stabilization penalizes the fluctuations of the gradient of the solution with a projection of the gradient onto the finite element space. The method applied to (3.3) reads: find $u_h \in V_h^p$ such that $\forall v_h \in V_h^p$

$$\begin{cases} \int_{\Omega_h} v_h \partial_t u_h\, dx + \int_{\Omega_h} v_h \nabla\cdot f(u_h)\, dx + \underbrace{\sum_{K\in\Omega_h} \int_K \tau_K \nabla v_h \cdot (\nabla u_h - w_h)\, dx}_{S(v_h, u_h)} = 0, \\ \int_{\Omega_h} v_h w_h\, dx - \int_{\Omega_h} v_h \nabla u_h\, dx = 0. \end{cases} \tag{3.25}$$

For this method, the stabilization parameter is evaluated as

$$\tau_K = \delta h_K \|\nabla_u f\|_K \tag{3.26}$$

where δ denotes a tunable stabilization parameter > 0 [37], and constant in space and time. The drawback of this method, with respect to CIP, is the requirement of a matrix inversion to project the gradient of the solution in the second equation of (3.25). This cost can be alleviated by the choice of elements and quadrature rules if they result in a diagonal mass matrix, as it will be the case for *Cubature* elements that we will describe below.

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

$$\epsilon(\psi_h) := \Big| \int_{\Omega_h} \psi_h \partial_t (u_h^e - u^e)\, dx - \int_{\Omega_h} \nabla\psi_h \cdot (f(u_h^e) - f(u^e))\, dx + \sum_{K\in\Omega_h} \tau_K \int_K \nabla\psi_h \cdot \nabla(u_h^e - u^e) + \sum_{K\in\Omega_h} \tau_K \int_K \nabla\psi_h \cdot (\nabla u^e - w_h^e) \Big| \leq Ch^{p+1}, \tag{3.27}$$

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

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

Finally, for a linear flux, periodic boundaries and taking $\tau_K = \tau$ constant along the mesh, we can test with $v_h = u_h$ in the first equation of (3.25), and with $v_h = \tau w_h$ in the second one and sum up the result to get

$$\int_{\Omega_h} \partial_t \frac{u_h^2}{2} = -\sum_K \int_K \tau_K (\nabla u_h - w_h)^2, \tag{3.28}$$

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

The truncation consistency error analysis presented above for the three stabilization terms is completely formal and it does not comprehend an entire classical error analysis. These estimations tell us that the stabilization terms that we introduced are of the wanted order of accuracy and that they are usable to aim at the prescribed order of accuracy. This type of analysis has been already done for multidimensional problems inter alia in [1]. More rigorous proof of error bounds with $h^{p+\frac{1}{2}}$ estimates can be found in [22] for the CIP. We did not consider in this work projection stabilizations involving higher derivatives.

3.3 Shock capturing technique

So far, we considered a continuous solution of our hyperbolic problem. However, in certain configurations, discontinuities can appear. In presence of discontinuities, it is necessary to add some diffusion in the scheme. An additional term must be added to smooth discontinuities and allows them to propagate. In particular, we introduce the entropy viscosity method developed in [101, 58] for linear and non-linear hyperbolic problems, which allows keeping a control region where entropy production is large (i.e. in presence of discontinuities). The main advantage of this technique is that it is very simple to implement, since it does not use slope limiters. Many other shock capturing techniques exist in the literature such in [121] also based on the entropy dissipation.

This supplementary term will be mainly used in the last chapter 6 and 7 for complex and realistic cases, where discontinuities can appear.

The stabilized variationnal formulation of 3.4 reads

$$\int_{\Omega_h} v_h \partial_t u_h \, dx + \int_{\Omega_h} v_h \nabla \cdot f(u_h) \, dx + \underbrace{\sum_K \mu_K(u_h) \int_K \nabla u_h \cdot \nabla v_h \, dx}_{S(v_h, u_h)} = 0, \tag{3.29}$$

with $\mu_K(u_h)$ the entropy viscosity parameter computed as follow: remembering the advection problem

$$u_t + \nabla \cdot f(u) = 0, \quad u \in \mathbb{R}, \mathbf{a} \in \mathbb{R}^2. \tag{3.30}$$

Considering the system eq. (3.30) admits a convex entropy function (see section 2.3.1) that satisfy the entropy inequality

$$\partial_t E + \nabla \cdot \mathbf{G} \leq 0, \tag{3.31}$$

The resulting energy dissipation can be defined by

$$-D = \partial_t E + \nabla \cdot \mathbf{G}. \tag{3.32}$$

In practice, D should be ≈ 0 in smooth region and ≥ 0 around discontinuity. It is a good criterion to detect shocks.
For each triangle K, we evaluate the cell-average value of the residual entropy $(r_E)_K^n$ at $t = t^n$ by

$$(r_E)_K^n = \frac{1}{|K|} \left(\int_K \partial_t E_h^n d\mathbf{x} + \int_K \nabla \cdot \mathbf{G}_h^n d\mathbf{x} \right) \tag{3.33}$$

$$= \frac{1}{|K|} \left(\int_K \partial_t E_h^n d\mathbf{x} + \int_{\partial K} \widehat{\mathbf{G}_h^n} \cdot \mathbf{n} dl \right), \tag{3.34}$$

with $\mathbf{n}$ the outward normal, and the numerical entropy flux $\widehat{\mathbf{G}^n}$ is evaluated on the triangle edge in the CG formulation and by a consistent flux in the DG case, although the use of the internal value is also possible. In practice, this choice has little impact.

We evaluate $\partial_t E^n$ by a $(p+1)^{th}$ order BDF time scheme (Backward Difference Formula), see appendix C for all BDF, $p = 1, 2, 3$. $(r_E)_K^n$ is computed once per time step and used for each intermediate RK stages.

We now introduce the artificial viscosity parameter μ_K following [101, 58]: we define a viscosity μ_E such that

$$\mu_E = \frac{c_E |(r_E)_K^n| h_K^2}{\Delta E} \tag{3.35}$$

where ΔE is the reference entropy, c_E a control parameter > 0. Then μ_{max} the viscosity upper bound based on the numerical flux

$$\mu_{max} = c_{max} h_K \max_K \left(|f'(u)| \right),$$

where c_{max} is a $\mathcal{O}(1)$ user defined parameter. Then the entropy viscosity parameter

$$\mu_K = min(\mu_{max}, \mu_E) \tag{3.36}$$

Note on the entropy viscosity technique applied to SW system in the Uhaina plaform This method, used in BRGM and INRIA code for Shallow Water system is described in [58, 101]. Remembering the Shallow Water system from section 2.2.4:

$$\begin{cases} \partial_t h + \partial_x(hu) + \partial_y(hv) & = 0 \\ \partial_t(hu) + \partial_x(hu^2 + g\frac{h^2}{2}) + \partial_y(huv) & = -gh(S_{ox} + S_{fx}) \\ \partial_t(hv) + \partial_x(huv) + \partial_y(hv^2 + g\frac{h^2}{2}) & = -gh(S_{oy} + S_{fy}) \end{cases} \tag{3.37}$$

The system eq. (3.37) admits a convex entropy function (see section 2.3.1) that satisfy the entropy inequality

$$\partial_t E + \nabla \cdot \mathbf{G} \leq 0, \tag{3.38}$$

$$\text{with} \quad E = \frac{1}{2}(h \parallel \mathbf{u} \parallel^2 + gh) + ghb, \; \mathbf{G} = \left(E + \frac{gh}{2}\right)\mathbf{u} \tag{3.39}$$

with $\mathbf{u} = (u, v)^T$.
NB: The entropy function E describe the total energy of the system and $\mathbf{G}$ the energy flux. Then, the artificial viscosity parameter μ_K is computed as follow: we define a viscosity μ_E such that

$$\mu_E = \frac{|(r_E)_K^n| h_K}{\tau \beta N} \tag{3.40}$$

where τ, β and N are control parameters > 0. Then μ_{max} the viscosity upper bound based

$$\mu_{max} = \alpha \kappa h_K \max_K \left(|f'(u)|\right),$$

where α and κ are $\mathcal{O}(1)$ user defined parameters. Then the entropy viscosity

$$\mu_K = \mu_{max} min(1, \mu_E) \tag{3.41}$$

Then, to compute the entropy flux, the maximum wave celerity of the SW system λ is given by:

$$\lambda = \max_{K \in \Omega} \max_{f \in \partial K} \left(|\mathbf{u}^n \cdot \mathbf{n}| + \sqrt{g h^n}\right). \tag{3.42}$$

Following [18], α is chosen from 0 and increased till removing all spurious oscillations, τ is chosen to be the best compromise to get stabilized shocks with a minimal imposed viscosity (in [18] they reduce the value until removing unwanted oscillations, $\tau = 1/10$), β is chosen from > 0 and is larger as the solution is smoother, i.e. we reduce the value of the viscosity term if the solution is smooth (ex: $\beta = 0.057$ for breaking waves), then μ should have a lower value for higher order methods, so it is shown in [18] that $\kappa = 1/p$ with p the degree of polynomial approximation gives satisfactory results and $N = g^{3/2} \bar{h}^{5/2}$. Finally

$$\mu_{max} = \alpha \kappa h_K \max_K \left(|\mathbf{u} \cdot \mathbf{n}| + \sqrt{gh}\right).$$

3.4 Numerical Fluxes

Numerous fluxes exist in the literature, the book of Professor Eleuterio F. Toro [124] contains most of them. A general interpretation of the numerical flux is given by:

$$f|_{K^+}(u_h)^* \cdot \mathbf{n}^+ = \frac{1}{2}\left(f|_{K^+}(u_h) + f|_{K^-}(u_h)\right) \cdot \mathbf{n}^+ + \frac{c}{2}\left(u_h|_{K^+} - u_h|_{K^+}\right), \tag{3.43}$$

with c a velocity field. A classical flux used in the coastal hydrodynamic context using a *discontinuous Galerkin* approach is the *Rusanov* flux which is the local interpretation of the *Lax-Friedrich* flux where $c_{LF} = \frac{C_{cfl}\Delta x}{\Delta t}$ with C_{cfl} the Courant number coefficient, usually chosen (empirically) ≈ 0.9. The Rusanov flux is defined by the maximum wave speed by elements. For Shallow Water equations, $c_R = max(Sp_1, Sp_2)$ with Sp_1 and Sp_2 defined in section 6.2.

3.5 Finite Element Spaces and Quadrature Rules

3.5.1 The one-dimensional finite element spaces

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 amount 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 $[x_i, x_{i+1}]$.

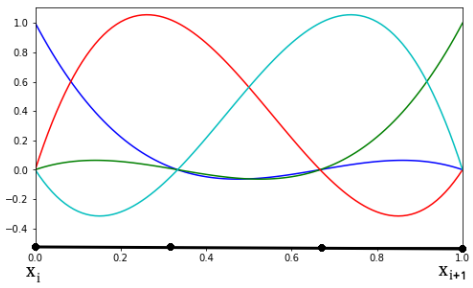


FIGURE 3.3: Lagrange polynomial basis functions defined on equidistant points, $p = 3$

2. Bernstein polynomials. Linearly mapping K onto $[0,1]$ they are defined for $j = 0,\dots,p$ by

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

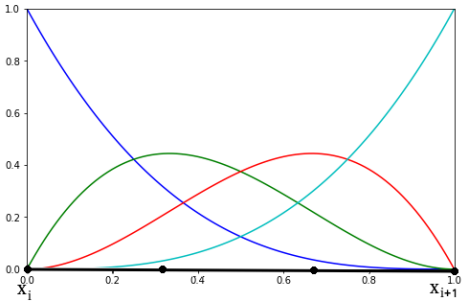


FIGURE 3.4: Bernstein polynomial basis functions defined on equidistant points, $p = 3$

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 of the involved matrices and thus to different stability properties due to the full discretization of the problem. Let us remark that the evaluation of integrals is done by Gaussian quadrature formulas, 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

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

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

Another important property that we need to effectively apply mass lumping [107] 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 formulas. 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 Gauss–Lobatto points and quadrature rule using the same points;
- *Bernstein* elements for Bernstein polynomials with Gauss–Legendre quadrature.

3.5.2 The two-dimensional finite element spaces

In this section we describe three finite element spaces on triangular grids that will be the object of the stability analysis. The first elements are locally defined by classical Lagrangian polynomials on equispaced points, then we will introduce the *Cubature* elements which are accompanied by carefully chosen quadrature rule, and finally the Bernstein polynomials. An example of an equispace repartition is given in fig. 3.1 for elements of degree $p = 1, 2$ and 3.

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

3.5.3 Basic Lagrangian equispaced elements

On triangles, we consider Lagrange polynomials with degrees at most p:

$$\mathbb{P}_p = \{ \sum_{\alpha+\beta \leq p} c_{\alpha,\beta} x^\alpha y^\beta \}$$

. We define the barycentric coordinates $\lambda_i(x, y)$ which are affine functions on $\mathbb{R}^2$ verifying the following relations

$$\lambda_i(v_j) = \delta_{ij}, \quad \forall i, j = 1, \dots, 3, \tag{3.44}$$

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

The polynomials so defined in a triangle form a partition of unity, but they have also negative values. This leads to negative or zero values of their integrals. This is problematic for some time discretization and we will see why. We will use these polynomials in combination with exact Gauss–Lobatto quadrature formulas for such polynomials and we will refer to them as *Basic* elements. An example of basis functions for $p = 3$ and $\alpha_3 = 0$ is shown in 3.5. In other words, basis functions corresponding to DOF $v_j = (x_j, y_j)$ when $y_j = 0$.

Algorithm 1 Lagrangian basis function in barycentric coordinates

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

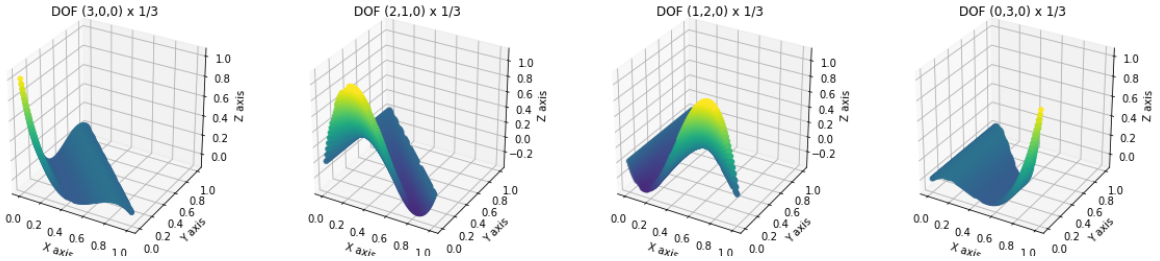


FIGURE 3.5: Basis functions of *Basic Lagrangian equispaced elements* for $p = 3$ and $\alpha_3 = 0$

3.5.4 Bernstein polynomials

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

$$\varphi_\alpha(x, y) = p! \prod_{i=1}^{3} \frac{\lambda_i^{\alpha_i}(x, y)}{\alpha_i!}. \tag{3.45}$$

Bernstein polynomials verify additional properties besides the one already cited for Lagrangian points. As before, they form a partition of unity, the basis functions are non-negative in any point of the triangle, and, hence, their integrals are positive. These properties lead also to the fact that the value at each point is a convex combination of the coefficients of the polynomials, hence, it is easy to bound minimum and maximum of the function by the minimum and maximum of the coefficients. This has been used in different techniques to preserve positivity of the solution [7, 75]. We will use these polynomials with corresponding high order accurate quadrature formulas. We will denote these elements with the symbol $\mathbb{B}_p$ and we refer to them as *Bernstein* elements. An example of basis functions for $p = 3$ and $\alpha_3 = 0$ is shown in 3.6. In other words, basis functions corresponding to DOF $v_j = (x_j, y_j)$ when $y_j = 0$.

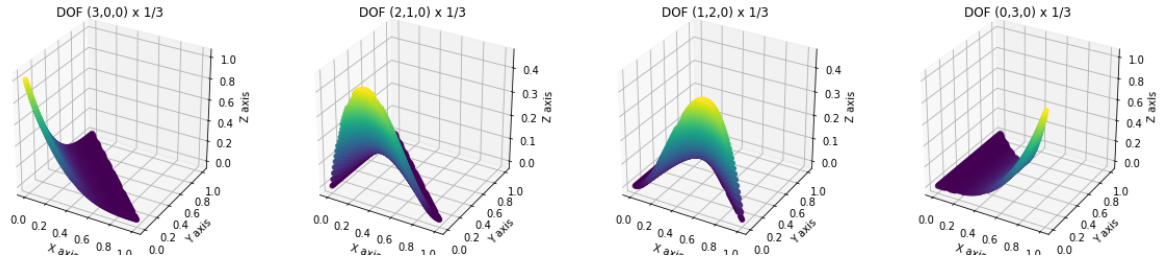


FIGURE 3.6: Basis functions of *Basic Bernstein equispaced elements* for $p = 3$ and $\alpha_3 = 0$

3.5.5 Cubature elements

Contrary to the work done in 1D [88], the extension of Legendre–Gauss–Lobatto points which minimize the interpolation error do not exist for the triangle. They have to be computed numerically such as *Fekete* points [70, 115, 122]. The problem of this approach is that it requires as classical finite elements the inversion of a sparse global mass matrix.
Cubature elements were introduced by G. Cohen and P. Joly in 2001 [39] for the wave equation (second order hyperbolic equation), and are an extension of Lagrange polynomials with the goal of optimizing the underlying quadrature formula error. We will denote with the symbol $\tilde{\mathbb{P}}_p$ and they will be contained in another larger space of *Lagrange* elements, i.e., $\mathbb{P}_p \subseteq \tilde{\mathbb{P}}_p \subseteq \mathbb{P}_{p'}$, with p' the smallest possible integer. Similar techniques have been used to minimize the interpolation error [70, 115, 122]. The objective of these polynomials is to use the points of the Lagrangian interpolation of the polynomials as quadrature points. This means that the obtained quadrature is $\int_K f(x,y) = \sum_\alpha \omega_\alpha f(x_\alpha, y_\alpha)$, where $\int_K \varphi_\alpha = \omega_\alpha$ and $\varphi_\alpha(x_\beta, y_\beta) = \delta_{\alpha\beta}$. This approach can be considered an extension of the Gauss–Lobatto quadrature in 1D for non Cartesian meshes. The biggest advantage of this approach is to obtain a diagonal mass matrix and, hence, saving a lot of computational time. The drawback is that one needs to increase the number of basis function inside one element to obtain an accurate enough quadrature rule. In our work, we propose to extend this approach to first order hyperbolic equations. A successful extension to elliptic problem is proposed in [102]. A comparison between the equispace repartition and the *Cubature* repartition for elements of degree $p = 3$ is shown in 3.7.

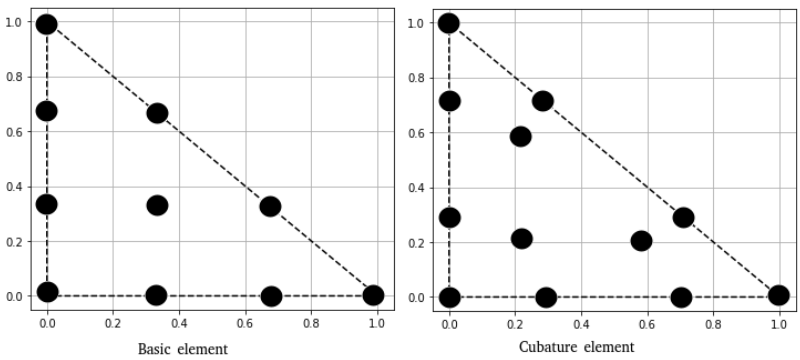


FIGURE 3.7: Comparison of the equispace repartition at left and the cubature repartition at right for elements of degree $p = 3$.

The challenges of this approach are the following:

- Obtain a quadrature which is highly accurate, at least $p + p' - 2$ order accurate [34];

- Obtain positive quadrature weights $\omega_\alpha > 0$ for stability reasons [123];
- Minimize the number of basis functions of $\tilde{\mathbb{P}}_p$;
- The set of quadrature points has to be $\tilde{\mathbb{P}}_p$-unisolvent;
- The number of quadrature points of edges has to be sufficient ensure the conformity of the finite element.

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

For $p = 1$ the *Cubature* elements do not differ from the *Basic* elements but in the quadrature formula. For $p = 2$ the *Cubature* elements introduce an other degree of freedom at the center of the triangle, leading to 7 quadrature points and basis functions per element. For $p = 3$ the additional degree of freedom in the triangle are 3, leading to 12 basis functions per triangle.

All the details of such elements can be found in [39, 55, 69] and appendix A. In the same appendix we also report all the details on the polynomial basis functions used here, not available in the published literature. We will use the symbol $\tilde{\mathbb{P}}_p$ and the name *Cubature* elements to refer to them.

Other elements such as *Fekete-Gauss* points [55] exist in the literature. They are optimized to interpolate and integrate with high accuracy. However, it is shown that they require more computing time to achieve similar results than *Cubature* points for high order of accuracy.

3.6 Time integration

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

$$\mathbb{M}\frac{dU}{dt} = \mathtt{r}(t) \tag{3.46}$$

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 correction iterations.

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

Runge–Kutta time integration methods are described by the following one step procedure

$$\begin{aligned} U^{(0)} &:= U^n, \\ U^{(s)} &:= U^n + \Delta t \sum_{j=0}^{s-1} \alpha_j^s \mathbb{M}^{-1} \mathtt{r}(U^{(j)}) \quad s = 1, \dots, S, \\ U^{n+1} &:= U^n + \Delta t \sum_{s=0}^{S} \beta_s \mathbb{M}^{-1} \mathtt{r}(U^{(s)}). \end{aligned} \tag{3.47}$$

Here, we use for the solution the superscript n to indicate the time step 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 B in table B.1.

A particular case is that of SSPRK methods introduced in [117]. They are essentially convex combinations of forward Euler steps, and can be rewritten as follows

$$\begin{aligned} U^{(0)} &:= U^n, \\ U^{(s)} &:= \sum_{j=0}^{s-1} \left(\gamma_j^s U^{(j)} + \Delta t \mu_j^s \mathbb{M}^{-1} \mathtt{r}(U^{(j)}) \right) \quad s = 1, \dots, S, \\ U^{n+1} &:= U^{(S)}, \end{aligned} \tag{3.48}$$

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 [117], the third order SSPRK(4,3) presented in [112, Page 189], and the fourth order SSPRK(5,4) defined in [112, Table 3]. For complete reproducibility of the results, we put all their Butcher' tableaux in appendix B in table B.3.

3.6.2 The *Deferred Correction* scheme

Deferred correction methods were originally introduced in [47] as explicit solvers of ODEs, but soon implicit [90] or positivity preserving [100] versions and extensions to PDE solvers [2] were studied. In [2, 104, 107] 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 [2], but also the *Cubature* elements can serve the purpose.

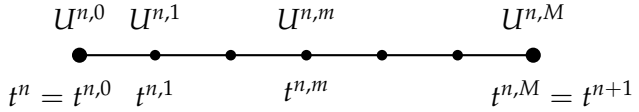


FIGURE 3.8: Subtimesteps inside the time step $[t^n, t^{n+1}]$

Consider a discretization of each timestep into M subtimesteps as in fig. 3.8. For each subtimestep the goal is to find the solution of the integral form of the semidiscretized ODE

eq. (3.46) as

$$\begin{aligned}\mathbb{M}\left(U^{n,m}-U^{n,0}\right)-\int_{t^{n,0}}^{t^{n,m}}\mathtt{r}(U(s))ds &\approx \text{Ł}^2(\underline{U})^m\\ :=\mathbb{M}\left(U^{n,m}-U^{n,0}\right)-\Delta t\sum_{z\in[\![0,M]\!]}\rho_z^m\mathtt{r}(U^{n,z}) &= 0,\end{aligned}\tag{3.49}$$

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 [2, 104, 107] for details). The algebraic system $\text{Ł}^2(\underline{U}^*)=0$ 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 low order easy–to–invert operator Ł^1. This operator is typically obtained using an explicit timestepping and a lumped mass matrix, i.e.,

$$\begin{aligned}\mathbb{M}\left(U^{n,m}-U^{n,0}\right)-\int_{t^{n,0}}^{t^{n,m}}\mathtt{r}(U(s))ds &\approx \text{Ł}^1(\underline{U})^m\\ :=\mathbb{D}\left(U^{n,m}-U^{n,0}\right)-\Delta t\beta^m\mathtt{r}(U^{n,0}) &= 0.\end{aligned}\tag{3.50}$$

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

$$U^{n,m,(0)}:=U^n \qquad m=0,\dots,M, \tag{3.51a}$$

$$U^{n,0,(k)}:=U^n \qquad k=0,\dots,K, \tag{3.51b}$$

$$\text{Ł}^1(\underline{U}^{(k)})=\text{Ł}^1(\underline{U}^{(k-1)})-\text{Ł}^2(\underline{U}^{(k-1)}) \qquad k=1,\dots,K, \tag{3.51c}$$

$$U^{n+1}:=U^{n,M,(K)}. \tag{3.51d}$$

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

Relying only on the inversion of the low order operator, the method 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 Ł^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 eq. (3.51c) as

$$U^{n,m,(k+1)}=U^{n,m,(k)}-\mathbb{D}^{-1}\mathbb{M}\left(U^{n,m,(k)}-U^{n,0,(k)}\right)+\sum_{j=0}^{M}\Delta t\rho_j^m\mathbb{D}^{-1}\mathtt{r}(U^{n,j,(k)}). \tag{3.52}$$

Comparing with eq. (3.48), 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 $m,r=0,\dots,M$ and $k=0,\dots,K-1$ and instead of the mass matrix, we use the

diagonal one.

Remark 3.6.1 (DeC with SUPG) *The iterative procedure of the DeC method allows even to overcome the difficulties that some implicit stabilization as the SUPG has. Indeed, the SUPG stabilization term can be added only to the* $\mathbb{L}^2$ *operator, keeping the high order accuracy of this operator. Since the* $\mathbb{L}^2$ *operator is applied to the previously computed iteration, all the terms of the SUPG, included the time derivative of* u *in eq.* (3.15)*, can be explicitly computed on* $U^{(k-1)}$*, keeping then the diagonal mass matrix for the whole scheme.*

Chapter 4

Analysis of the one-dimensional formulation

Chapter Abstract

We study continuous finite element discretizations for one dimensional hyperbolic partial differential equations. The main contribution of this chapter 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 orthogonal subscale stabilization (OSS). 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.It allows 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 OSS stabilization are the most promising combinations.

Outline

4.1 Introduction **40**

4.2 Basic spectral theory **41**

4.3 Fourier Analysis **42**

4.3.1 Preliminaries and time continuous analysis 43

4.3.2 Space-continuous analysis for *Runge-Kutta* schemes 47

4.3.3 Fully discrete analysis 49
4.4 Results of the fully discrete spectral analysis 51
4.4.1 Dispersion and damping 54
4.5 A note on nonlinear stability 58
4.6 Numerical Simulations 60
4.6.1 Linear advection equation 60
4.6.2 Application to Burgers' equation 63
4.6.3 Application to Shallow Water equations 66
4.7 Conclusion 67

4.1 Introduction

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

$$\partial_t u(x,t) + \partial_x f(u(x,t)) = 0 \quad x \in \Omega \subset \mathbb{R},\, t \in \mathbb{R}^+, \tag{4.1}$$

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

$$f(u(x,t)) = au(x,t)\,, \quad a = \text{const}\,. \tag{4.2}$$

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 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 [2]. There, to avoid the inversion of the 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 [82]. 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 [106, 105], the CG methods are not always $\mathbb{L}_2$–stable at the discrete level for hyperbolic problems and there is the need of additional stabilization terms. In particular, when periodic boundary conditions (BC) are applied to the problem, the instability shows larger effects. That is why many different stabilization techniques have been introduced for CG methods. These 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 $\mathbb{L}_2$–stable [108, 91]. The stabilization terms play an important role and we will compare three of them. The first is the streamline upwind Petrov–Galerkin (SUPG) stabilization [28, 21], 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 [25, 27, 23], 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 orthogonal subscale stabilization [37], 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 used in practice. The stability of such schemes will be practically computed thanks to a von Neumann analysis, which allows to determine whether the $\mathbb{L}_2$–norm of the approximated solution is bounded by the initial one. Other types of norms are sometimes more interesting as object of study, or simpler to be bound in other contexts, for example in an analytical one. We will focus on the $\mathbb{L}_2$ discrete norm stability in the majority of this work.

Numerical simulations 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 chapter is organized as follows. In section 4.2, we introduce some notions about the Spectral theory. sections 4.3 and 4.4 are dedicated to the Fourier stability analysis. In section 4.5 we provide some elements concerning the extension of the stabilization methods discussed to nonlinear problems, and finally in section 4.6 we show numerical results on linear and nonlinear problems. The paper is ended by a summary and overlook on future perspectives in section 4.7.

4.2 Basic spectral theory

In general, the numerical solution u_h does not satisfy the original PDE

$$\partial_t u + a\partial_x u = 0 \tag{4.3}$$

exactly, but only approximately, i.e. in general we have

$$\partial_t u_h + a\partial_x u_h \neq 0. \tag{4.4}$$

To quantify the errors, we will consider the so-called *modified equation* or *equivalent differential equation*

$$\partial_t u_h + a\partial_x u_h = \sum_{l=2}^{\infty} c_l \frac{\partial^l u_h}{\partial x_l} \tag{4.5}$$

which is solved *exactly* by the numerical method. An analysis of the eq. (4.5) gives important informations concerning the stability, the accuracy and the dispersive behaviour of methods used. The error coefficients c_l are depending on the advection speed a, the mesh size and the

time step.

To understand the physical effects of the error, we consider the evolution of one single Fourier mode (monochromatic wave) as follows:

$$u(x,t) = Ae^{i(kx-\omega t)}, \quad \text{with } k = \frac{2\pi}{\lambda}, \omega = \frac{2\pi}{T}, i^2 = -1, \tag{4.6}$$

where k denotes the wave number and ω the angular frequency. Substitution of this equation into eq. (4.5), we obtain the *dispersion relation* of the PDE

$$-i\omega + aik = \sum_{l=2}^{\infty} c_l (ik)^l, \quad \text{with } i^l = \begin{cases} (-1)^m, & \text{if} \quad l = 2m \\ i(-1)^m, & \text{if} \quad l = 2m+1 \end{cases} \tag{4.7}$$

Then, the angular frequency ω is defined by

$$\omega = ak + i \sum_{m=1}^{\infty} (-1)^m c_{2m} k^{2m} - \sum_{m=1}^{\infty} (-1)^m c_{2m+1} k^{2m+1} \tag{4.8}$$

The exact solution of eq. (4.5) for a single Fourier mode is given by

$$u(x,t) = Ae^{ik\left(x - t\left(a - \overbrace{\sum_{m=1}^{\infty} (-1)^m c_{2m+1} k^{2m}}^{\text{dispersion error}}\right)\right)} \cdot e^{\overbrace{t \sum_{m=1}^{\infty} (-1)^m c_{2m} k^{2m}}^{\text{diffusion error}}} \tag{4.9}$$

We can clearly identify the dispersion and the diffusion terms. To assure the stability of the methods, this quantity must be ≤ 0, in order to non increase the amplitude of the Fourier mode.

4.3 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 (4.2) with periodic boundary conditions:

$$\partial_t u + a\partial_x u = 0, \quad x \in [0,1]. \tag{4.10}$$

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

$$u = Ae^{i(kx-\xi t)} = Ae^{i(kx-\omega t)}e^{\epsilon t} \qquad \text{with} \quad \xi = \omega + i\epsilon, \quad i = \sqrt{-1}. \tag{4.11}$$

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

$$C = \frac{\omega}{k} \tag{4.12}$$

represents the celerity with which waves propagate in space, and it is in general a function of the wavenumber. Substituting eq. (4.11) in the advection equation eq. (4.10) leads to the

well known result

$$C = a \quad \text{and} \quad \epsilon = 0. \tag{4.13}$$

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 (CFL number) and stabilization parameter δ. In practice, for each choice we will evaluate the accuracy of the discrete approximation of ω and ϵ, and we will provide conditions for the non-positivity of the damping ϵ. For completeness, the study is performed first in the semi-discrete time continuous case in section 4.3.1. We then consider the fully discrete schemes in section 4.3.3.

4.3.1 Preliminaries and time continuous analysis

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

Theorem 4.3.1 (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.,*

$$\int_0^1 u^2(x)dx = \sum_{k\in\mathbb{Z}} |\hat{u}(k)|^2. \tag{4.14}$$

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 eq. (4.11) 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

$$\theta := k\Delta x \tag{4.15}$$

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

$$eq.\ (4.10)\ \text{and}\ eq.\ (4.11) \quad \Rightarrow \quad -i\xi\mathbb{M}\mathbf{U} + a\mathcal{K}_x\mathbf{U} = 0 \tag{4.16}$$

$$\text{with} \quad (\mathbb{M})_{ij} = \int_{\Omega_h} \phi_i\phi_j dx, \qquad (\mathcal{K}_x)_{ij} = \int_{\Omega_h} \phi_i\partial_x\phi_j dx + S(\phi_i,\phi_j), \tag{4.17}$$

with ϕ_j the finite element basis functions and $\mathbf{U}$ the array of all the degrees of freedom. A difference must be pointed out for the SUPG stabilization. In that case the time derivative appears in the stabilization term, hence it contributes to the mass matrix with an additional term

$$\mathbb{M}_{ij} = \sum_{K\in\Omega_h} \int_K (\phi_i\phi_j + \tau_K\partial_x\phi_i\phi_j)dx$$

and the corresponding term must be removed in stabilization term $S(\phi_i,\phi_j)$.

Although system (4.16) is in general a global eigenvalue problem, we can reduce its complexity by exploiting more explicitly the ansatz eq. (4.11). More exactly, we can introduce

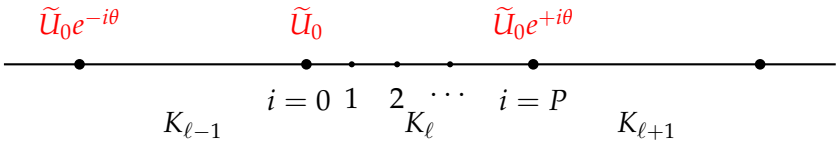
elemental vectors of unknowns $\widetilde{\mathbf{U}}_K$, which, for continuous finite elements, are a arrays of p degrees of freedom including only one of the two boundary nodes. Using the periodicity of the solution and denoting by $K \pm 1$ the neighboring elements, we have

$$\widetilde{\mathbf{U}}_{K\pm1} = e^{\pm i\theta}\widetilde{\mathbf{U}}_K. \tag{4.18}$$

This allows to show that (4.16) is equivalent to a compact system (we drop the subscript $_K$ as this system is equivalent for all cells)

$$-i\xi\widetilde{\mathbb{M}}\widetilde{\mathbf{U}} + a\widetilde{\mathcal{K}}_x\widetilde{\mathbf{U}} = 0, \tag{4.19}$$

where the matrices $\widetilde{\mathbb{M}}$ and $\widetilde{\mathcal{K}}$ are readily obtained from the elemental discretization matrices by using eq. (4.18). In practice, for one dimensional problems, the construction of $\widetilde{\mathbb{M}}$ from $\mathbb{M}$ can be done as follows [116]: if DOFs in the element $K_\ell \in \mathbb{P}_{P-1}$ are placed as follows



Denoting by $\widetilde{K}_\ell = \{0, 1, \cdots, P-1\}$. $\forall (k,j) \in \widetilde{K}_\ell \times \widetilde{K}_\ell$,

$$\widetilde{\mathbb{M}}_\ell[k,j] = \mathbb{M}_\ell[k,j] + \mathbb{M}_\ell[k,P]e^{i\theta}\delta_{j0} + \mathbb{M}_\ell[P,j]e^{-i\theta}\delta_{k0} + \mathbb{M}_\ell[P,P]\delta_{j0}\delta_{k0} \tag{4.20}$$

where $\mathbb{M}_\ell[k,P]e^{i\theta}\delta_{j0}$ corresponds to the contribution of $\widetilde{U}_0 e^{+i\theta}$ on the element $\widetilde{K}_\ell$, $\mathbb{M}_\ell[P,j]e^{-i\theta}\delta_{k0}$ corresponds to the contribution of DOFs from $\widetilde{K}_{\ell-1}$ on $\widetilde{U}_0$, and $\mathbb{M}_\ell[P,P]\delta_{j0}\delta_{k0}$ comes from the assembly procedure of the global mass matrix.

As shown in [116] some particular cases can be easily studied analytically. For example for the semidiscretized $\mathbb{P}_1$ CG scheme without stabilization one easily finds that

$$\frac{\omega}{k} = a\frac{\sin(\theta)}{\theta}\frac{3}{2+\cos(\theta)} \quad \text{and} \quad \epsilon = 0. \tag{4.21}$$

Indeed, considering the neighboring of K, $K \pm 1$, the matricial system is written as follow

$$M = \frac{\Delta x}{6}\begin{pmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{pmatrix}, \qquad K_x = \frac{1}{2}\begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \tag{4.22}$$

and so, using eq. (4.18) the system becomes

$$\left((i\omega+\epsilon)\Delta x\frac{e^{-ik\Delta x}+4+e^{ik\Delta x}}{6} + a\frac{-e^{-ik\Delta x}+e^{ik\Delta x}}{2}\right)U_K = 0 \tag{4.23}$$

$$\left((i\omega+\epsilon)\Delta x\frac{2+\cos(k\Delta x)}{3} + a\sin(k\Delta x)\right)U_K = 0. \tag{4.24}$$

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 [116]) reading

$$\frac{\omega_{1,2}}{k} = a\frac{4\sin(\theta) \pm 2\sqrt{40\sin^2(\frac{\theta}{2}) - \sin^2(\theta)}}{\theta(\cos(\theta) - 3)}. \tag{4.25}$$

Here, two eigenvalues are the solution of the problem, the positive one is the principal one, while the negative one is the parasite one. They are both depicted in fig. 4.1. For more general cases, the study needs to be performed numerically.

Defining with $\lambda_i(\theta)$ the eigenvalues of eq. (4.19), $\omega_i(\theta) = \mathrm{Im}(\lambda_i(\theta))$ 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 eq. (4.19) 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 fig. 4.1 we plot ω and ϵ and we see that the CG scheme does not have diffusive terms, or, in other words, there is no damping ($\epsilon = 0$) in the CG scheme. We plot in fig. 4.1 the principal and the parasite eigenvalues for each system $p = 1, 2, 3$. We can clearly identify the principal one, being the one that minimizes $|\omega_i - ak|$, when $\theta \ll \pi$, while for larger values of θ the distinction is not so clear, from a numerical point of view. As expected, with $\mathbb{P}_1$ elements, the scheme is more dispersive than with $\mathbb{P}_2$ or $\mathbb{P}_3$ elements, i.e., the principal eigenvalue is more distant from the line $\omega = ak$, while, for all of them, there is no dissipation, since the scheme is not stabilized and there is no time discretization providing further dissipation.

We apply the same analysis to stabilized methods. The results obtained with SUPG, CIP and OSS stabilizations lead to almost identical results, that is why we show in fig. 4.2 only the OSS data. A similar analysis, with analogous results, for CIP and standard elements can also be found in [108]. The interested reader can access all the other plots online [87]. 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.

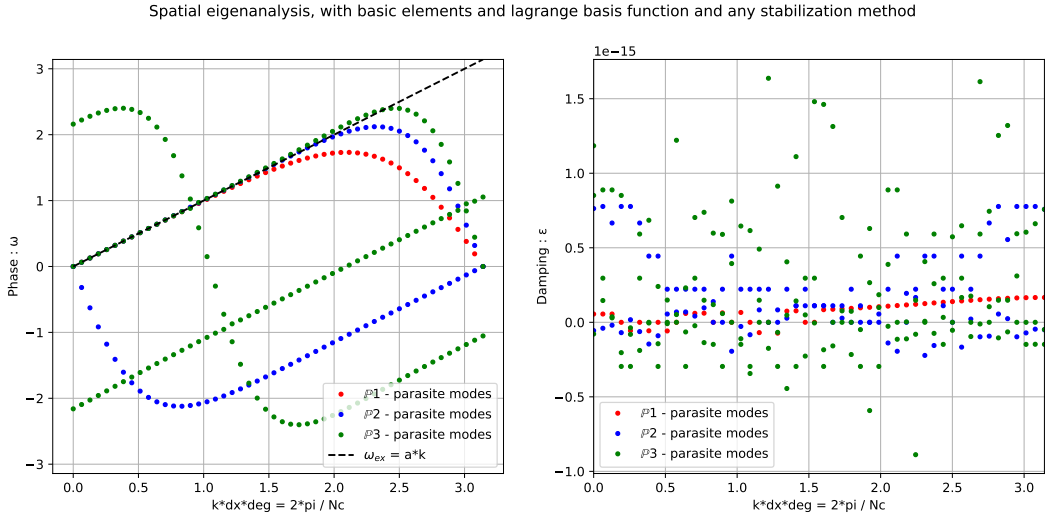


FIGURE 4.1: Phase ω (left) and amplification ϵ (right) with *Basic* elements without stabilization for $\mathbb{P}_1, \mathbb{P}_2$ and $\mathbb{P}_3$.

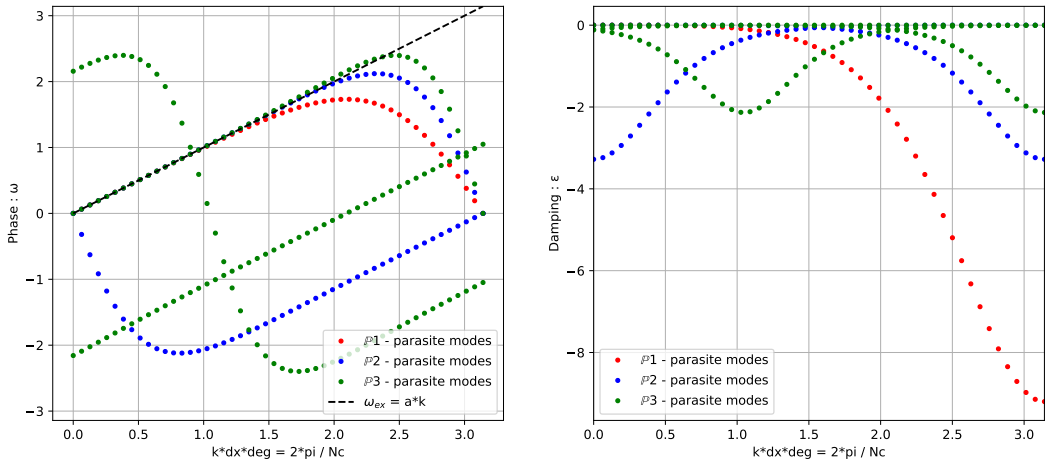


FIGURE 4.2: Phase ω (left) and amplification ϵ (right) with *Basic* elements with OSS stabilization for $\mathbb{P}_1, \mathbb{P}_2$ and $\mathbb{P}_3$.

The number of modes for each system depends on the spatial discretization and p the degree of elements. In other word, the number of modes depends on the number of DOF (degree of freedom per element). In 1D, the number of modes is exactly p, see table 4.1 for the number of modes for all schemes (which is not true in 2D, see table 5.1).

Element	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Cub.	1	2	3
Basic.	1	2	3
Bern.	1	2	3

TABLE 4.1: Summary table of number of modes per systems.

4.3.2 Space-continuous analysis for *Runge-Kutta* schemes

Following the RK or SSPRK formulation, we can write

$$\begin{aligned} U^{(i)} &= \sum_{k=0}^{i-1} \left(\alpha_{ik} U^{(k)} + \Delta t \beta_{ik} L(U^{(k)}) \right), \quad i \in [\![1, m]\!], \quad L(U) = -\partial_x U \\ U^{(0)} &= U^n, \quad U^{(m)} = U^{n+1} \end{aligned} \tag{4.26}$$

where α_{ij} and β_{ik} are defined in section 3.6.

In the case continuous in space and discrete in time, we can also write

$$\begin{aligned} eq.\ (4.10) \quad \& \quad eq.\ (4.11) \quad \Rightarrow \quad U^{n+1} &= U^n + \sum_{\alpha=1}^{step} \gamma_\alpha (-ika)^\alpha (\Delta t)^\alpha U^n \\ \text{Or} \quad \left| U^{n+1} \right| &= (1 + \sum_{\alpha=1}^{step} \gamma_\alpha (-ika)^\alpha (\Delta t)^\alpha) \left| U^n \right| \\ \Rightarrow \quad e^{\epsilon \Delta t} e^{-i\omega \Delta t} &= (1 + \sum_{\alpha=1}^{step} \gamma_\alpha (-ika)^\alpha (\Delta t)^\alpha) = \lambda \\ \Leftrightarrow \quad \frac{w}{k} &= atan\left(\frac{-\mathbb{I}(\lambda)}{\mathbb{R}(\lambda)} \right) \frac{1}{k \Delta t} \end{aligned}$$

Following this approximation, we compute the dispersion C_k/C with $C_k = \frac{w}{k}$ for all wave-number k and $C = a$ in eq. (4.13).
For classical *Runge-Kutta* methods (RK) [117], we show dispersion curves in fig. 4.4. The first and already known result is that the larger the order of accuracy is, the less the method is dispersive. Now we compare the dispersion behaviour with several *Strong-Stability-Preserving Rung-Kutta* methods (SSPRK).

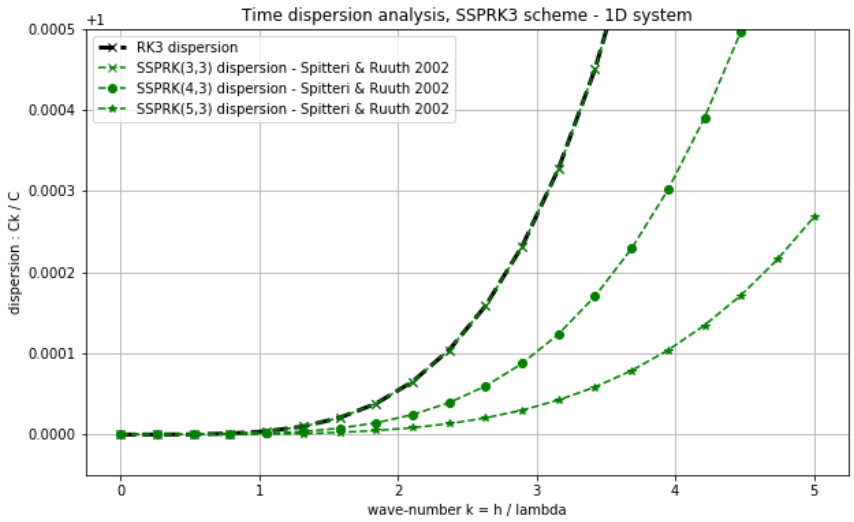


FIGURE 4.3: Dispersion analysis, SSPRK3 schemes

Notation: SSPRK(s,o) corresponds to a o^{th} order of approximation SSPRK method with s steps, with $s \geq o$. And RKo corresponds to a o^{th} order of approximation RK method.

Methods used can be found in [118] for the third order of approximation (see figure 4.3) and [118, 112, 56] for the fourth order (see figure 4.5).

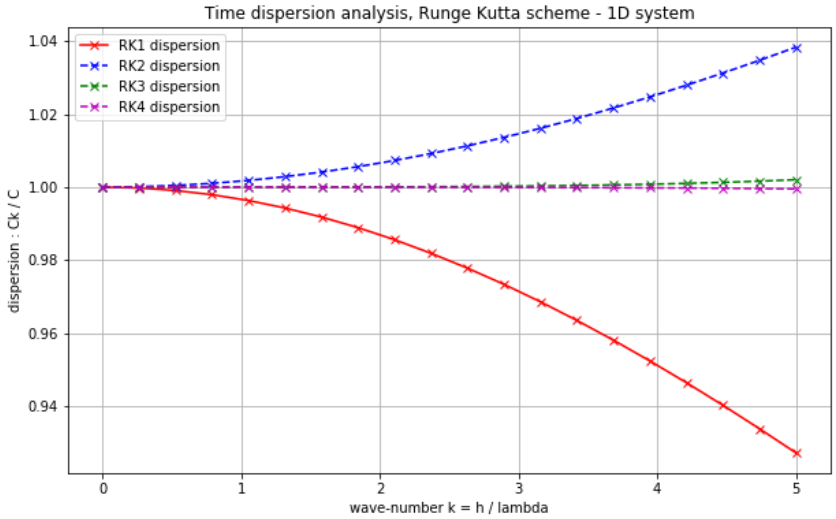


FIGURE 4.4: Dispersion analysis, RK schemes

The figure 4.3 show that the SSPRK(3,3) scheme is as much accurate than RK3. Then, it's clear that the SSPRK(5,3) is the less dispersive than others third order method, including SSPRK(4,3) which already less dispersive than SSPRK(3,3). However, to see the benefit of the use of SSPRK(5,4) instead of SSPRK(4,3) in term of cpu-time, it requires a CFL condition at least multiply by $5/4$. Even if it is possible, we choose to limit our time integration methods at $s = p + 1$ steps, as it is chosen in the Aerosol code.

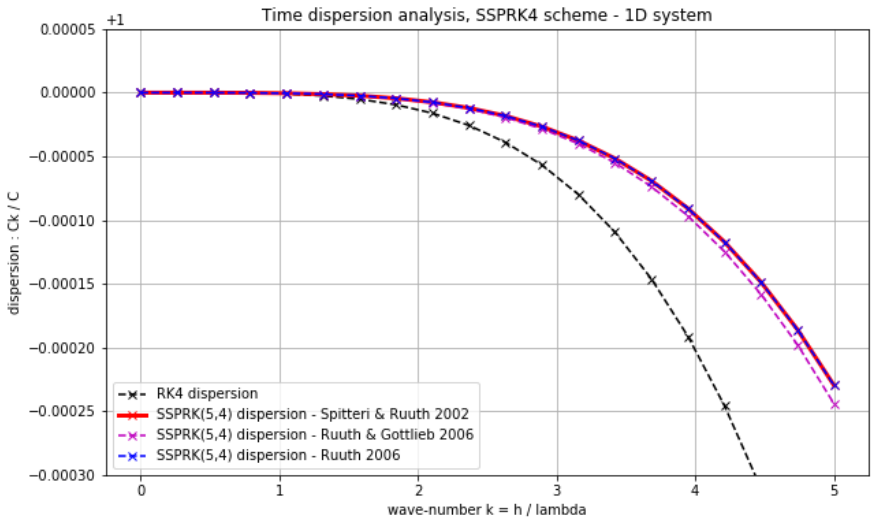


FIGURE 4.5: Dispersion analysis, SSPRK4 schemes

In fig. 4.5, we compare three different types of fourth order SSPRK with 5 steps. They are quite similar in terms of dispersion. Because of negative coefficients in the Butcher's tab of the method *SSPRK(5,4) - Ruuth & Gottlieb 2006*, we decide to do not use it. Indeed, non-negative coefficient schemes are more appropriate to apply to general problems because they do not use downwind-biased (positive coefficients correspond to upwind-biased [112]. Then, knowing that *SSPRK(5,4) - Ruuth 2006* is an improvement of *SSPRK(5,4) - Spitteri & Ruuth 2002* [112], we decide to keep it for our numerical tests.

4.3.3 Fully discrete analysis

Methodology

We analyze now the fully discrete schemes obtained using the RK, SSPRK and DeC time marching methods presented in 3.6. Let us consider as an example the SSPRK schemes 3.48. If we define as $A := \mathbb{M}^{-1}\mathcal{K}_x$ we can write the schemes as follows

$$\begin{cases} \mathbf{U}^{(0)} := & \mathbf{U}^n \\ \mathbf{U}^{(s)} := & \sum_{j=0}^{s-1} \left(\gamma_{sj}\mathbf{U}^{(j)} + \Delta t \mu_{sj} A \mathbf{U}^{(j)} \right), \quad s \in [\![1, S]\!], \\ \mathbf{U}^{n+1} := & \mathbf{U}^{(S)}. \end{cases} \tag{4.27}$$

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

$$\mathbf{U}^{n+1} = \mathbf{U}^{(0)} + \sum_{j=1}^{S} \nu_j \Delta t^j A^j \mathbf{U}^{(0)} = \left(\mathcal{I} + \sum_{j=1}^{S} \nu_j \Delta t^j A^j \right) \mathbf{U}^n, \tag{4.28}$$

where coefficients ν_j in eq. (4.28) are obtained as combination of coefficient γ_{sj} and μ_{sj} in eq. (4.27) 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 eq. (4.18) 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

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

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 formulas for the corresponding phase ω_i and damping coefficient ϵ_i

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

For the DeC method we can proceed with the same analysis transforming also the other involved matrices into their Fourier equivalent ones. Using (3.52) 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 von Neumann 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. In particular, we are looking for the largest CFL number

$$\text{CFL} := |a|\frac{\Delta t}{\Delta x}, \tag{4.29}$$

with constant a, that provides stability to the method [42]. Implicitly, the CFL constraint implies a bound on the timestep Δt. We remark that this CFL constraint is comprehensive of the whole space–time discretization and cannot hence be assumed only by the time scheme or the spatial discretization. In particular for the DeC schemes, it is not possible to decouple the spatial and the time discretization. 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 3.2, we will in particular set

$$\begin{aligned}
\text{SUPG}: \ & \tau_K = \delta \Delta x / |a|, \\
\text{OSS}: \ & \tau_K = \delta \Delta x |a|, \\
\text{CIP}: \ & \tau_f = \delta \Delta x^2 |a|.
\end{aligned}$$

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

$$\begin{aligned}
\left[\frac{u(t) - u_{ex}(t)}{u_{ex}(t)}\right]^2 &= \left[e^{\epsilon t - it(\omega - \omega_{ex})} - 1\right]^2 && (4.30)\\
&= \left[e^{\epsilon t}\cos(t(\omega - \omega_{ex})) - 1\right]^2 + \left[e^{\epsilon t}\sin(t(\omega - \omega_{ex}))\right]^2 && (4.31)\\
&= e^{2\epsilon t} - 2e^{\epsilon t}\cos(t(\omega - \omega_{ex})) + 1. && (4.32)
\end{aligned}$$

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

$$\left[\frac{u(t) - u_{ex}(t)}{u_{ex}(t)}\right]^2 \approx \underbrace{\left[e^{\epsilon t} - 1\right]^2}_{\text{Damping error}} + \underbrace{e^{\epsilon t} t^2 \left[\omega - \omega_{ex}\right]^2}_{\text{Dispersion error}}. \tag{4.33}$$

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

$$\eta_u(\omega, \epsilon)^2 := \frac{3}{2\pi}\left[\int_0^{\frac{2\pi}{3}} (e^{\epsilon} - 1)^2 dk + \int_0^{\frac{2\pi}{3}} e^{\epsilon}(\omega - \omega_{ex})^2 dk\right]. \tag{4.34}$$

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

$$(\mathrm{CFL}^*, \delta^*) := \arg\max_{\mathrm{CFL}} \left\{\eta(\omega(\mathrm{CFL}, \delta)), \epsilon(\mathrm{CFL}, \delta)) < \mu \min_{(\mathrm{CFL},\delta)\text{stable}} \eta(\omega(\mathrm{CFL}, \delta), \epsilon(\mathrm{CFL}, \delta))\right\}. \tag{4.35}$$

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

$$\eta_\omega^2(\omega) := \int_0^{\frac{2\pi}{3}} \left(\frac{\omega - \omega_{ex}}{\omega_{ex}}\right)^2 dk. \tag{4.36}$$

As before we choose the optimal parameters from eq. (4.35).

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.

4.4 Results of the fully discrete spectral analysis

The typical results reported in figs. 4.6 to 4.10 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 4.2 to 4.4.

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

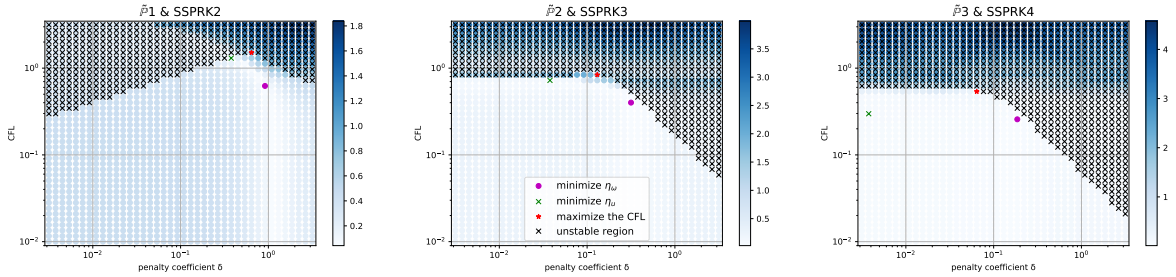


FIGURE 4.6: Computation of optimal parameters according to errors η_ω and η_u. (CFL, δ) plot of η_u (blue scale) and instability area (black crosses) for *Cubature* elements SSPRK scheme with SUPG stabilization method. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$. The purple circle is the optimizer of η_u, the green cross is the optimizer of η_ω, the red star is the maximum stable CFL.

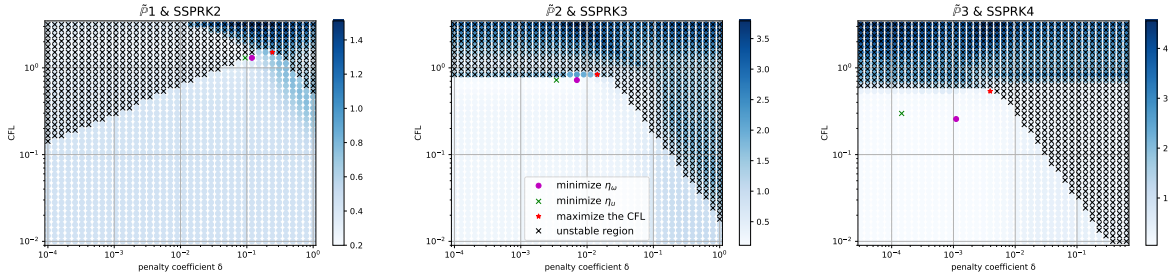


FIGURE 4.7: Computation of optimal parameters according to errors η_ω and η_u. (CFL, δ) plot of η_u (blue scale) and instability area (black crosses) for *Cubature* elements SSPRK scheme with CIP stabilization method. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$. The purple circle is the optimizer of η_u, the green cross is the optimizer of η_ω, the red star is the maximum stable CFL.

mass matrices. Similarly, 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 figs. 4.6 and 4.7, 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 figs. 4.6 and 4.7 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

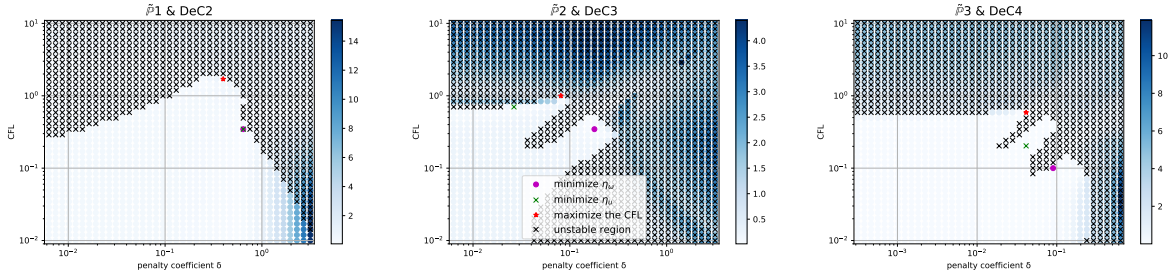


FIGURE 4.8: Computation of optimal parameters according to errors η_ω and η_u. (CFL, δ) plot of η_u (blue scale) and instability area (black crosses) for *cubature* elements DeC scheme with SUPG stabilization method. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$. The purple circle is the optimizer of η_u, the green cross is the optimizer of η_ω, the red star is the maximum stable CFL.

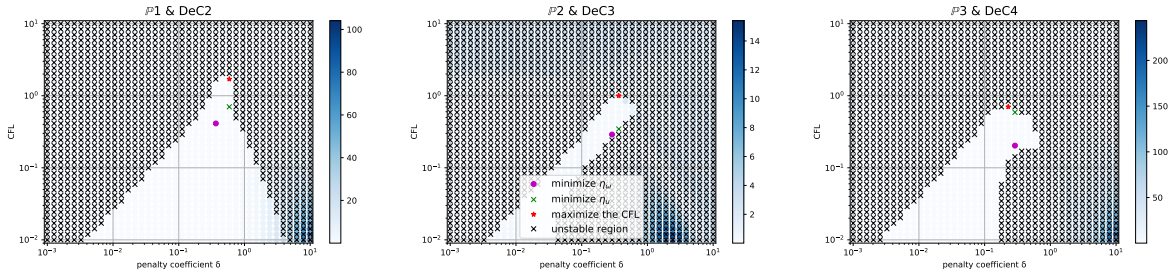


FIGURE 4.9: Computation of optimal parameters according to errors η_ω and η_u. (CFL, δ) plot of η_u (blue scale) and instability area (black crosses) for *Bernstein* elements DeC scheme with SUPG stabilization method. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$. The purple circle is the optimizer of η_u, the green cross is the optimizer of η_ω, the red star is the maximum stable CFL.

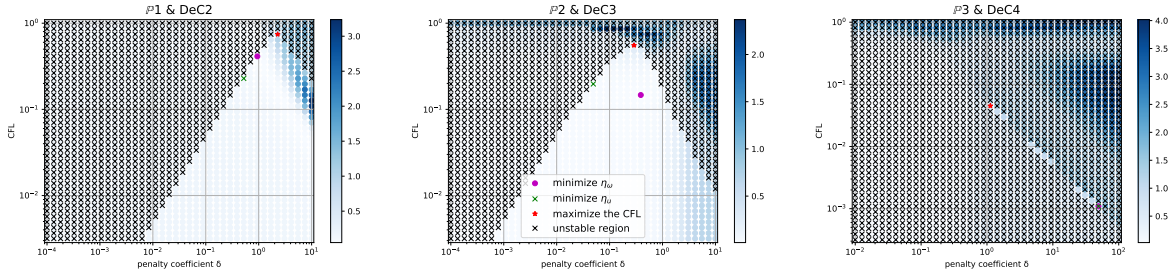


FIGURE 4.10: Computation of optimal parameters according to errors η_ω and η_u. (CFL, δ) plot of η_u (blue scale) and instability area (black crosses) for *Basic* elements DeC scheme with OSS stabilization method. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$. The purple circle is the optimizer of η_u, the green cross is the optimizer of η_ω, the red star is the maximum stable CFL.

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 $\mathrm{CFL} > (C_3/C_1)^{1/C_2}$ and $\delta > C_3$. This is for example visible in fig. 4.9. In the second case, see for example fig. 4.8, we also have two bounds of the type $\mathrm{CFL} \geq C_1$ 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 4.5.

- Finally, the DeC scheme with *Basic* elements and $p = 3$ shows essentially everywhere instability for CIP and OSS stabilization. The study finds some very thin oblique stripes of stability, but they are not wide enough to find stable regions. See fig. 4.10 for an example.

4.4.1 Dispersion and damping

In figs. 4.11 and 4.12 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 OSS 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

[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	/	0.389	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)*	0.492 (0.089)*
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)*	0.588 (0.041)*
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)*	0.702 (0.229)*

Element &		OSS			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 4.2: Optimized CFL and penalty coefficient δ in parenthesis, only maximizing CFL. The sign / means unconditionally unstable.
* These values do not allow to decrease the CFL.

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 D. There we used the (CFL,δ) coefficients found by minimizing η_u in table 4.3, and we try also to compare the obtained results. Nevertheless, 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 4.3, we can compare the different elements, stabilization terms and time integration techniques and obtain some conclusions.

- All the first order unstabilized $p = 1$ schemes are unconditionally unstable, i.e., for all CFL.
- In general SSPRK time integration methods allow to use higher CFL with respect to both classical RK methods and DeC. In particular, for some of these tests CFL> 1, meaning that the combination of spatial discretization and the time discretization allow to set the ratio $\Delta t/\Delta x$ larger than usual. This should not be a surprise as the SSPRK schemes are taylored to maximize the CFL number.

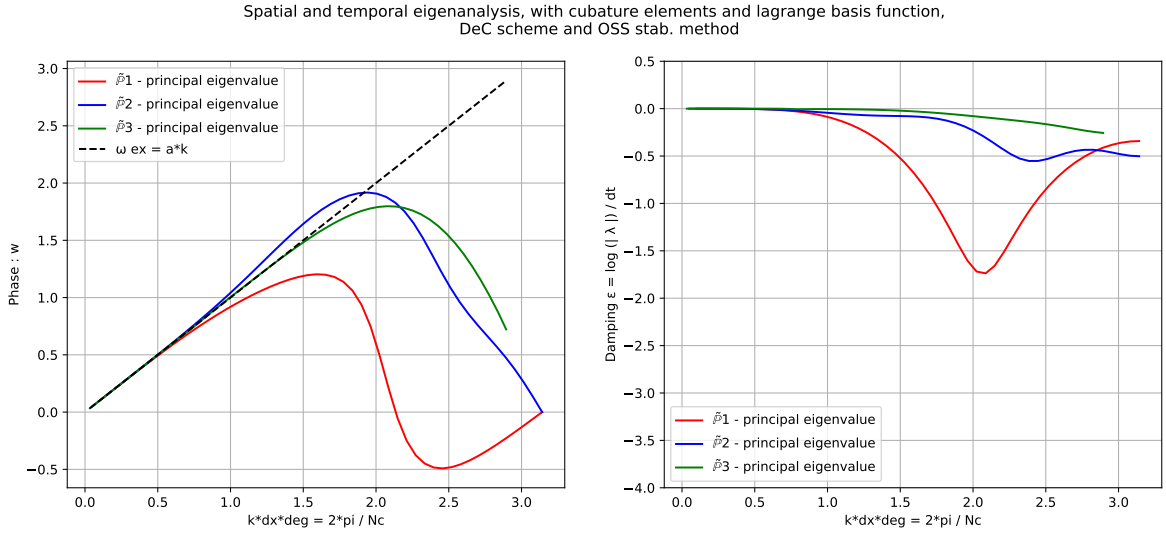


FIGURE 4.11: Comparison of dispersion in the fully discrete case, using coefficients from 4.3, *Cubature* elements, DeC scheme and OSS stabilization method. $\mathbb{P}_1$ elements in red, $\mathbb{P}_2$ elements in blue and $\mathbb{P}_3$ elements in green. The phase ω of the principal eigenvalues is on the left and the damping ϵ_i on the right

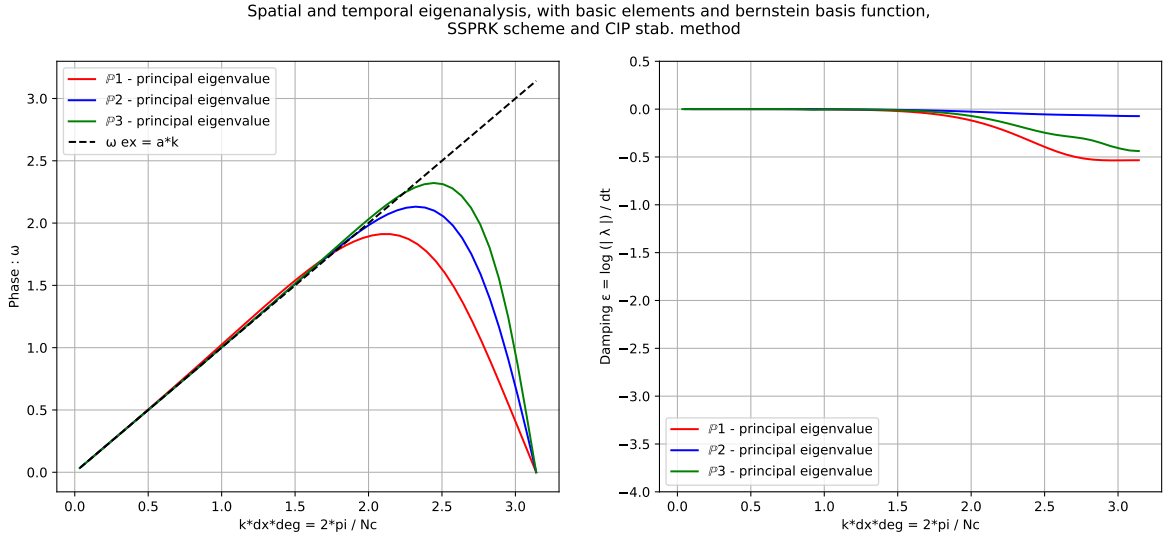


FIGURE 4.12: Comparison of dispersion in the fully discrete case, using coefficients from 4.3, *Bernstein* elements, SSPRK scheme and CIP stabilization method. $\mathbb{B}_1$ elements in red, $\mathbb{B}_2$ elements in blue and $\mathbb{B}_3$ elements in green. The phase ω of the principal eigenvalues 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	/	0.151	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)*	0.588 (0.289)*

Element &		OSS			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 4.3: Optimized CFL and penalty coefficient δ in parenthesis, minimizing η_u. The sign / means unconditionally unstable.
* These values do not allow to decrease the CFL.

- With *Cubature* elements we can use larger CFL 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 OSS, 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 OSS 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 4.2 to 4.4 such schemes and we put in table 4.5 reliable values of (CFL,δ).

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)*	0.017 (0.113)*
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)*	0.1 (0.09)*
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)*	0.203 (0.289)*

Element &		OSS			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 4.4: Optimized CFL and penalty coefficient δ in parenthesis, minimizing η_ω. The sign / means unconditionally unstable.
* These values do not allow to decrease the CFL.

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

4.5 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 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 formulas.

Consider any convex smooth entropy $\rho(u)$, i.e., $\rho_{uu}(u) > 0$, the respective entropy variables $v := \rho_u(u)$ and the entropy flux $g(u)$ such that $\rho_u f_u = g_u$. In the following discussion, we consider the entropy variable $v_h = \rho_u(u)_h$ 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 $v \to u = u(v)$, as proposed in [1].

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

$$\sum_K \int_K \nu_h \left(\partial_t u_h + \partial_x f(u_h)\right) dx = \sum_K \int_K \partial_t \rho_h + \partial_x g_h dx = \int_\Omega \partial_t \rho_h + [g_h]_{\partial K}, \tag{4.37}$$

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

$$s(v,u) := \sum_{K,\mathsf{f}\in K} \int_{\mathsf{f}} [\partial_x v^T] \rho_{uu}(u)^{-1} [\partial_x \nu(u)] d\Gamma, \tag{4.38}$$

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

$$s(\nu,u_h) = \sum_{K,\mathsf{f}\in K} \int_{\mathsf{f}} \underbrace{[\partial_x \nu_h^T] \rho_{uu}(u_h)^{-1} [\partial_x \nu_h]}_{>0} d\Gamma. \tag{4.39}$$

It would guarantee a decrease in the discrete total entropy. Moreover, this formulation coincide with (3.21) when we are dealing with the energy as entropy.

For the OSS we modify, similarly the formulation (3.25) into

$$\begin{cases} s(v,u) := \sum_K \tau_K \int_K \partial_x v^T \rho_{uu}(u)^{-1} (\partial_x \nu(u) - w) dx, \text{ with} \\ \int_K z^T (w - \partial_x \nu(u)), \quad \forall z \in V_h \end{cases} \tag{4.40}$$

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

$$\begin{aligned} s(\nu_h,u_h) = &\sum_K \tau \int_K \partial_x \nu_h^T \rho_{uu}(u_h)^{-1} (\partial_x \nu_h - w_h) + \rho_{uu}(u_h) w_h^T \rho_{uu}(u_h)^{-1} (w_h - \partial_x \nu_h) dx = \\ &\sum_K \tau \int_K (\partial_x \nu_h - w_h)^T \rho_{uu}(u_h)^{-1} (\partial_x \nu_h - w_h) dx \geq 0. \end{aligned} \tag{4.41}$$

As for the CIP we can say that the OSS 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 3.2.1 shows, the spatial and temporal derivatives need to be properly combined. This can be done easily for space-time discretizations (see e.g. in [10]), 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 [1, 73, 58], 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.

4.6 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 4.3. The integral formulas 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 formulas with $p+1$ points per cell.

4.6.1 Linear advection equation

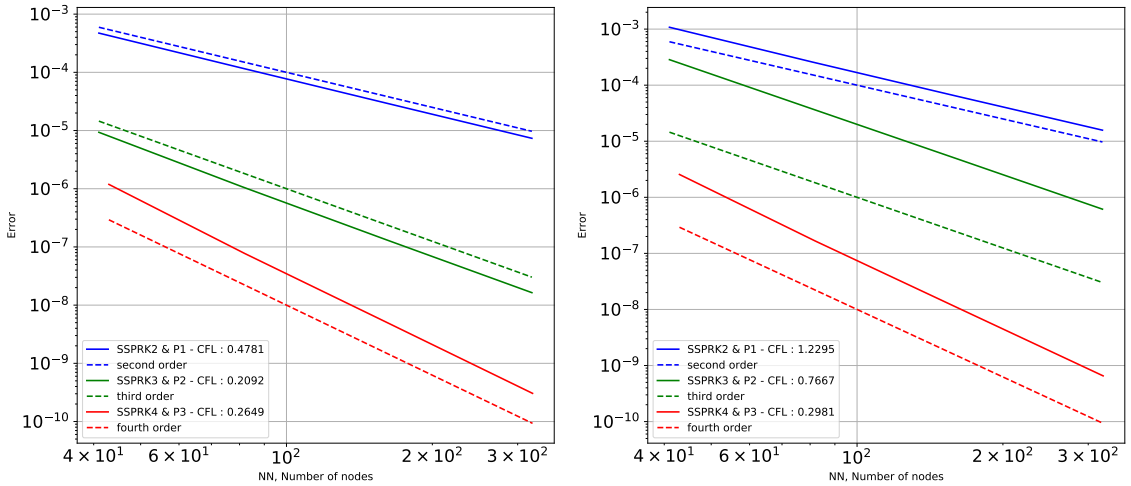
We start with the one dimensional initial value problem for the linear advection eq. (4.10) on the domain $\Omega = [0,2]$ using periodic boundary conditions:

$$\begin{cases} \partial_t u(x,t) + a\partial_x u(x,t) = 0 & (x,t) \in \Omega \times [0,5], \quad a \in \mathbb{R}, \\ u(x,0) = u_0(x), & \\ u(0,t) = u(2,t), & t \in [0,5], \end{cases} \tag{4.42}$$

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 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 of 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, OSS and CIP) and with *Basic*, *Cubature* and *Bernstein* elements.

A representative result is provided as an example in figs. 4.13(a) and 4.13(b): it shows a comparison between *Cubature* and *Basic* elements with OSS stabilization and SSPRK time integration. As we can see, the two schemes have very similar error behavior, but the *Basic* elements require stricter CFL conditions, see table 4.3, and have larger computational costs because of the inversion of the mass matrix. A summary table with the order of accuracy reached by each simulations in table 4.6. The plots and all the errors are available at the repository [87].



(a) *Basic* elements. (b) *Cubature* elements.

FIGURE 4.13: Error decay for linear advection with the OSS stabilization and SSPRK. $\mathbb{P}_1$, $\mathbb{P}_2$ and $\mathbb{P}_3$ elements are, respectively, in blue green and red.

Element &		No stabilization			SUPG			OSS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
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	2.81**	3.98	2.0	3.05**	3.98	2.0	3.03**	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	2.81**	3.98	2.0	3.05**	3.98	2.0	3.03**	3.97
	DeC	/	/	/	/	/	/	1.98	3.05	2.04	1.98	3.0	2.0

TABLE 4.6: Summary table of convergence orders, using coefficients obtained by minimizing η_u in table 4.3.
** These values are computed using parameters from the minimization of η_ω in table 4.4

Looking at the table we can make the following observations. First of all, we remark that despite the weak stability obtained for unstabilized methods 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 fail in doing so and deserve some comments. In particular, we notice the failure of DeC for *Basic* $\mathbb{P}_3$ and *Bernstein* $\mathbb{B}_3$ elements.While disappointing, this negative result is not completely new. Indeed, in [107] obtaining correct convergence with DeC for some orders required both increasing the number of substeps, 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, for instance using time relaxation methods [40, 11],

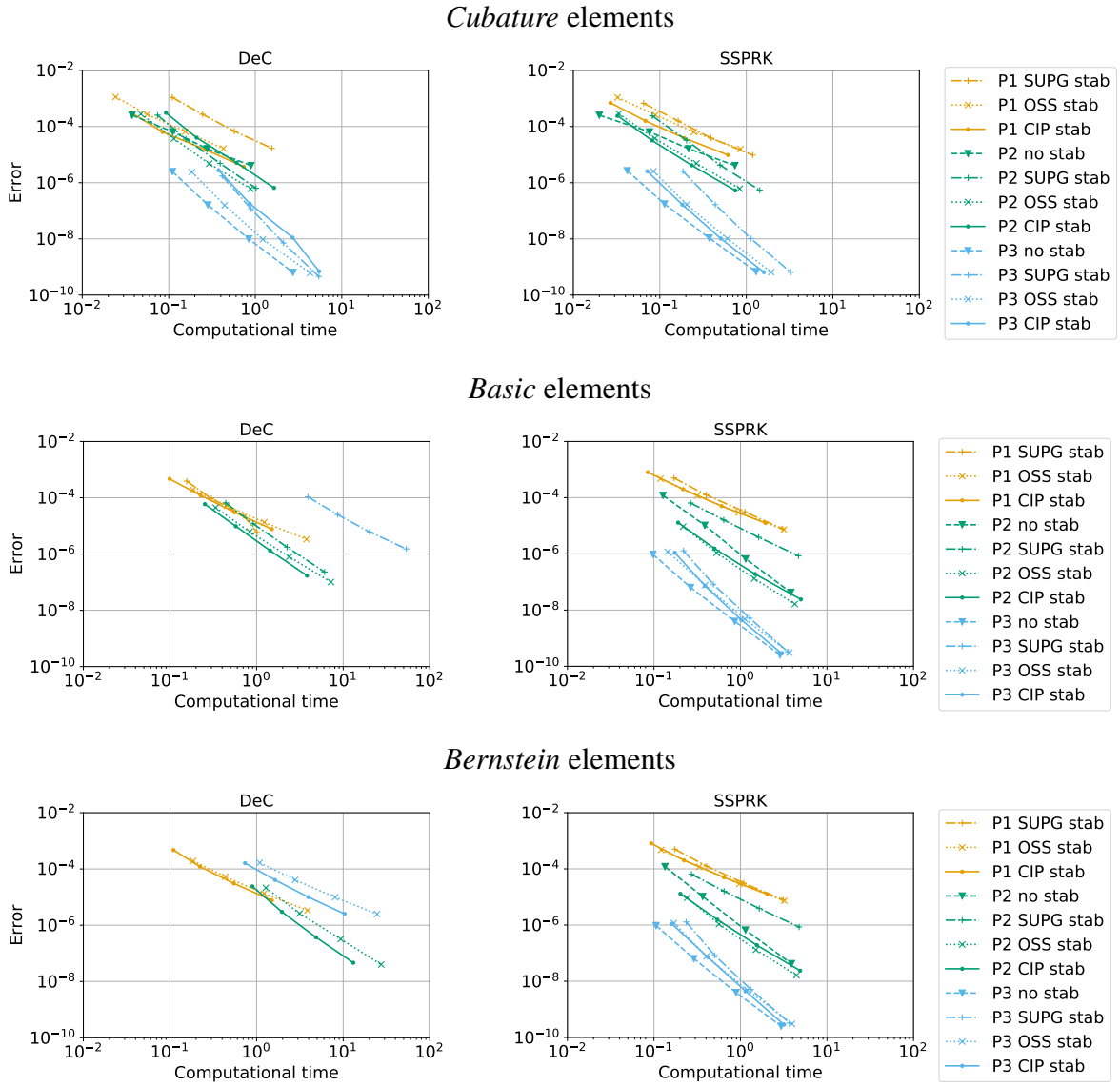


FIGURE 4.14: Error for linear advection problem (4.42) with respect to computational time for all elements and stabilization techniques: DeC on the left, SSPRK on the right

or using the jumps of high-order derivatives of variables [62]. 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 fig. 4.14 where we plot the error of each method against computational time. Note that the simulations are all obtained using the CFL and the penalty coefficient δ reported in table 4.3, except in particular cases** where the minimization process with η_u found values that do not dissipate enough the most dispersive waves, hence for these schemes we use the parameters reported in table 4.4. 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 4.3 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, OSS and CIP behave very similarly (also their CFLs 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 CFL 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.

Such a care in avoiding the inversion of mass matrices is meaningful when talking about mass matrices coming, for example, from multi-dimensional problems or, at least, high order methods. In simple cases where the mass matrix is tridiagonal ($\mathbb{P}_1$ elements in one dimensional problems), the linear systems given by the mass matrix can be solved with an $\mathcal{O}(N)$ of arithmetical computations, hence, not changing the computational cost order of these types of methods.

To see the benefit of stabilization techniques when the initial solution is not continuous, we consider now the *step* initial data

$$u(x,0) = \begin{cases} 1, & \text{if } x < 1.1, \\ 0, & \text{else.} \end{cases} \tag{4.43}$$

For this study, with consider $t_f = 0.35s$ and 201 nodes, i.e. $\Delta x_1 = 0.01$, $\Delta x_2 = 0.02$ and $\Delta x_3 = 0.03$. As expected, all stabilization terms reduce numerical instabilities which

Cubature elements and SSPRK schemes

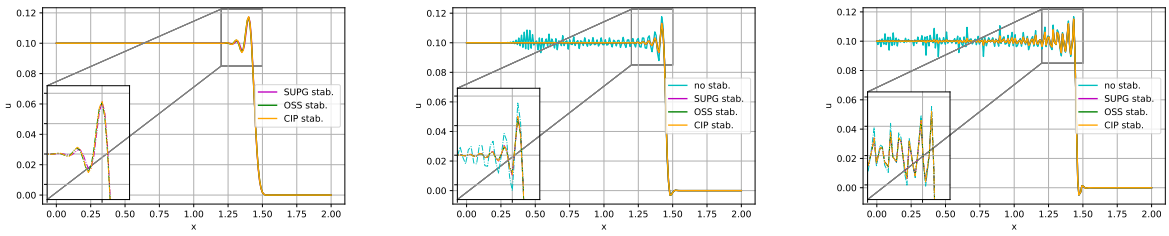


FIGURE 4.15: Solution of linear advection equation with discontinuous initial condition using *Cubature* elements and SSPRK schemes: $\mathbb{P}_1$ at left, $\mathbb{P}_2$ at the center and $\mathbb{P}_3$ at right.

appear without any stabilization (in cyan in fig. 4.15). The SUPG, OSS and CIP techniques behave similarly, moreover the first order unstabilized method shows wild oscillations that scale differently from all the other solutions. All the stabilized solutions have comparable accuracy for all orders.

4.6.2 Application to 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

$$\begin{cases} \partial_t u(x,t) + \partial_x \frac{u^2(x,t)}{2} = 0 & (x,t) \in \Omega \times [0, t_f], \\ u(x,0) = u_0(x), & x \in \Omega \\ u(x_D,t) = g(x_D,t), & x_D \in \partial\Omega, \end{cases} \tag{4.44}$$

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

$$\chi = x - u_0(\chi)t \tag{4.45}$$

for all $(x,t) \in \Omega \times [0,t_f]$, solving the nonlinear equation (4.45) for χ at every point (x,t). To obtain the exact solution we employed the Broyden method implemented in SciPy library [125]. Note that the analytical solution shows a shock at time

$$t_s = -\frac{1}{\min\limits_{x\in\Omega} u_0'(x)} = \frac{1}{4}. \tag{4.46}$$

This knowledge allows to set for this study $t_f = 0.5t_s = 0.125$, 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 4.3 we obtain the experimental order of convergence in table 4.7.

Element &		No stabilization			OSS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
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 4.7: Summary table of convergence order, using coefficients obtained in table 4.3. The sign / means unstable.

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 [87]. Here we focus on the comparison between error and computational time, reported in fig. 4.16.

Again for *Cubature* elements it is clear that there is an advantage in using high order methods, in particular for SSPRK methods, which has less stages than DeC. For this test, we only compare CIP and OSS 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

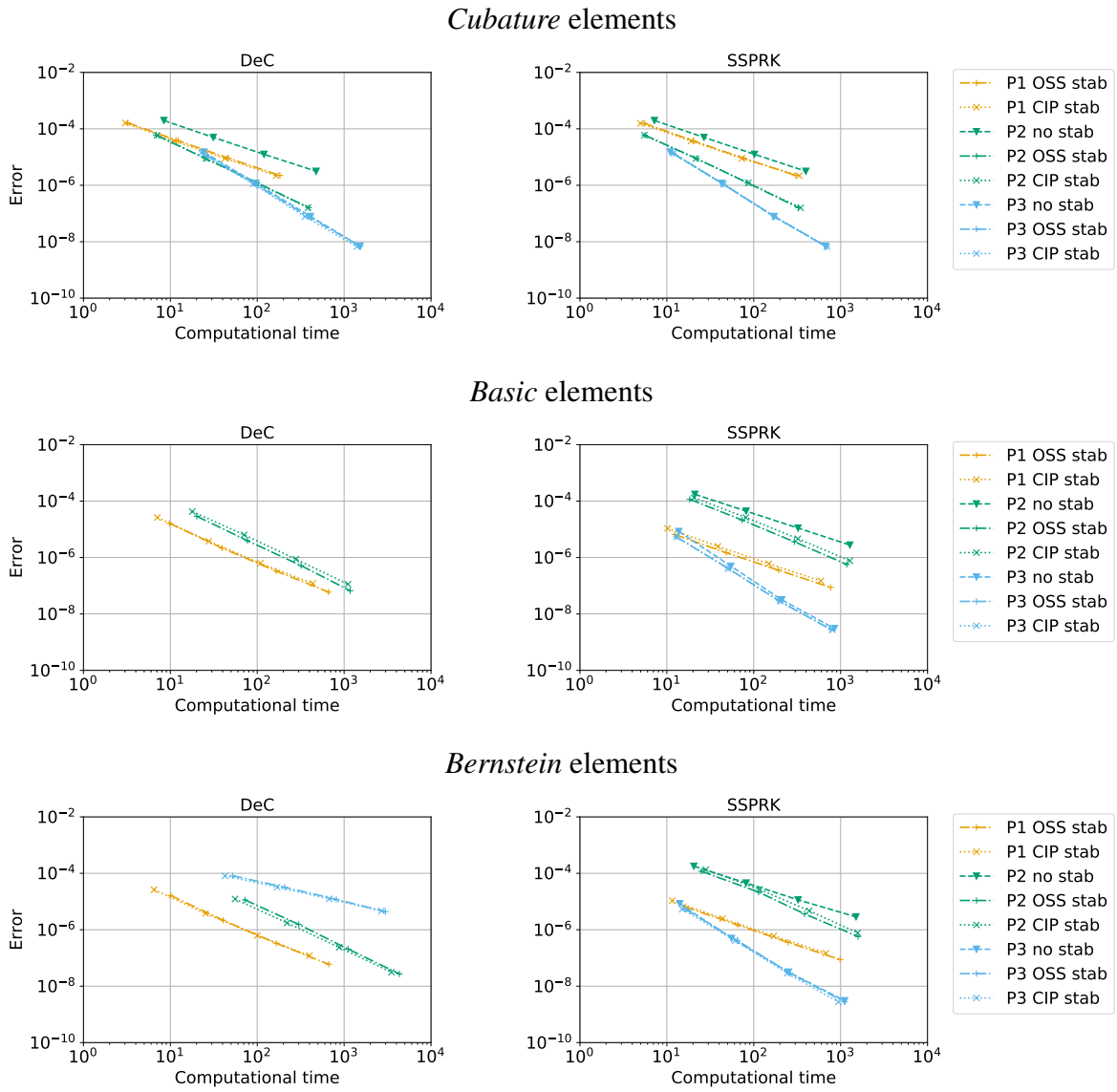


FIGURE 4.16: Error for Burgers' equation (4.44) with respect to computational time for all elements and stabilization techniques: DeC on the left, SSPRK on the right

multidimensional case where the OSS may be penalized on elements requiring the inversion of the 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 4.7. 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.

To see the benefit of stabilization techniques when a shock occurs, we consider now $t_f = 0.3 > t_s$ in fig. 4.17. The simulation is done using 201 nodes, i.e. $\Delta x_1 = 0.01$, $\Delta x_2 = 0.02$ and $\Delta x_3 = 0.03$. As expected, all the solutions introduce some numerical dispersion around the shock, even if, the $\mathbb{L}_2$ norm of the solutions is dissipated. As we can see, stabilization terms slightly reduce the numerical instability which appears in the simulations without any stabilization (in cyan in fig. 4.17). Once again, OSS and CIP behave

Cubature elements and SSPRK schemes

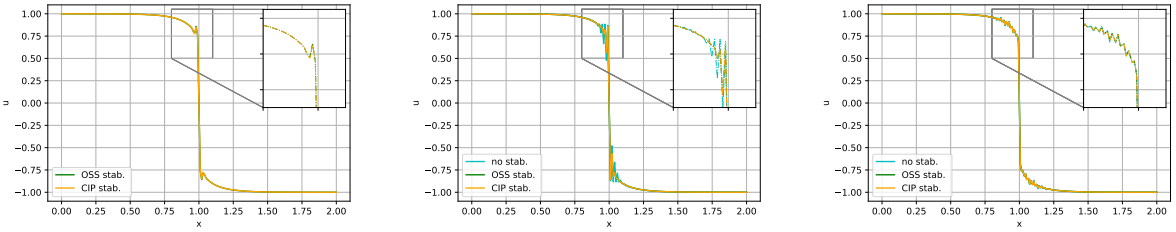


FIGURE 4.17: Non linear instabilities for Burgers' equation (4.44) when $t_f > t_s$ using *Cubature* elements and SSPRK schemes: $\mathbb{P}_1$ at left, $\mathbb{P}_2$ at the center and $\mathbb{P}_3$ at right.

similarly with a shock, and first order accurate schemes behave slightly better than high order schemes when a shock occurs.

4.6.3 Application to Shallow Water equations

As a final application we consider the non linear Shallow Water equations:

$$\begin{cases} \partial_t h + \partial_x(hu) & = 0, \\ \partial_t(hu) + \partial_x(hu^2 + g\frac{h^2}{2}) + \Phi & = 0, \end{cases} \quad x \in \Omega,\ t \in [0,5]. \tag{4.47}$$

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

$$\begin{cases} h_{ex}(x,t) = h_0 + \epsilon h_0 \text{sech}^2(\kappa(x - ct)), \\ u_{ex}(x,t) = c\left(1 - \frac{h_0}{h_{ex}(x,t)}\right), \\ \kappa = \sqrt{\frac{3\epsilon}{4h_0^2(1+\epsilon)}}, \quad c = \sqrt{gh_0(1+\epsilon)}. \end{cases} \tag{4.48}$$

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 4.8 we show the convergence orders for this Shallow Water problem with the CFL and δ coefficients found in table 4.3.

Element &		No stabilization			OSS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
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 4.8: Summary tab of convergence order, using coefficients obtained by minimizing η_u. The sign / means unstable.

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* elements for both $\mathbb{B}_2$ and $\mathbb{B}_3$. 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, fig. 4.18, in favor of these higher polynomial degrees.

4.7 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 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 [122, 39, 55] and references therein. Note that the parameters found in the present study may not provide stable results in all cases when passing to multiple space dimensions, especially when considering non-tensorial representations as e.g. on simplex elements. However, our preliminary investigations suggest that similar constraints can be formulated also in these cases.

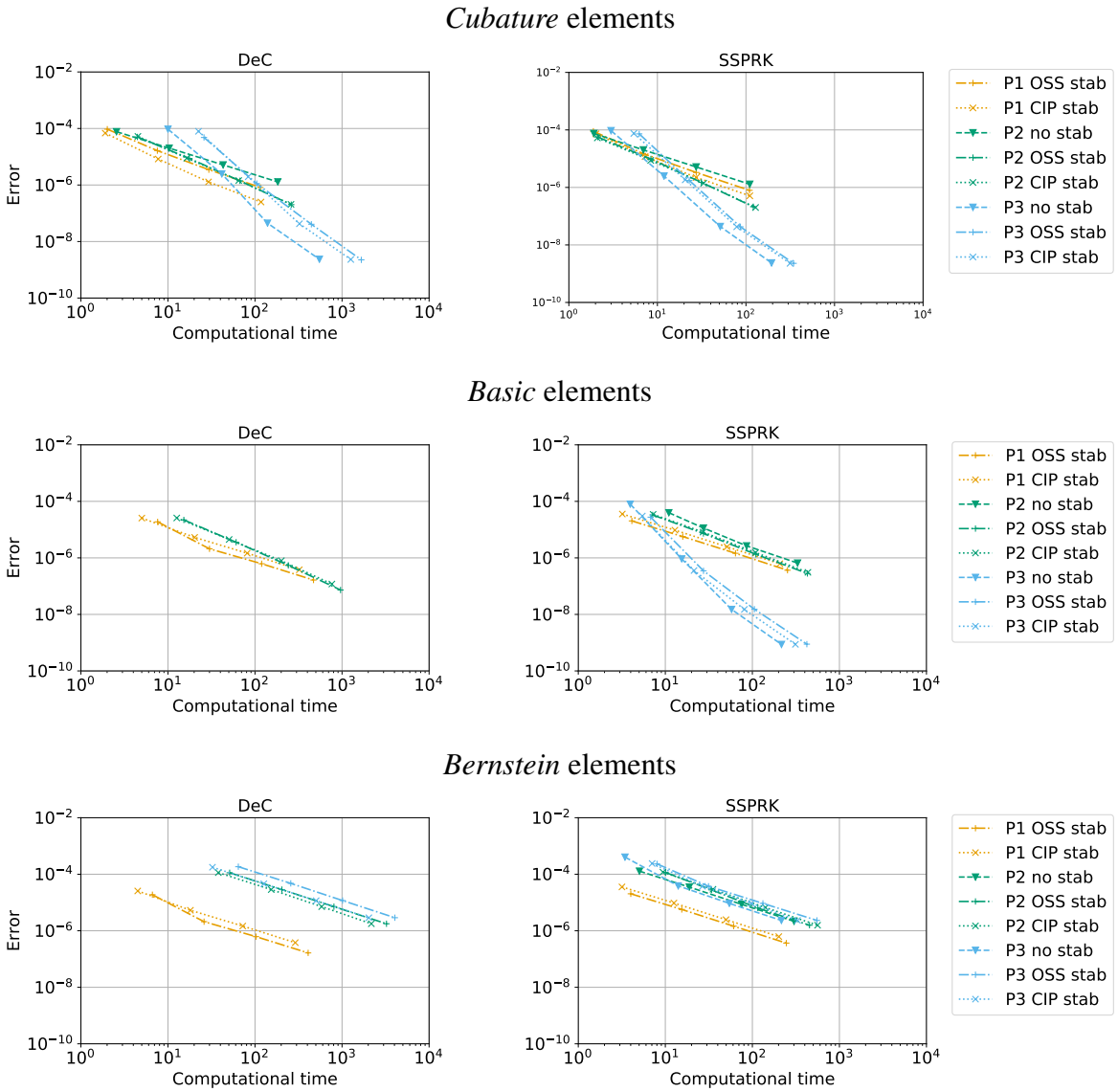


FIGURE 4.18: Error for Shallow Water equations (4.47) with respect to computational time for all elements and stabilization techniques: DeC on the left, SSPRK on the right

Chapter 5

Extension to the two-dimensional formulation

Chapter Abstract
In this chapter, we study continuous finite element dicretizations for two-dimensional hyperbolic partial differential equations. The main contribution of this chapter as in [88], 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 orthogonal subscale stabilization (OSS). Three different choices for the continuous finite element space are compared: Bernstein polynomials, Lagrangian polynomials on equispaced nodes, and Lagrangian polynomials on 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.
The new aspect of this study comparing with the mono-dimensional one [88], is the additional degree of freedom of mesh topology, whose influence is also accounted for. In particular, fully discrete Fourier analysis are performed considering two different mesh configurations and different wave angles. 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 OSS stabilization are the most promising combinations.

Outline

5.1 Introduction 70
5.2 Fourier Analysis 70
5.2.1 Preliminaries and time continuous analysis 70
5.2.2 The eigenvalue system 71
5.2.3 The fully discrete analysis 73

5.2.4 Methodology 74
5.2.5 Results of the fourier analysis using the *X* type mesh 77
5.2.6 Comparison with a space-time split stability analysis 79
5.2.7 Different mesh patterns 81
5.2.8 Final results of the stability analysis 82
5.2.9 Complementary analysis using viscosity terms 85
5.3 Validation of the fourier analysis 87
5.3.1 Linear advection equation test 87
5.3.2 Shallow Water equations 90
5.4 Numerical Simulations on arbitrary mesh 92
5.4.1 Linear advection test 93
5.4.2 Shallow Water equations 95
5.4.3 Remark on the steady vortex case 96
5.5 Conclusion 96

5.1 Introduction

We will perform a spectral analysis of the two-dimensional form of the continuous stabilized finite elements methods introduced in chapter 3. To this end we consider the scalar advection equation

$$\partial_t u(x,t) + \vec{a} \cdot \nabla u(x,t) = 0 \quad x \in \Omega \subset \mathbb{R}^2,\, t \in \mathbb{R}^+, \tag{5.1}$$

with periodic boundary conditions. The analysis follows and extends the one of chapter 4 (cf. [88]).

5.2 Fourier Analysis

5.2.1 Preliminaries and time continuous analysis

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

$$\partial_t u(t,\mathbf{x}) + \mathbf{a} \cdot \nabla u(t,\mathbf{x}) = 0, \quad \mathbf{a} \in \mathbb{R}^2, \quad (t,\mathbf{x}) \in \mathbb{R} \times \Omega, \tag{5.2}$$

with $\Omega = [0,1] \times [0,1]$. For simplicity, we consider $\mathbf{a} = (\cos(\Phi), \sin(\Phi))$ with $\Phi \in [0, 2\pi]$.
The main idea is to investigate the fully discrete evolution of periodic waves using the following ansatz for the approximated solution

$$u_h(\mathbf{x},t) = A e^{i(\mathbf{k}\cdot\mathbf{x} - \xi t)} = A e^{i(\mathbf{k}\cdot\mathbf{x} - \omega t)} e^{\epsilon t} \tag{5.3}$$

$$\text{with} \quad \xi = \omega + i\epsilon, \quad i = \sqrt{-1}, \quad \mathbf{k} = (k_x, k_y)^T. \tag{5.4}$$

Here, ϵ denotes the damping rate, while the wavenumbers are denoted by $\mathbf{k} = (k_x, k_y)$, with $k_x = 2\pi/L_x$ and $k_y = 2\pi/L_y$ with L_x and L_y the wavelengths in x and y directions respectively. In the ansatz we also include the parametrization of the wave number as proportional to the speed $\mathbf{a}$, i.e., $(k_x, k_y) = (k\cos(\Phi), k\sin(\Phi))$.
We recall that the phase velocity defined as

$$C = \frac{\omega}{k} \tag{5.5}$$

represents the celerity with which waves propagate in space, and it is in general a function of the wavenumber. Substituting (5.3) in the advection equation (5.2) for an exact solution we obtain that

$$C = |\mathbf{a}| \quad \text{and} \quad \epsilon = 0. \tag{5.6}$$

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

5.2.2 The eigenvalue system

The Fourier analysis for numerical schemes on the periodic domain is based on the Parseval theorem 4.3.1.

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

$$\theta_x := k_x \Delta x \ \text{ and } \ \theta_y := k_y \Delta y \tag{5.7}$$

are the discrete reduced wave numbers which naturally appear all along the analysis. Here, Δx and Δy are defined by the size of the elementary periodic unit that is highlighted with a red square as an example in fig. 5.1.

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

$$\text{(5.2) and (5.3)} \quad \Rightarrow \quad -i\xi\mathbb{M}\mathbf{U} + \mathbf{a}\cdot(\mathcal{K}_x\mathbf{U}, \mathcal{K}_y\mathbf{U}) + \delta\mathbb{S}\mathbf{U} = 0 \tag{5.8}$$

$$\text{with} \quad (\mathbb{M})_{ij} = \int_\Omega \phi_i\phi_j dx, \qquad (\mathcal{K}_x)_{ij} = \int_\Omega \phi_i\partial_x\phi_j dx, \qquad (\mathcal{K}_y)_{ij} = \int_\Omega \phi_i\partial_y\phi_j dx \tag{5.9}$$

with ϕ_j being any finite element basis functions, $\mathbf{U}$ the array of all the degrees of freedom and $\mathbb{S}$ being the stabilization matrix defined through one of the stabilization techniques of

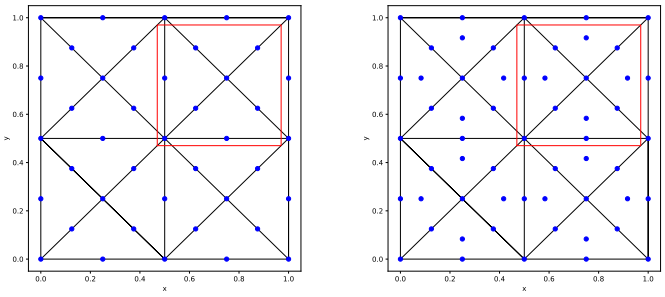


FIGURE 5.1: The X type triangular mesh. At left, the *Basic* finite element discretisation with $\mathbb{P}_2$ elements. At right, the grid configuration for $\tilde{\mathbb{P}}_2$ *Cubature* elements. The red square represents the periodic elementary unit that contains the degrees of freedom of interest for the Fourier analysis

section 3.2. Although system (5.8) is in general a global eigenvalue problem, we can reduce its complexity by exploiting more explicitly the ansatz (5.3). More precisely, as it is done in [115] we can introduce elemental vectors of unknowns $\widetilde{\mathbf{U}}_{K_{ij}}$, which, for continuous finite elements, is an array of d degrees of freedom inside a periodic unitary block, excluding two boundaries (one on the top and one on the right for example). This number depends on the chosen (periodic) mesh type and on the elements. As an example, in fig. 5.1 we display for the X type mesh the periodic elementary unit (in the red square) with *Basic* and *Cubature* degrees of freedom with $p = 2$. All the degrees of freedom inside the red square are repeated periodically in the mesh and we consider only one block of degrees of freedom inside $\widetilde{\mathbf{U}}_{K_{i,j}}$. In the X mesh for *Basic* elements $p = 2$ we have $d = 8$, while for *Cubature* $p = 2$ we have $d = 12$. Using the periodicity of the solution and the ansatz (5.3) and denoting by $K_{i\pm1,j\pm1}$ the neighboring elementary units, we can write the neighboring degrees of freedom by

$$\widetilde{\mathbf{U}}_{K_{i\pm1,j}} = e^{\pm\theta_x}\widetilde{\mathbf{U}}_{K_{i,j}}, \qquad \widetilde{\mathbf{U}}_{K_{i,j\pm1}} = e^{\pm\theta_y}\widetilde{\mathbf{U}}_{K_{i,j}}, \tag{5.10}$$

and by induction all other degrees of freedom of the mesh. This allows to show that (5.8) is equivalent to a compact system of dimension d (we drop the subscript $_K$ as they system is equivalent for all cells)

$$-i\xi\widetilde{\mathbb{M}}\widetilde{\mathbf{U}} + a_x\widetilde{\mathcal{K}}_x\widetilde{\mathbf{U}} + a_y\widetilde{\mathcal{K}}_y\widetilde{\mathbf{U}} + \delta\widetilde{\mathbb{S}}\widetilde{\mathbf{U}} = 0, \tag{5.11}$$

where the matrices $\widetilde{\mathbb{M}}$, $\widetilde{\mathcal{K}}_x$, $\widetilde{\mathcal{K}}_y$ and $\widetilde{\mathbb{S}}$ are readily obtained from the elemental discretization matrices by using (5.10).

We apply the same analysis to stabilized methods. The interested reader can access all 2D dispersion plots online [86]. 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.

An example of dispersion curves is given in fig. 5.2. The picture refers to *Cubature* $\tilde{\mathbb{P}}_2$ elements, the CIP stabilization technique, and a wave angle $\theta = 5\pi/4$. We here show all 12 *parasite* modes (see fig. 5.1). The *principal* mode of this system is represented in green. This figure also shows the complexity of the analysis because of the number of modes to consider.

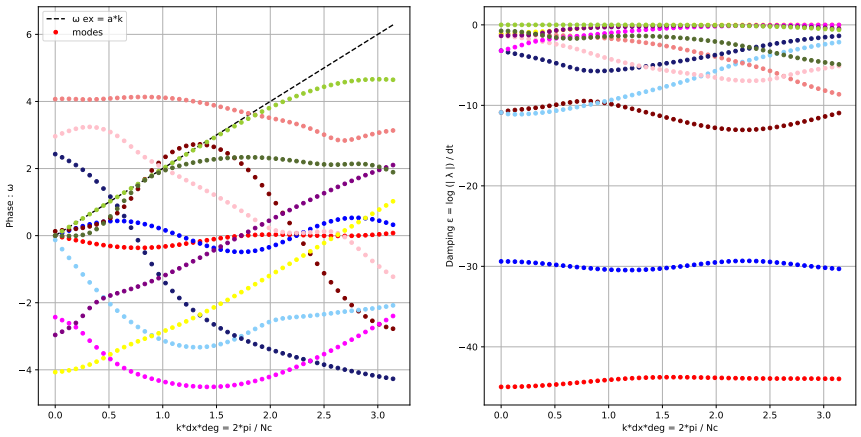


FIGURE 5.2: Dispersion curves using *Cubature* $\tilde{\mathbb{P}}_2$ elements, the CIP stabilization technique, and a wave angle $\theta = 5\pi/4$.
Phases ω (left) and amplifications ϵ (right).

We summarize the number of modes for the X mesh in 5.1. A representation of each mesh is done in E.1 for element of degree $p = 2$ and 3.

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

TABLE 5.1: X mesh: Summary table of number of modes per systems.

5.2.3 The fully discrete analysis

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

$$\begin{cases} \mathbf{U}^{(0)} := & \mathbf{U}^n \\ \mathbf{U}^{(s)} := & \sum_{j=0}^{s-1} \left(\gamma_{sj}\mathbf{U}^{(j)} + \Delta t\mu_{sj}A\mathbf{U}^{(j)}\right), \quad s \in [\![1, S]\!], \\ \mathbf{U}^{n+1} := & \mathbf{U}^{(S)}. \end{cases} \tag{5.12}$$

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

$$\mathbf{U}^{n+1} = \mathbf{U}^{(0)} + \sum_{j=1}^{S} \nu_j \Delta t^j A^j \mathbf{U}^{(0)} = \left(\mathcal{I} + \sum_{j=1}^{S} \nu_j \Delta t^j A^j\right) \mathbf{U}^n, \tag{5.13}$$

where coefficients ν_j in (5.13) are obtained as combination of coefficient γ_{sj} and μ_{sj} in (5.12) 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 (5.10) one easily shows that the problem can be written in terms of the local $d \times d$ matrices $\widetilde{A} := \widetilde{\mathbb{M}}^{-1}\left(a_x\widetilde{\mathcal{K}_x} + a_y\widetilde{\mathcal{K}_y} + \delta\widetilde{\mathbb{S}}\right)$ and in particular that

$$\widetilde{\mathbf{U}}^{n+1} = G\widetilde{\mathbf{U}}^n \quad \text{with} \quad G := \left(\widetilde{\mathcal{I}} + \sum_{j=1}^{S} \nu_j \Delta t^j \widetilde{A}^j\right) = e^{\epsilon\Delta t}e^{-i\omega\Delta t}, \tag{5.14}$$

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

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

For the DeC method we can proceed with the same analysis transforming also the other involved matrices into their Fourier equivalent ones. Using (3.52) these terms would contribute to the construction of G not only in the $\widetilde{A}$ matrix, but also in the coefficients ν_j, which become matrices as well. At the end we just study the final matrix G and its eigenstructure, whatever process was needed to build it up.

The matrix G describes one timestep evolution of the Fourier modes for all the d different types of degrees of freedom. The damping coefficients ϵ_i tell if the modes are increasing or decreasing in amplitude and the phase coefficients ω_i describe the phases of such modes. We remark that a necessary condition for stability of the scheme is that $|\lambda_i| \leq 1$ or, equivalently, $\epsilon_i \leq 0$ for all the eigenvalues. The goal of our study is to find the largest CFL number for which the stability condition is fulfilled and such that the dispersion error is *not too large*.

For our analysis, we focus on the X type triangular mesh in fig. 5.1 with elements of degree 1, 2 and 3. This X type triangular mesh is also used in [81] for Fourier analysis of the acoustic wave propagation system.

5.2.4 Methodology

The methodology we explain in the following, will be applied to all the combination of schemes we presented above (in time: RK, SSPRK and DeC, discretisation in space: *Basic*, *Cubature* and *Bernstein* elements, stabilization techniques: CIP, OSS and SUPG), in order to find the best coefficients (CFL, δ), as in [88].

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

What we aim to do is an optimization process also on the stabilization parameter and the CFL number. With the notation of [88], we will set for the different stabilizations

$$\begin{aligned} \text{OSS}: &\ \tau_K = \delta \Delta x |a|, \\ \text{CIP}: &\ \tau_f = \delta \Delta x^2 |a|, \\ \text{SUPG}: &\ \tau_K = \delta \Delta x / |a|. \end{aligned}$$

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

In the following we describe the strategy we adopt to find the best parameters couple (CFL,δ) that minimizes a global solution error, denoted by η_u, while maximizing the CFL in the stable area. In particular, we use the relative square error of u defined in the previous chapter (eq. (4.32)). Moreover, we need to check that the stability condition holds for all the possible angles $\Phi \in [0, 2\pi]$. We seek for the couple (CFL^*, δ^*) such that

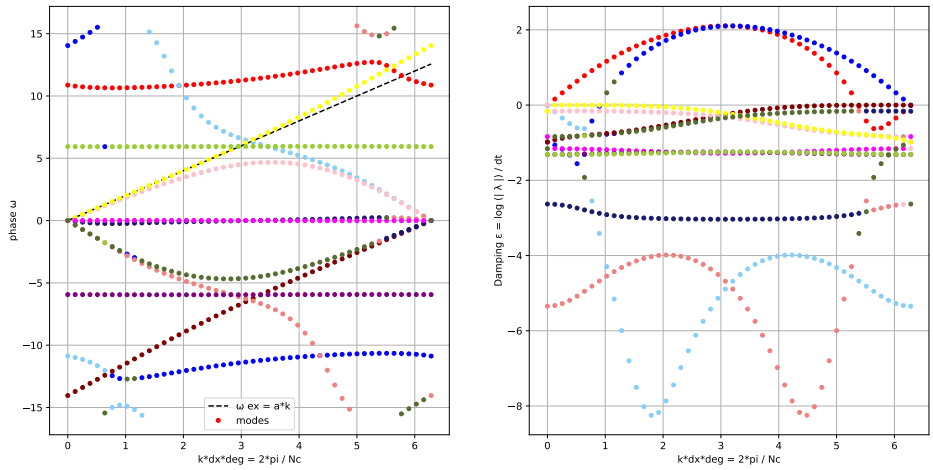
$$(\text{CFL}^*, \delta^*) = \arg\max_{\text{CFL}} \left\{ \eta(\omega, \epsilon, \Phi') < \mu \min_{\text{stable } (\text{CFL},\delta)} \max_{\Phi} \eta(\omega, \epsilon, \Phi), \quad \forall\, \Phi' \in [0, 2\pi] \right\}, \tag{5.15}$$

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

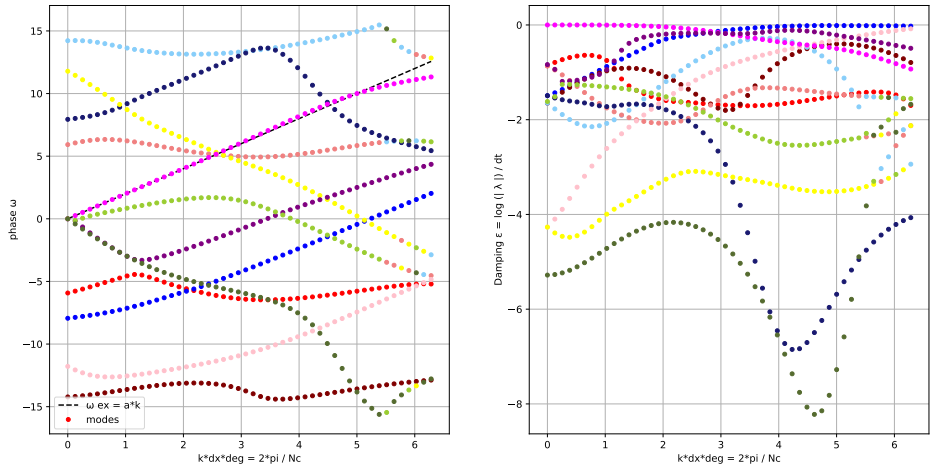
To show the influence of the angle Φ on the optimization problem we show an example for the X mesh. For a given couple of parameters (CFL,δ) = $(0.4, 0.01)$ we compare the results for $\Phi = 0$ and $\Phi = 3\pi/16$. In fig. 5.3 we compare the phases ω_i and the damping coefficients ϵ_i for the two angles. It is clear that for the angle $\Phi = 0$, on the left, there are some modes which are not stable $\epsilon_i > 0$, while for $\Phi = 3\pi/16$ all modes are stable.

The angle can widely influence the whole analysis as one can observe in the plot of $\max_i \epsilon_i$ in fig. 5.4, where we observe that for the only angle $\Phi = 3\pi/16$ we would obtain an optimal parameter in (CFL,δ) = $(0.4, 0.01)$, while, using all angles, this value is not stable anymore.

Remark: In theory, we should see only machine precision damping coefficients inside the stable area, but, since the eigenvalue problem we need to solve in eq. (5.14) is only approximated thanks to the linear algebra package of `numpy`, the results might be not so accurate.



(a) $\Phi = 0$



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

FIGURE 5.3: Comparison of dispersion curves ω_i and damping coefficients ϵ_i, for *Cubature* $\tilde{\mathbb{P}}_2$ elements, with SSPRK time discretization and OSS stabilization: $\Phi = 0$ and $\Phi = 3\pi/16$.

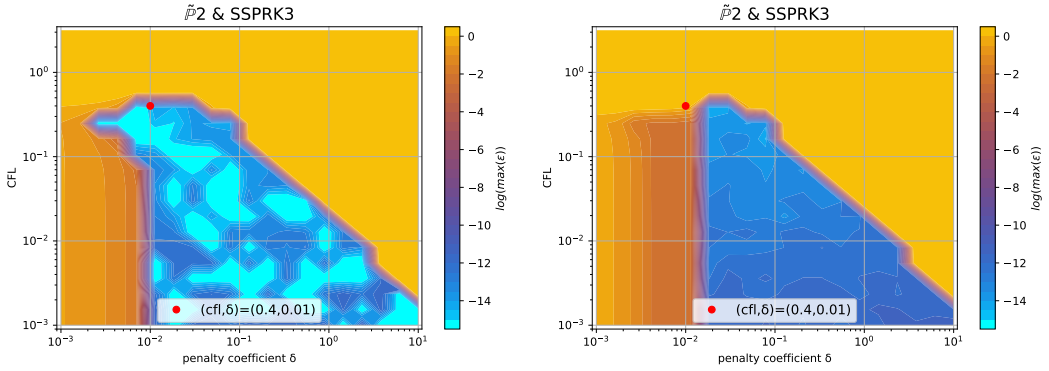


FIGURE 5.4: Plot of $\log(\max_i \epsilon_i)$ for *Cubature* $\tilde{\mathbb{P}}_2$ elements, SSPRK time discretization and OSS stabilization. The blue and light blue region is the stable one. At the left only for $\Phi = 3\pi/16$, at the right we plot the maximum over all Φ.

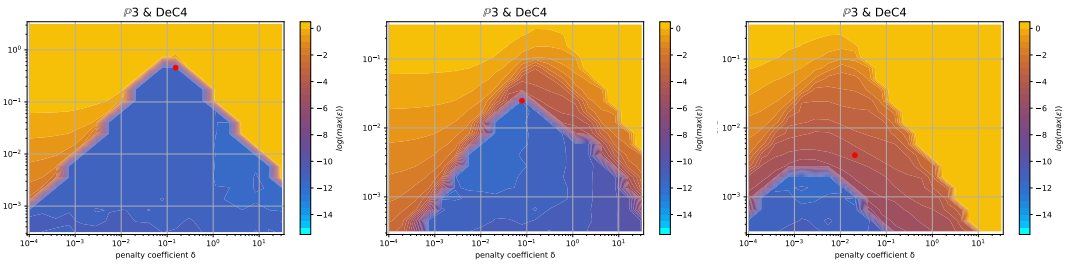


FIGURE 5.5: Damping coefficients $\log(\max_i \epsilon_i)$ for $\mathbb{B}_3$ *Bernstein* elements and the DeC method with, from left to right, SUPG, OSS and CIP stabilization.

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

In this section, we illustrate the result obtained with the methodology explained above. For clarity not all the results are reported in this chapter, however, we place all the plots for all possible combination of schemes in an online repository [86]. We will provide some examples here and a summary of the main results that we obtained.

The first type of plot we introduce is helping us in understanding how we can define the stability region in the (CFL, δ) plane. So, for every (CFL, δ) we plot the maximum of $\log(\epsilon_i)$ over all modes and angles $\Phi \in [0, 2\pi]$ (thanks to the symmetry of the mesh we can reduce this interval). An example is given in the right plot of fig. 5.4, it is clear that the whole blue area is stable and the yellow/orange area is unstable. In other cases, this boundary is not so clear and setting a threshold to determine the stable area can be challenging. In fig. 5.5 we compare different stabilizations for DeC with $\mathbb{B}_3$ elements. In the CIP stabilization case, we clearly see that there is no clear discontinuity between unstable values and stable ones, as in SUPG, because there is a transient region where $\max_i \epsilon_i$ varies between 10^{-7} and 10^{-4}.

The second type of plot combines the chosen stability region with the error η_u. We plot on the (CFL, δ) plane some black crosses on the unstable region, where there exists an i and Φ such that $\epsilon_i > 10^{-7}$. The color represents $\log(\eta_u)$ and the best value according to the previously described method is marked with a red dot. In figs. 5.6 to 5.9 we show some examples of these plots for some schemes, for different $p = 1, 2, 3$. In figs. 5.6 and 5.7 we test the *Basic* elements with the SSPRK time discretization, while in figs. 5.8 and 5.9 we use the *Cubature* elements with DeC time discretization. We compare also different stabilization technique: in figs. 5.6 and 5.8 we use the OSS, while in figs. 5.7 and 5.9 the CIP. One can observe many differences among the schemes. For instance, for $p = 3$ we see a much wider stable area for SSPRK than with DeC and, in the *Cubature* DeC case, we see that the CIP requires a reduction in the CFL number with respect to the OSS stabilization.

We summarize the results obtained by the optimization strategy in table 5.2 for all the combinations of spatial, time and stabilization discretization. The CFL and δ presented there are optimal values obtained by the process above described, which we aim to use in simulations to obtain stable and efficient schemes. Unfortunately, as already mentioned above, for some schemes the stability area is not so well defined for several reasons. One of these reasons is the *"shape"* of the stability area as for one-dimensional problems, see [88]. Other issues that affect this analysis are the numerical precision, see section 5.2.6, and the

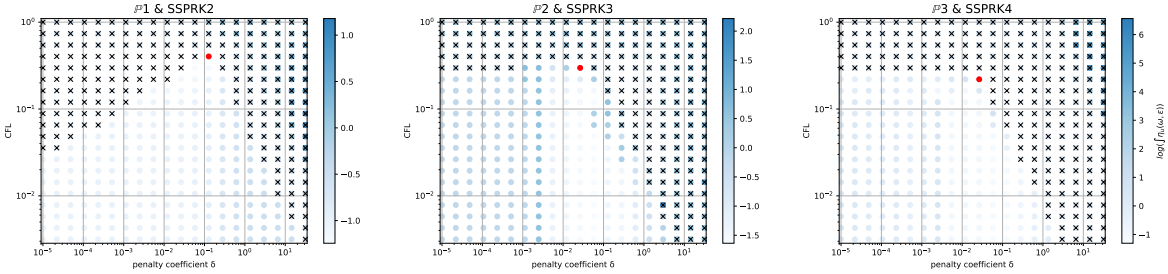


FIGURE 5.6: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ *Basic* elements with SSPRK scheme and OSS stabilization

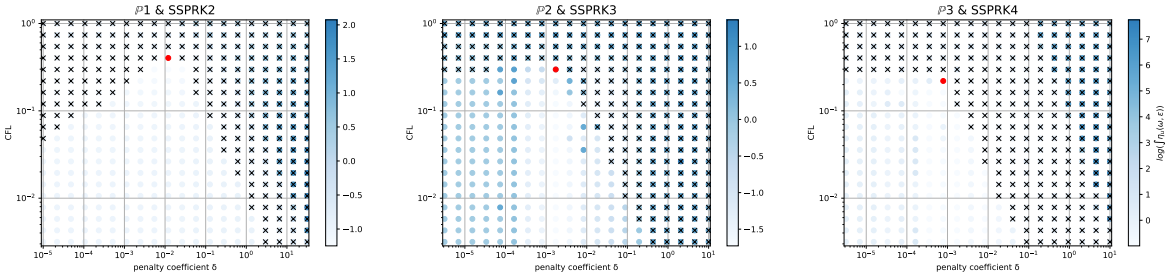


FIGURE 5.7: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ *Basic* elements with SSPRK scheme and CIP stabilization

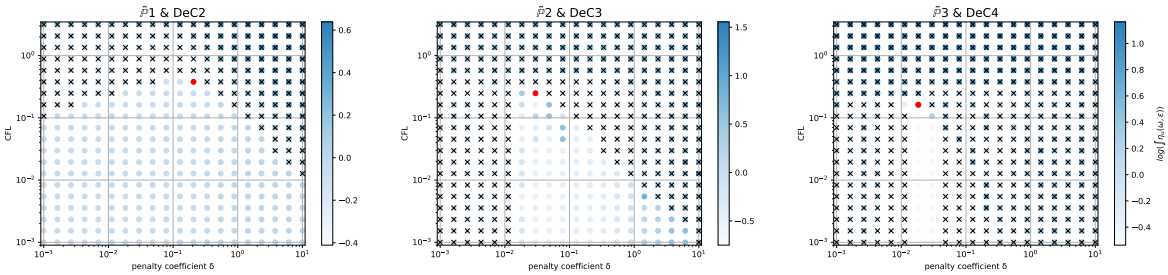


FIGURE 5.8: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ *Cubature* elements with DeC scheme and OSS stabilization

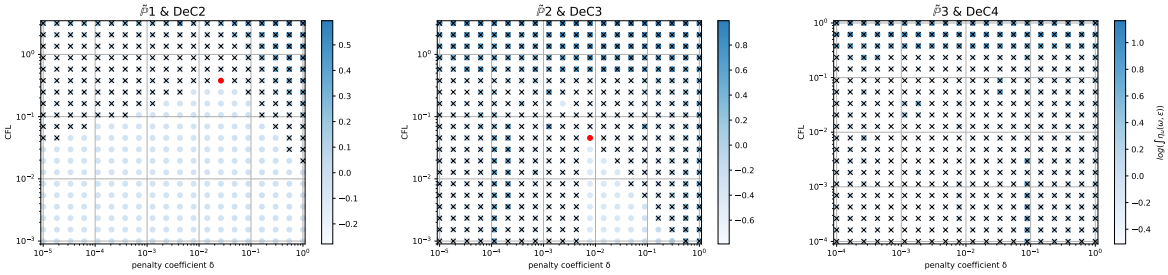


FIGURE 5.9: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ *Cubature* elements with DeC scheme and CIP stabilization

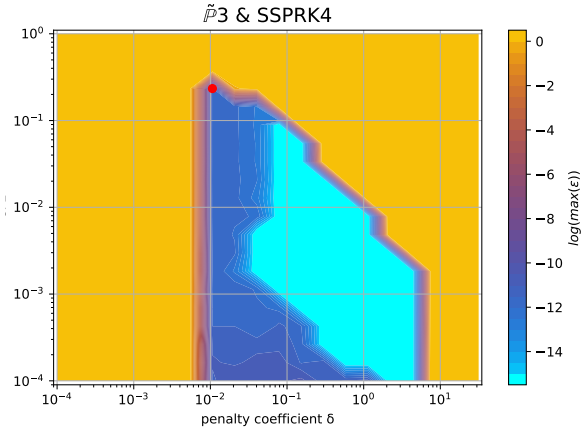


FIGURE 5.10: Logarithm of the amplification coefficient $\log(\max_i(\varepsilon_i))$ for SUPG stabilization with $\tilde{\mathbb{P}}_3$ *Cubature* elements and the SSPRK method. Unstable region in yellow, optimal (5.15) parameters in red

mesh configuration, see section 5.2.7. In the following we study more in details these cases and how one can find better values.

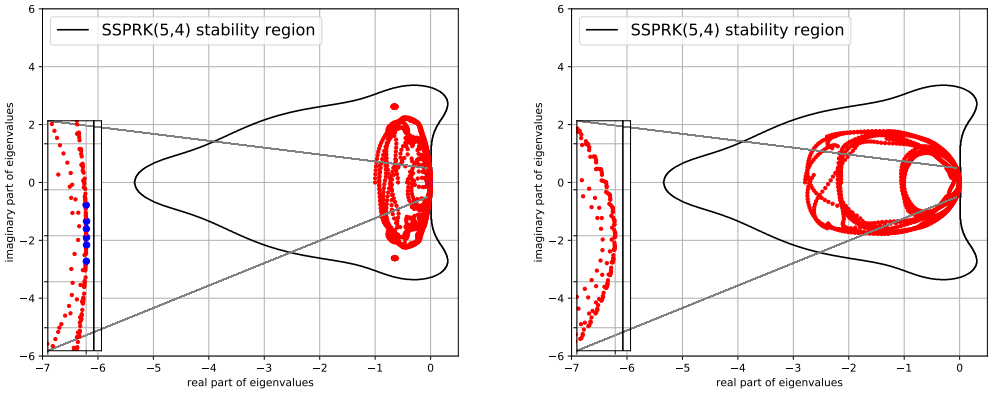
Element &		SUPG		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	0.739 (0.127)	0.298 (0.058)	0.22 (0.026)
	RK	0.403 (0.127)	0.298 (0.026)	0.22 (5.46e-03)
Cub.	DeC	0.616 (0.28)	0.234 (0.04)*	0.144 (0.04)
	SSPRK	1.062 (0.28)	0.379 (0.021)*	0.234 (0.011)*
	RK	0.616 (0.28)	0.234 (0.04)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

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

TABLE 5.2: X mesh: Optimized CFL and penalty coefficient δ in parenthesis, minimizing η_u.
"/" means that the fourier analysis shown that the scheme is unstable.
* These values are not reliable, see section 5.2.6.

5.2.6 Comparison with a space-time split stability analysis

In this section, we show another stability analysis to slightly improve the results obtained above. Indeed, the solution of the eigenvalue problem (5.14) is computed with numerical



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

FIGURE 5.11: Eigenvalues of $\widetilde{A}$ using *Cubature* discretization and the SUPG stabilization (varying k) and stability area of the SSPRK method. In red the stable eigenvalues, in blue the unstable ones.

solver and it might bring numerical error. In order to have a different result to compare, we can decompose the time integration and the spatial discretization for methods which use the method of line (not DeC). In this way, we find the eigenvalues only of the spatial discretizations, and then we check whether they belong to the stability area of the time discretization.

As an example, we explore the case of *Cubature* $\tilde{\mathbb{P}}_3$ elements, using SSPRK method and the SUPG stabilization technique of which the previously presented analysis results in the plot of fig. 5.10. The red point represents the optimal parameters (CFL,δ), in the sense of eq. (5.15). The optimal value is surrounded by unstable schemes, hence, it is not so safe to use the parameters found. Intuitively one should try to decrease the CFL and increase δ, in order to move the parameters in a safer region.

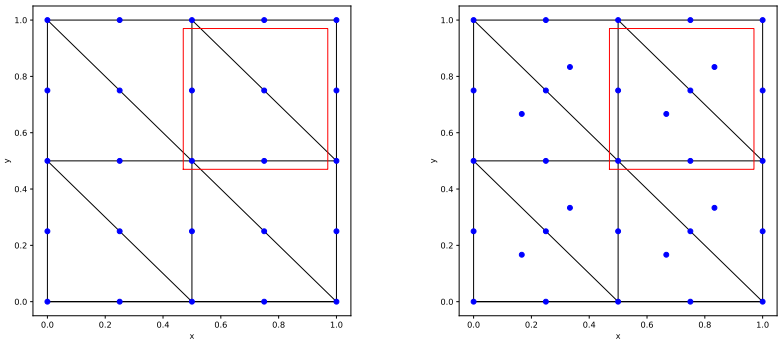
Following [29], we write the time discretization for Dahlquist's equation

$$\partial_t u - \lambda u = 0, \tag{5.16}$$

in this example, we consider the SSPRK discretization eq. (5.12). From eq. (5.13) we can write the amplification coefficient $\Gamma(\lambda)$, i.e.,

$$\mathbf{U}^{n+1} = \mathbf{U}^{(0)} + \sum_{j=1}^{S} \nu_j \Delta t^j \lambda^j \mathbf{U}^{(0)} = \underbrace{\left(\mathcal{I} + \sum_{j=1}^{S} \nu_j \Delta t^j \lambda^j \right)}_{\Gamma(\lambda)} \mathbf{U}^n. \tag{5.17}$$

The stability condition for this SSPRK scheme is given by $\Gamma(\lambda) \leq 1$. Now, when we substitute the Fourier transform of the spatial semidiscretization $\widetilde{A}$ to the coefficient λ and we diagonalize the system (or we put it in Jordan's form), we obtain a condition on the eigenvalues of $\widetilde{A}$. Then, using the parameters provided by the previous analysis (CFL,δ)=$(0.234, 0.011)$, in table 5.2, we plot the eigenvalues of $\widetilde{A}$ and the stability region of the SSPRK scheme for different $\theta \in [0, \pi]$. We notice that for some values of θ, not all the eigenvalues belong to the stable area, see fig. 5.11(a). There are, indeed, few eigenvalues dangerously close to the imaginary axis and some of them have actually positive real part (blue dots). As suggested



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

FIGURE 5.12: The T type triangular mesh with degrees of freedom in blue and periodic unit in the red square for the Fourier analysis

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

TABLE 5.3: Number of modes in the periodic unit for different elements in the T mesh

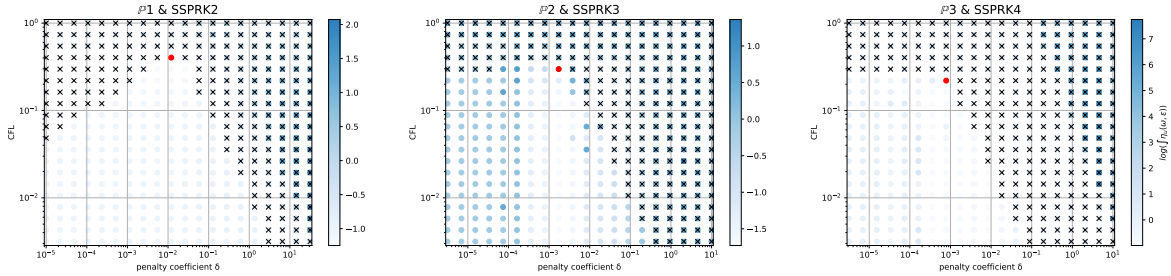
before, if we decrease the CFL and increase δ, we move towards a safer region, so considering (CFL,δ)=$(0.18, 0.04)$ with the same θ, we obtain all stable eigenvalues, as shown in fig. 5.11(b).

The summary of the optimal parameters of table 5.2 updated taking into account also a larger safety region in the (CFL, δ) plane (as explained in this section) can be found in table E.1 in Appendix E.2.

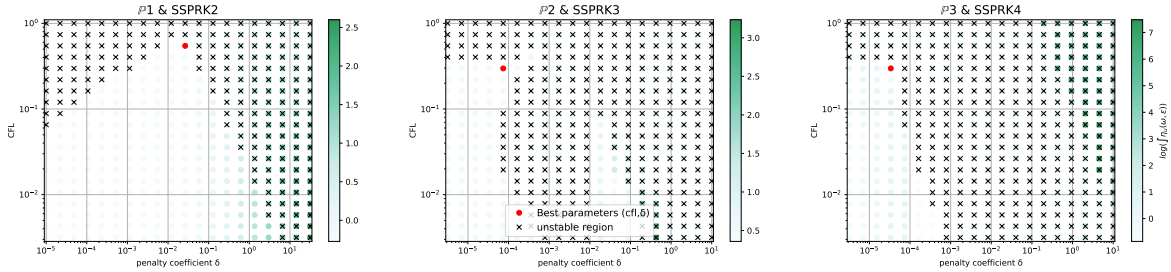
5.2.7 Different mesh patterns

Another important aspect about this stability analysis is the influence of the mesh structure on the results. As an example, we introduce another regular and structured mesh type that we denote by T mesh depicted in fig. 5.12. In fig. 5.12 we plot also the degrees of freedom for elements of degree 2 and the periodic elementary unit that we take into consideration for the Fourier analysis. The number of modes in the periodic unit for this mesh type are summarized in table 5.3. The elements of degree 3 can be found in fig. E.1 in Appendix E.1.

Even if for several methods we observe comparable results for the two mesh types, for some of them the analyses are quite different. An example is given by the *Basic* elements with SSPRK schemes and CIP stabilization. For this method, we plot the dispersion error eq. (4.36) and the stability area in fig. 5.13(a) for the X mesh and in fig. 5.13(b) for the T mesh. We see huge differences in $\mathbb{P}_2$ and $\mathbb{P}_3$ where in the former a wide region becomes unstable for $\delta_L \leq \delta \leq \delta_R$ and for the latter we have to decrease a lot the value of δ to obtain stable schemes.



(a) X mesh



(b) T mesh

FIGURE 5.13: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ *Basic* elements with SSPRK scheme and CIP stabilization

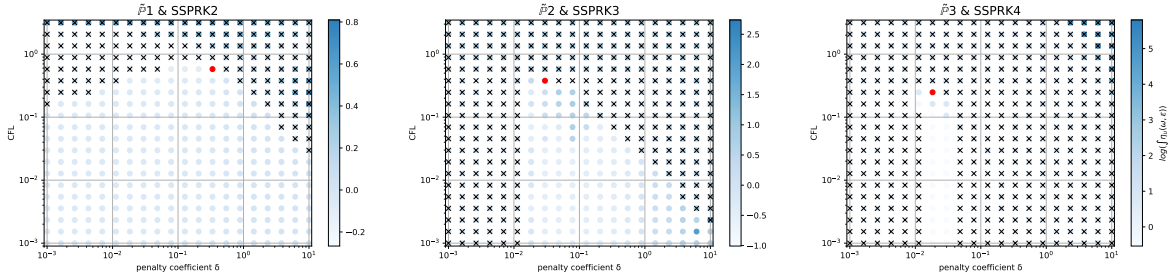
In the case of *Cubature* elements with the OSS stabilization and SSPRK time integration, we have already seen in the previous section that the optimal parameters found were in a dangerous area. Repeating the stability analysis for the T mesh we see that the situation is even more complicated. In fig. 5.14(a) we plot the analysis for the X mesh and in fig. 5.14(b) the one for the T mesh. $\tilde{\mathbb{P}}_3$ elements, though being stable for some parameters for the X mesh, are never stable on the T mesh. This means, that, when searching general parameters for the schemes, we have to keep in mind that different meshes leads to different results.

For completeness, we present the optimal parameters also for the T mesh in table E.2 in Appendix E.2.

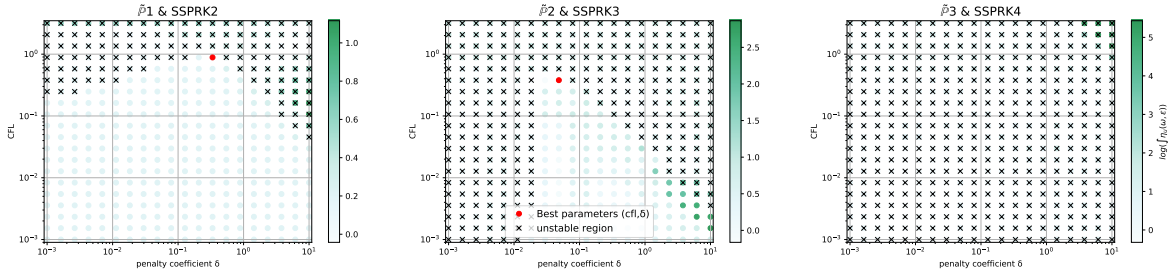
In general, it is important to consider more mesh types when doing this analysis. In practice, we will use the two presented above (X and T meshes). In the following, we will consider the stability region as the intersection of stability regions of both meshes.

5.2.8 Final results of the stability analysis

Taking into consideration all the aspects seen in the previous sections, it is important to have a comprehensive result, which tells which parameters can be used in the majority of the situations. A summary of the parameters obtained for the X and T mesh is available in Appendix E.2. In table 5.4, instead, we present parameters obtained using the most restrictive case among different meshes and that insure an enough big area of stability around them, as explained in section 5.2.6. These parameters can be safely used in many cases and we will validate them in the numerical sections, where, first, we validate the results of the X mesh

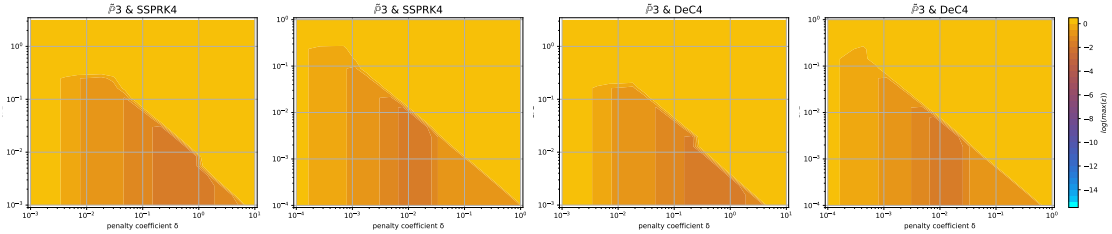


(a) X mesh



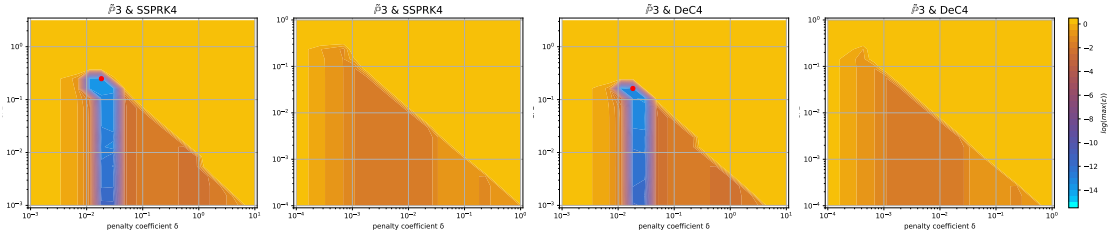
(b) T mesh

FIGURE 5.14: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ *Cubature* elements with SSPRK scheme and OSS stabilization



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

FIGURE 5.15: Maximum logarithm of the amplification coefficient $\log(\max_i(\varepsilon_i))$ for $\tilde{\mathbb{P}}_3$ *Cubature* elements on the X and T meshes



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

FIGURE 5.16: Logarithm of the amplification coefficient $\log(\max_i(\varepsilon_i))$ for $\tilde{\mathbb{P}}_3$ *Cubature* elements on the X mesh

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

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

TABLE 5.4: Optimized CFL and penalty coefficient δ in parenthesis, combining the two mesh configurations. The values denoted by * are not the optimal one, but they lay in a safer region, see Section 5.2.6. The values marked by ** cannot be used on the T mesh. "/" means that it is unstable for every parameter

on a linear problem on an X mesh, then we used the more general parameters in table 5.4 for nonlinear problems on unstructured meshes.

A special remark must be done for *Cubature* $\tilde{\mathbb{P}}_3$ elements used with the OSS and the CIP stabilizations. In fig. 5.15 we see how the amplification coefficient $\max_i \varepsilon_i$ has always values far away from zero. For the CIP stabilization this is always true and even for the $\tilde{\mathbb{P}}_2$ elements the stability region is very thin. As suggested in [26, 76] higher order derivatives jump stabilization terms might fix this problem, but it introduces more parameters. This has not been considered here. Another remark is that the T configuration is very peculiar and, as we will see, on classical Delaunay triangulations the issue seem to not affect the results. Finally, the use of additional discontinuity capturing operators may alleviate this issue as some additional, albeit small, dissipation is explicitly introduced in smooth regions.

In section 5.2.9, we propose to add an additional stabilization term for these unstable schemes, i.e., *Cubature* $\tilde{\mathbb{P}}_3$ elements and OSS or CIP stabilization techniques. This term is based on viscous term [1, 58, 74, 83] and allows to stabilize numerical schemes for any mesh configuration.

For the OSS stabilization we observe a similar behavior in fig. 5.15. The stability that we see in that plot are only due to the the T mesh. Indeed, for the OSS stabilization on the X mesh there exists a corridor of stable values, which turn out to be unstable for the T mesh, see fig. 5.16. In practice, on unstructured grids we have not noticed instabilities when running with the parameters found with the X mesh. Hence, we suggest anyway some values of CFL and δ for these schemes, which are valid for the X mesh, noting that they might be dangerous for very simple structured meshes. The validation on unstructured meshes also for more complicated problems will be done in the next sections.

Overall, table 5.4 gives some insight on the efficiency of the schemes. We remind that, in general, we prefer matrix free schemes, so this aspect must be kept in mind while evaluating the efficiency of the schemes. All the SUPG schemes, except when with DeC, and all the basic element schemes have a mass matrix that must be inverted. Among the others we see that for first degree polynomials schemes the DeC with Bernstein polynomials and SUPG

stabilization gives one of the largest CFL result, while for second degree polynomials the OSS *Cubature* SSPRK scheme seems the one with best performance and, for fourth order schemes, again the *Bernstein* elements combined with the DeC and SUPG is one of the best.

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

- The extension of the Fourier analysis to the two-dimensional space leads to significantly different results with respect to the one-dimensional one. Both in terms of global stability of the schemes, and in terms of optimal parameters. Moreover, in opposition to [88], *Bernstein* elements with SUPG stabilization technique lead to stable and efficient schemes. *Cubature* elements, which were the most efficient in one-dimensional problems, have stability issues on the 2D mesh topologies studied.

- The complexity of the analysis in two-dimensional space is increased. This not only implies a larger number of degrees of freedom, but also more parameters to keep everything into account, including the angle of the advection term and the possible different configuration of the mesh. The visualization of the stability region of the time scheme as shown in fig. 5.11 with the eigenvalues of the semidiscretization operators helps in understanding the effect of CFL and penalty coefficient on the stability of the scheme, only for methods of lines. This helps in choosing and optimizing the couple of parameters.

Remark 5.2.1 *Another possibility to characterize the linear stability of numerical method is proposed by J. Miller [89]. This method is based on the study of the characteristic polynomial of the amplification matrix G. However, this method does not provide information about the phase ω, since it does not compute eigenvalues of G. For this reason, we choose the eigenanalysis.*

5.2.9 Complementary analysis using viscosity terms

To ensure the stability of numerical scheme also in the configurations for which the stabilization techniques are not stable, we can introduce a further viscosity term (e.g. for non-linear term based on the local entropy production [1, 71] for discontinuous Galerkin formulation, [58, 74, 83] for continuous Galerkin formulation and references therein). The variational formulation reads: for any $v_h \in W_h$, find $u_h \in V_h^p$ that satisfies

$$\int_\Omega v_h(\partial_t u_h + \nabla \cdot f(u_h))dx + S(v_h, u_h) + \underbrace{\sum_K \int_K \mu_K(u_h)\nabla v_h \cdot \nabla u_h}_{\text{Viscosity term}} = 0, \tag{5.18}$$

where $S(v_h, u_h)$ corresponds to stabilization technique described in section 3.2 and $\mu_K = \mathcal{O}(h^{p+1}) > 0$ is the viscosity parameter for $K \in \Omega_h$.

Note on the stability of the method

As it is done for previous stabilization terms in section 3.2, we can characterize the accuracy of this method estimating the truncation error for a polynomial approximation of degree p. Considering the smooth exact solution $u^e(t,x)$ of (5.18), for all functions ψ of

class at least $\mathcal{C}^1(\Omega)$ of which ψ_h denotes the finite element projection, we obtain

$$\epsilon(\psi_h) := \Big| \int_{\Omega_h} \psi_h \partial_t (u_h^e - u^e)\, dx - \int_{\Omega_h} \nabla\psi_h \cdot (f(u_h^e) - f(u^e))\, dx + \sum_{K\in\Omega_h} \mu_K \int_K \nabla\psi_h \cdot \nabla(u_h^e - u^e) \Big| \leq Ch^{p+1}, \tag{5.19}$$

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

Then, for a linear flux, periodic boundaries and taking $\mu_K = \mu$ constant along the mesh, we can test with $v_h = u_h$ in (5.18), we get

$$\int_{\Omega_h} d_t \frac{u_h^2}{2} = -\sum_K \int_K \mu(\nabla u_h)^2 \leq 0, \tag{5.20}$$

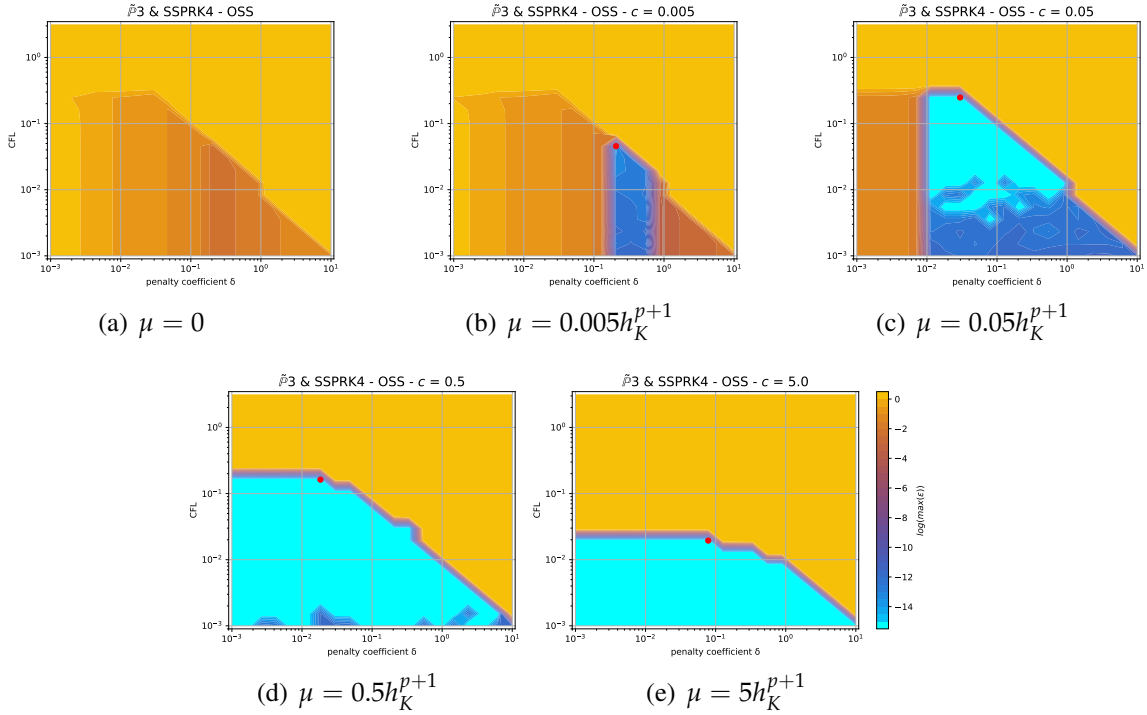
which can be integrated in time to obtain a bound on the $\mathbb{L}_2$ norm of the solution.

The von Neumann analysis

As we saw in section 5.2.8, the T mesh configuration has stability issues. In particular, the numerical schemes using *Cubature* $\tilde{\mathbb{P}}_3$ elements, SSPRK and DeC time integration methods, and the OSS and the CIP stabilization techniques are unstable. We propose to evaluate these schemes adding the viscosity term in (5.18). For the von Neumann analysis, we use $\mu_K(u) = ch_K^{p+1}$ in (5.18), with $c \in \mathbb{R}^+$, h_K the cell diameter and p the degree of polynomial approximation. We show the plot of $\max_i \epsilon_i$ to understand how the stability region behaves with respect to c using *Cubature* $\tilde{\mathbb{P}}_3$ elements. In fig. 5.17 the maximum amplification factor ϵ is represented for varying c, using the OSS stabilization technique and the SSPRK time integration method. We note that the same behaviour is observed with CIP and DeC. Plots are available online [86].

We can observe two main results. First, increasing the parameter c up to around 0.1 allows to expand the stability region. Second, when the viscosity coefficients reaches too high values, it is necessary to decrease the CFL (see fig. 5.17(c) with $\mu = 0.05$ and fig. 5.17(d) with $\mu = 0.5$ as an example).

Remark 5.2.2 (Stability of entropy viscosity) *In our formulation the coefficient $\mu_K = ch^{p+1}$ leads to nongeneralizable conclusions and, hence, it is not recommended for general cases. Indeed, with this formulation, the amplification matrix G depends on CFL, on c, on δ, and also on h^p. This means that changing h, keeping all the other coefficients fixed might lead to unstable schemes. A more reliable formulation, which can be used for general cases, is given in J.L. Guermond et al. [58]. The viscous term is non-linear and it is based on the local entropy production.*



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

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

FIGURE 5.17: T mesh - Von Neumann analysis using an additional viscosity term (see (5.18)). *Cubature* $\tilde{\mathbb{P}}_3$ elements with SSPRK and OSS. Comparison of different μ.

5.3 Validation of the fourier analysis

We now perform numerical tests to check the validity of our theoretical findings using the X mesh configuration. We will use elements of degree p, with p up to 3, with time integration schemes of the corresponding order of accuracy to ensure an overall error of $\mathcal{O}(\Delta x^{p+1})$, under the CFL conditions discussed earlier and available in table E.1 in appendix E.2. The integral formulas 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* elements we use Gauss–Legendre quadrature formulas.

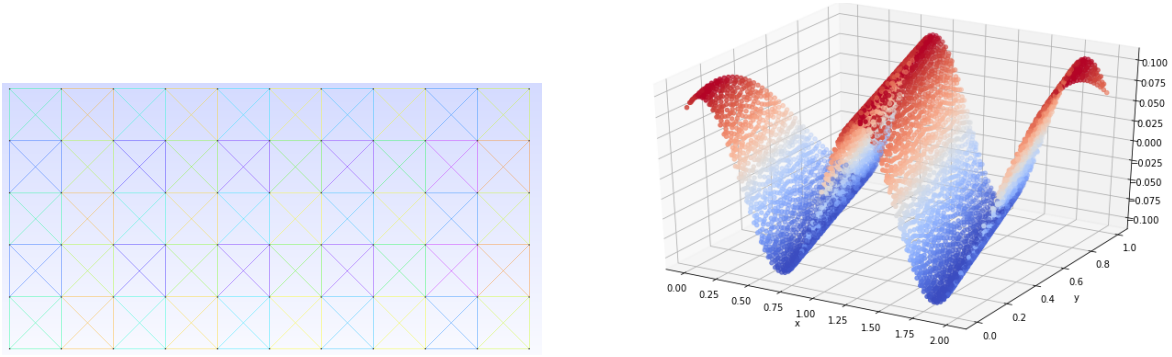
The mesh used in the Fourier analysis is the basis of the one we will use in the numerical simulations. We will extend it periodically for the whole domain, see an example in fig. 5.18(a).

5.3.1 Linear advection equation test

We start with the linear advection eq. (5.1) on the domain $\Omega = [0,2] \times [0,1]$ using Dirichlet boundary conditions:

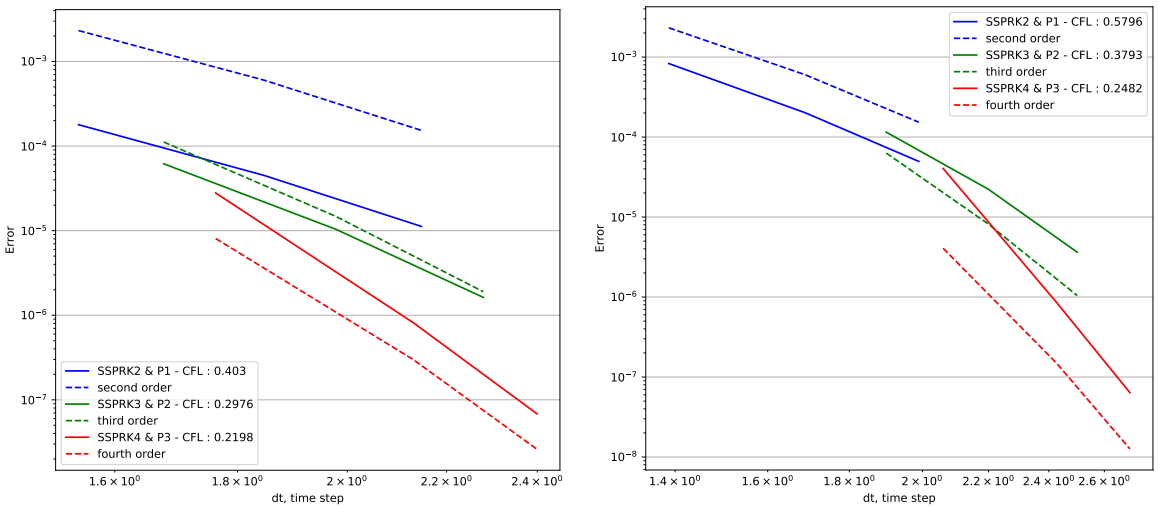
$$\begin{cases} \partial_t u(t,\mathbf{x}) + \mathbf{a} \cdot \nabla u(t,\mathbf{x}) = 0, & (t,\mathbf{x}) \in [t_0, t_f] \times \Omega, \quad \mathbf{a} = (a_x, a_y)^T \in \mathbb{R}^2, \\ u(0,\mathbf{x}) = u_0(\mathbf{x}), & \\ u(t,\mathbf{x_D}) = u_{ex}(t,\mathbf{x_D}), & \mathbf{x_D} \in \Gamma_D = \{(x,y) \in \mathbb{R}^2, x \in \{0,2\} \text{ or } y \in \{0,1\}\}, \end{cases} \tag{5.21}$$

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



(a) X mesh on $\Omega = (0,2) \times (0,1)$ (b) Cosinus test case with $\theta = 3\pi/16$

FIGURE 5.18: Linear advection simulation on the X mesh



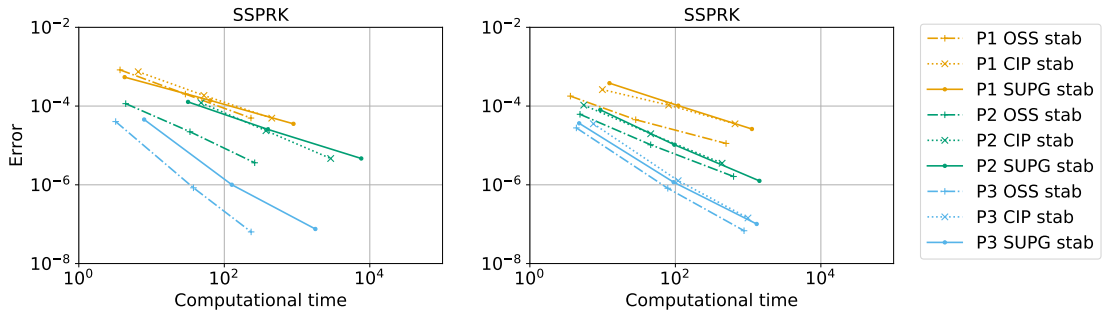
(a) *Basic* elements. (b) *Cubature* elements.

FIGURE 5.19: Error decay for linear advection problem with different elements and OSS stabilization and SSPRK time discretization: $\mathbb{P}_1$ in blue, $\mathbb{P}_2$ in green and $\mathbb{P}_3$ in red

of the simulation is $t_f = 2s$. Clearly the exact solution is $u_{ex}(\mathbf{x},t) = u_0(x - a_x t, y - a_y t)$ for all $\mathbf{x} = (x,y) \in \Omega$ and $t \in \mathbb{R}^+$. The initial conditions are displayed in fig. 5.18(b). We discretize the domain with uniform squares with edge length Δx and then in each square we reproduce the X mesh pattern, see fig. 5.18(a). In particular, we will use different Δx to test the convergence for each order of accuracy: $\Delta x_1 = \{0.1, 0.05, 0.025\}$ 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 guarantees to have approximately the same number of degrees of freedom for different p.

A representative result is provided in figs. 5.19(a) and 5.19(b): it shows a comparison between *Cubature* and *Basic* elements with OSS stabilization and SSPRK time integration. As we can see, the two schemes have very similar errors except for $\mathbb{P}^1$ where the larger CFL increases the error. The *Basic* elements require stricter CFL conditions, see table E.1, and have larger computational costs because of the inversion of the mass matrix.

To show the main benefit of using the *Cubature* elements (diagonal mass matrix), we plot in fig. 5.20 the computational time of *Basic* and *Cubature* elements for the SSPRK



(a) *Cubature* elements (b) *Basic* elements

FIGURE 5.20: Error for linear advection problem (5.21) with respect to computational time for SSPRK time discretization, comparing *Basic* and *Cubature* elements and all stabilization techniques

time scheme and all stabilization techniques. As a first interesting result of numerical test, looking at the fig. 5.20, we can clearly see that, for a fixed accuracy, *Cubature* elements obtain better computational times with respect to *Basic* elements. Moreover, as expected, the SUPG stabilization technique requires more computational time as it requires the inversion of a mass matrix, even in the case where the CFL used in is larger than the ones for OSS or CIP stabilization, see table E.1.

The order of accuracy reached by each simulations is shown in table 5.5. The plots and all the errors are available at the repository [86].

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

TABLE 5.5: Convergence order for all schemes on linear advection test, using coefficients obtained in table E.1.
"/" means that the Fourier analysis showed that the scheme is unstable.

Looking at the table 5.5, we observe that almost all the stabilized schemes provide the expected order of accuracy. Exception to this rule are several $\mathbb{P}_2$ discretization which reach an order of accuracy of ≈ 2.5, and all *Bernstein* $\mathbb{B}_3$ elements with the *DeC* which reach an order of accuracy of 2. This result is very disappointing and it does not improve even adding more corrections, as suggested in [107, 2]. We will show in section 5.4.3 that the previous problem does not appear in a steady case, which opens the door to further research.
Note that we do not show results for *Bernstein* elements with *SSPRK* technique because they are identical to *Basic* elements, but are more expensive because of the projection in the *Bernstein* element space and the interpolation in the quadrature points.
More comparisons on different grids (unstructured) will be done in section 5.4.

5.3.2 Shallow Water equations

We consider the non linear Shallow Water equations (no friction and constant topography):

$$\begin{cases} \partial_t h + \partial_x(hu) + \partial_y(hv) & = 0, \\ \partial_t(hu) + \partial_x(hu^2 + g\frac{h^2}{2}) + \partial_y(huv) & = 0, \\ \partial_t(hv) + \partial_x(huv) + \partial_y(hv^2 + g\frac{h^2}{2}) & = 0, \end{cases} \quad \begin{array}{l} x \in \Omega = [0,2] \times [0,1], \\ t \in [0, t_f] \\ t_f = 1s. \end{array} \tag{5.22}$$

An analytical solution of this system is given by travelling vortexes [85]. We use here a vortex with compact support and in $\mathcal{C}^6(\Omega)$ described by the following parameters:

$$\begin{cases} \mathbf{X_c} = (0.5, 0.5) & \text{the center of the vortex at t=0s} \\ r_0 = 0.45 & \text{radius of the vortex,} \\ \Delta h = 0.1 & \text{amplitude of the vortex,} \\ \omega = \pi / r_0 & \text{angular wave frequency,} \\ \Gamma = \frac{12\pi\sqrt{g\Delta h}}{r_0\sqrt{315\pi^2 - 2048}} & \text{vortex intensity parameter,} \\ h_c = 1. & \text{stady state,} \\ u_c = 0.6 & \text{speed in } x \text{ direction,} \\ v_c = 0. & \text{speed in } y \text{ direction,} \\ \mathcal{I}(\mathbf{x}, t) = \mathbf{x} - \mathbf{X_c} - (u_c t, v_c t)^T & \text{coordinates with respect to the vortex center,} \\ \mathcal{R}(\mathbf{x}, t) = \|\mathcal{I}(\mathbf{x}, t)\| & \text{distance from the vortex center.} \end{cases} \tag{5.23}$$

The analytical solution is written as

$$\begin{pmatrix} h(x,t) \\ u(x,t) \\ v(x,t) \end{pmatrix} = \begin{cases} \begin{pmatrix} h_c + \frac{1}{g}\frac{\Gamma^2}{\omega^2} \times (\lambda(\omega\mathcal{R}(\mathbf{x},t)) - \lambda(\pi)), \\ u_c + \Gamma(1 + \cos(\omega\mathcal{R}(\mathbf{x},t)))^2 \times (-\mathcal{I}(\mathbf{x},t)_y), \\ v_c + \Gamma(1 + \cos(\omega\mathcal{R}(\mathbf{x},t)))^2 \times (\mathcal{I}(\mathbf{x},t)_x), \end{pmatrix}, & \text{if } \omega\mathcal{R}(\mathbf{x},t) \leq \pi, \\ \begin{pmatrix} h_c & u_c & v_c \end{pmatrix}^T, & \text{else,} \end{cases} \tag{5.24}$$

with

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

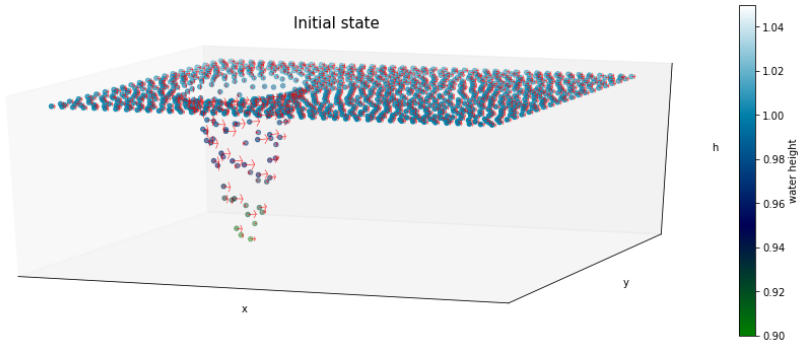
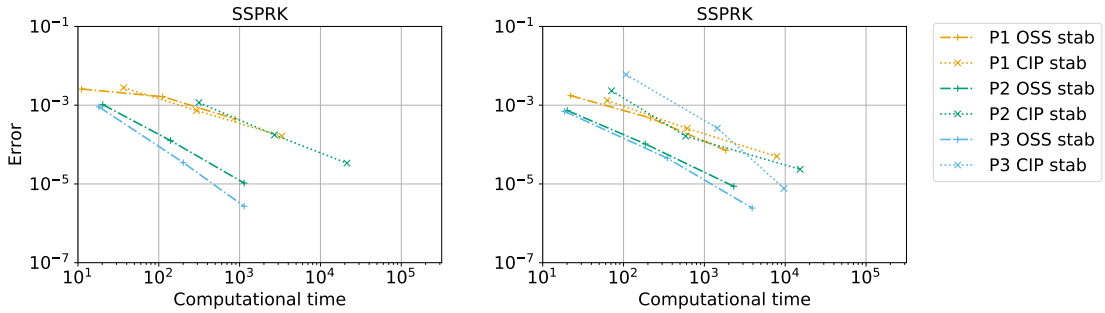


FIGURE 5.21: Initial water elevation and velocity field (red arrows) - Travelling vortex test case, $\Omega = [0,2] \times [0,1]$.

We discretize the mesh with uniform square intervals of length Δx (see figure 5.18(a)), and as before we perform a grid convergence by respecting the constraint $\Delta x_2 = 2\Delta x_1$ for $\mathbb{P}_2$ elements and $\Delta x_3 = 3\Delta x_1$ for $\mathbb{P}_3$ elements. Because of the high cost of the SUPG technique, we only compare the OSS and the CIP stabilization techniques. As an example of results, we again show the benefit of using *Cubature* elements in 5.22. We can see that since the dimension of the discretized system is even larger than before (three times larger), the differences between *Cubature* and *Basic* elements are even more highlighted in the error-computational time plot.



(a) *Cubature* elements

(b) *Basic* elements

FIGURE 5.22: Error for Shallow Water system (5.22) with respect to computational time for SSPRK method with *Cubature* (left) and *Basic* (right) elements and CIP and OSS stabilizations.

In table 5.6 we show the convergence orders for this Shallow Water problem with the CFL and δ coefficients found in table E.1.

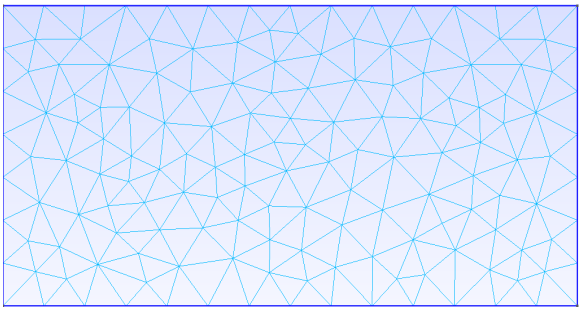


FIGURE 5.23: Unstructured mesh on $\Omega = [0,2] \times [0,1]$.

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

TABLE 5.6: Convergence order on Shallow Water, using coefficients obtained in E.1.
"/" means that the fourier analysis shown that the scheme is unstable.

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

5.4 Numerical Simulations on arbitrary mesh

We now perform numerical tests to check the validity of our theoretical findings using an unstructured mesh, and the most restrictive parameters in table 5.4. These parameters make sure that we are stable for both T and X mesh configurations. The results have similar convergence rate to the tests on the structured meshes of the previous section.
The unstructured mesh used in this section is shown in fig. 5.23, and it was created by the mesh generator *gmsh*[1].

[1] https://gmsh.info/

5.4.1 Linear advection test

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

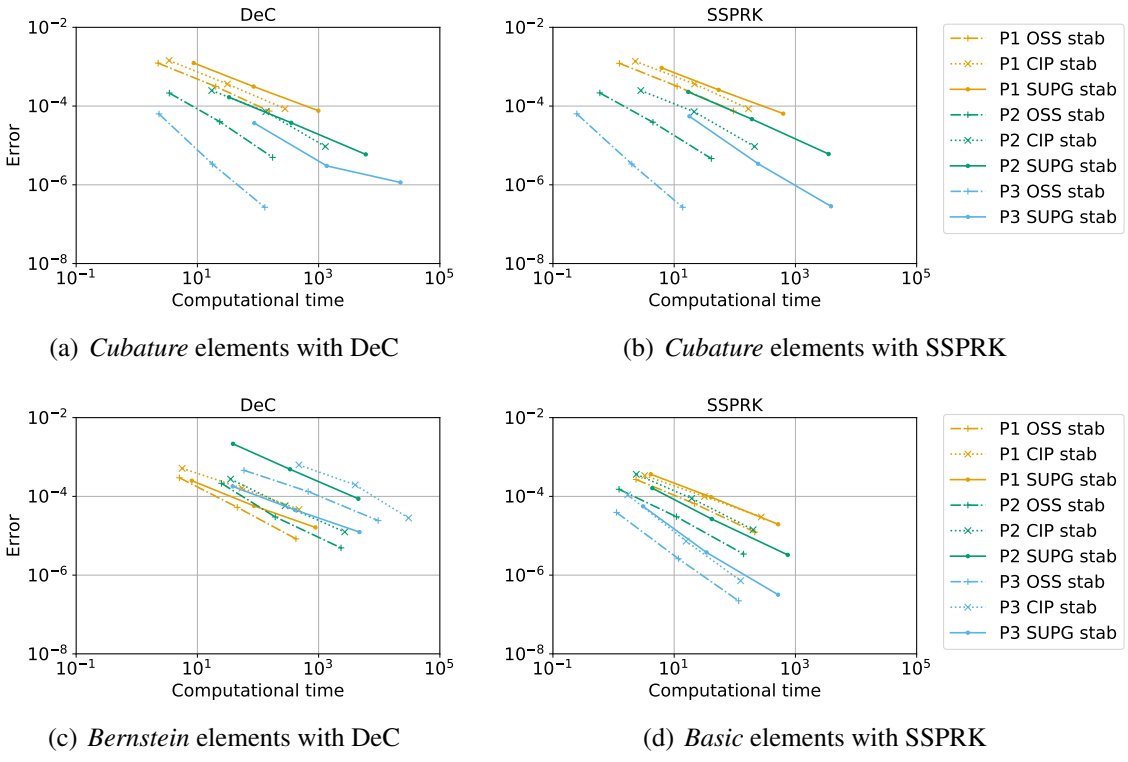
TABLE 5.7: Convergence order for linear advection on unstructured mesh, using coefficients obtained in table 5.4.
** These values are found using only the X mesh (see fig. 5.15).
"/" means that the scheme is clearly unstable.

We use the same test case of section 5.3.1. Convergence orders for all schemes are summarized in table 5.7. We observe that all $\mathbb{P}_1$ discretizations provide the proper convergence order. For $\mathbb{P}_2$ discretization we spot a slight reduction of the order of accuracy, which lays for most of the schemes between 2 and ≈ 2.5 instead of being 3. For polynomials of degree 3, we observe an order reduction to 2 for the same schemes that lost the right order of accuracy also for X mesh in the previous section. In particular, we have that *Bernstein* $\mathbb{B}_3$ elements with the *DeC* result in an order of accuracy of ≈ 2 instead of 4, as well as the $\tilde{\mathbb{P}}_3$ discretization with the combination DeC and SUPG stabilization. As for the X mesh, the *Basic* $\mathbb{P}_3$ discretization reach order of accuracy ≈ 4 for all stabilization techniques, as well as *Cubature* $\tilde{\mathbb{P}}_3$ with SUPG and OSS stabilizations.
Also in this case, the results obtained with $\tilde{\mathbb{P}}_3$ *Cubature* elements and OSS stabilization are stable as we can see from the convergence analysis. This might mean that just few unfortunate mesh configurations, as the T one, result in an unstable scheme and that, most of the time, the parameters found in table 5.4 are reliable for this scheme. On the other hand, the combination $\tilde{\mathbb{P}}_3$ and CIP gives an unstable scheme.

We compare error and computational time for all methods presented above in fig. 5.24. Looking at $\mathbb{P}_2$ and the $\mathbb{P}_3$ discretizations, as expected, the mass-matrix free combination, i.e., *Cubature* elements with SSPRK and OSS, gives smaller computational costs than other combinations with *Basic* elements. Conversely, the SUPG technique increase the computational costs with respect to all other stabilizations for all schemes. That is why we will not use it for the next test. The plots and all the errors are available at the repository [86].

Remark 5.4.1 (Entropy viscosity) *As remarked in section 5.2.9, we can improve the stability of some schemes (Cubature OSS) with extra entropy viscosity. Here, we test the convergence rate on the T mesh configuration, i.e., the one with more restrictive CFL conditions and most unstable. This test is performed using Cubature* $\tilde{\mathbb{P}}_3$ *elements, SSPRK and DeC time integration methods, and the OSS and the CIP stabilization techniques. We solve again problem* (5.21).

Using formulation (5.18) *and tuning stability coefficient* δ*, CFL and viscosity coefficient* c *found in fig. 5.17, we obtain fourth order accurate schemes. These tuned coefficients, and the corresponding convergence orders are summarized in table 5.8.*



(a) *Cubature* elements with DeC

(b) *Cubature* elements with SSPRK

(c) *Bernstein* elements with DeC

(d) *Basic* elements with SSPRK

FIGURE 5.24: Error for linear advection problem (5.21) with respect to computational time for all elements and stabilization techniques

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

TABLE 5.8: Convergence order of methods using *Cubature* $\tilde{\mathbb{P}}_3$ elements and viscosity term (5.18) with tuned parameters

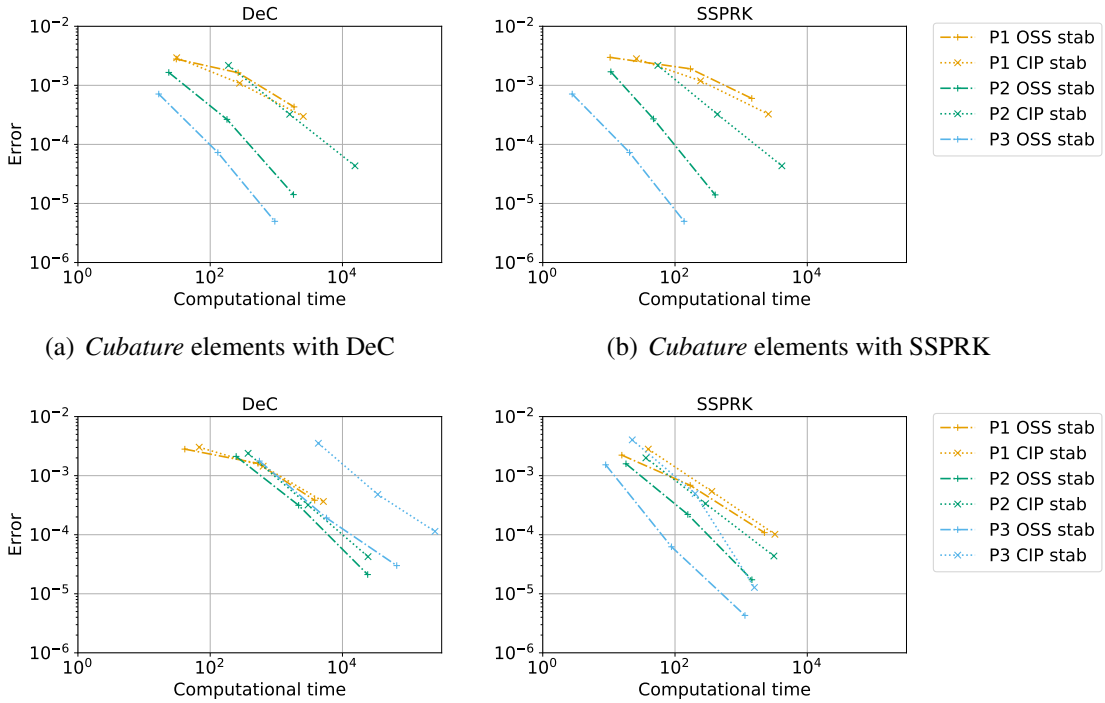
Many other formulations of viscosity terms exist in literature and can ensure convergent methods of order $p+1$ (using $\mathbb{P}_p$ elements) [58, 74, 83]. The majority use a nonlinear evaluation of the parameter μ_K, based on the local entropy production.

5.4.2 Shallow Water equations

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

TABLE 5.9: Convergence order on Shallow Water for unstructured mesh, using coefficients obtained in table 5.4.
** These values are found using only the X mesh (see fig. 5.15).
"/" means that the scheme is clearly unstable.

In this section we test the proposed schemes on the test case of section 5.3.2 with the unstructured mesh in fig. 5.23. Convergence orders are summarized in table 5.9. Also for



(a) *Cubature* elements with DeC

(b) *Cubature* elements with SSPRK

(c) *Bernstein* elements with DeC

(d) *Basic* elements with SSPRK

FIGURE 5.25: Error for Shallow Water problem (5.22) with respect to computational time for all elements and stabilization techniques

the Shallow Water equations, we have results that resemble the ones of the structured mesh. There are small differences in the order of accuracy in both directions in different schemes. Comparing also the computational time of all the schemes in fig. 5.25, we can choose what we consider the best numerical method for these test cases: *Cubature* discretization with the OSS stabilization technique. This performance seems fully provided by the free mass-matrix inversion, as the CFLs for the OSS technique (with SSPRK scheme) is approximately the same between *Basic* and *Cubature* elements (see table 5.4).

The plots and all the errors are available at the repository [86].

5.4.3 Remark on the steady vortex case

We now do a quick remark on the steady vortex case as in [107] for the isentropic Euler equations. Consider the travelling vortex proposed in section 5.3.2 with $t_f = 0.1s$. We compare the convergence orders between $u_c = 0$ (steady case) and $u_c = 0.6$ (unsteady case) in, respectively, table 5.10 and table 5.11. As we can see, in the steady case we obtain the expected convergence order for all schemes, in particular for the DeC with Bernstein polynomial function. These results agree with the ones in [107]. Comparing with the unsteady case, all the other schemes reach similar order of accuracy as obtained in table 5.9. Running the test with additional corrections in DeC scheme, as often suggested in [107, 2], does not improve the convergence order in the unsteady case (we tried even with $K = 50$).

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

TABLE 5.10: Summary tab of convergence order, steady vortex, $t_f = 0.1s$. "/" means that the scheme is clearly unstable.

Element &		OSS			CIP		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	2.34	2.68	3.86	1.94	2.53	3.61
Cub.	SSPRK	2.03	3.13	3.57	1.74	2.7	/
	DeC	2.13	3.09	3.57	1.71	2.7	/
Bern.	DeC	2.33	3.19	2.87	1.75	2.77	2.76

TABLE 5.11: Summary tab of convergence order, unsteady vortex, $t_f = 0.1s$. "/" means that the scheme is clearly unstable.

This results show that a numerical error appears in the spatio-temporal integration part of the solution eq. (3.49), which might be related to the fact that the high order derivatives are never penalized in our stabilizations and might produce some small oscillations.

5.5 Conclusion

This chapter shows also that the stability results obtained in the one dimensional analysis [88] can not be generalized for two dimensional problems on triangular meshes. In this direction, it could be interesting to perform the stability analysis on Cartesian quadrilateral meshes, to check whether in that situation the one dimensional results still hold true.

In the numerical test section, the order of accuracy found is not the expected one for all the methods, i.e., $p+1$ using $\mathbb{P}_p$ elements. For several cases, we reach only $p+1/2$

or p. Among the schemes that are stable and with the right order of accuracy, the method that uses *Cubature* elements with OSS stabilization technique and SSPRK method of order 4 has proven to be the most accurate and less expensive. Secondly, comparing to the SUPG stabilization technique, very often used in the literature for hyperbolic system, we showed that other stabilization techniques such as CIP and OSS can provide the same accuracy and are cheaper in term of computational costs.

In this direction, it would be interesting to evaluate the stability of the CIP adding a additional penalty term on the jump of higher order derivatives as suggested in [26, 22, 107]. Moreover, it could be interesting to see the stability of *Cubature* elements using higher degree polynomials. Another interesting point to explore is the loss of accuracy obtained using the DeC with Bernstein third order polynomial basis functions for unsteady cases.

To finish, as it is proposed in section 5.2.9, we suggest to add viscous term in our numerical models [58, 74], which smoothen discontinuities in the solution when shock appears, and alleviate instabilities also in smooth regions.

Chapter 6

Shallow Water equations: bore capturing and well balanced

Chapter Abstract
In this benchmarking chapter, we will perform more complex benchmarks solving the two-dimensional Shallow Water equations. Following the work done in chapter 4 (or [88]) and chapter 5 for the one and two-dimensional case on smooth solution, we evaluate our numerical methods/approaches in presence of non-smooth solution and non constant topography. In particular, having a discontinuous topography, then having a discontinuous initial solution. On one hand we study the well balanced character of the discretization in presence of bathymetry. Then, to improve the stability of the solution, we add an additional shock capturing term (see section 3.3).
In this chapter, we choose to compare best continuous Galerkin numerical approaches highlighted in previous chapters. In particular the continuous interior penalty (CIP) stabilization method and the orthogonal subscale stabilization (OSS), combining with two different choices for the continuous finite element space: Lagrangian polynomials on equispaced nodes, and Lagrangian polynomials on Cubature nodes using the strong stability preserving RK (SSPRK) time integration technique.
The principal contribution of this chapter is to propose a fully explicit, mass matrix free, high order, well-balanced and shock capturing method to solve Shallow Water equations. The chapter is organized as follows, we firstly introduce the well-balanced property in section 6.2. Secondly, we perform in several numerical tests to validate the well-balanced formulation in section 6.3. Then, using the additional entropy viscosity term, we propose several numerical tests to firstly verify the order of accuracy and highlight the benefit of using mass lumping in section 6.4, and secondly to evaluate the behaviour of this shock capturing method in presence of discontinuities in the solution with a constant topography in section 6.5 and a non constant topography in section 6.6.

Outline

6.1 Introduction 100
6.2 The well balanced formulation 100
6.3 The lake at rest solution 102
6.3.1 The lake at rest initial condition 103
6.3.2 The Lake at rest with a perturbation 104

6.4 The entropy viscosity technique 105
6.5 The asymmetric break of dam on flat bathymetry 107
6.5.1 Without shock capturing technique 108
6.5.2 Using shock capturing technique 109
6.6 Circular discontinuous perturbation on a lake at rest 111
6.7 Conclusion 115

6.1 Introduction

Numerous numerical methods are given in the literature to solve the Shallow Water equations. The main numerical challenges are the discretization of the bathymetry and friction terms. In the literature, we talk about asymptotic preserving character or *well balancedness* of a discretization [109, 57] also called the *Conservation* property, or *C-property* [12].
This property refers to the ability of the numerical discretization to preserve some steady equilibrium. The typical example, and perhaps the one most commonly encountered in nature, is the *lake at rest* state: flat free surface, and no flow.

Another important numerical challenge when solving the Shallow Water system is the numerical treatment of nearly dry regions. We talk about wetting/drying strategy [6, 16, 17, 33, 32, 49]. We won't deal with in this work.

6.2 The well balanced formulation

As it is written in section 2.2.4 the Shallow Water system can be write as follows:

$$\begin{cases} \partial_t h + \partial_x(hu) + \partial_y(hv) & = 0 \\ \partial_t(hu) + \partial_x(hu^2 + g\frac{h^2}{2}) + \partial_y(huv) & = -gh(S_{ox} + S_{fx}) \\ \partial_t(hv) + \partial_x(huv) + \partial_y(hv^2 + g\frac{h^2}{2}) & = -gh(S_{oy} + S_{fy}) \end{cases} \tag{6.1}$$

It is well known that Shallow Water equations provide satisfactory results for long wave phenomena such as tid/storm surge and tsunami (the wave length $\lambda \gg A$ the amplitude of the wave). However, they can create discontinuity in time even from smooth initial state. For these reasons, we will develop several numerical properties to satisfy in order to ensure the stability of methods. We also add stabilization technique developed above to deal with discontinuities.
Writing 6.1 in a matricial form, we obtain

$$\begin{cases} \partial_t U + \nabla \cdot \mathcal{F}(U) = \mathbf{S}(U) \\ \mathcal{F} = (F_1, F_2) \end{cases} \tag{6.2}$$

$$U = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, F_1(U) = \begin{pmatrix} hu \\ hu^2 + g\frac{h^2}{2} \\ huv \end{pmatrix}, F_2(U) = \begin{pmatrix} hv \\ huv \\ hv^2 + g\frac{h^2}{2} \end{pmatrix}, \text{and } \mathbf{S}(U) = \begin{pmatrix} 0 \\ -gh(S_{ox} + S_{fx}) \\ -gh(S_{oy} + S_{fy}) \end{pmatrix}$$

With (Sf_x, Sf_y) the friction term which is equal to zero in our case. Then

$$\begin{aligned} \text{(eq. (6.2))} \quad &\Leftrightarrow \quad \partial_t U + \nabla_u \mathcal{F}(U) \cdot \nabla U = \mathbf{S}(U) \\ &\Leftrightarrow \quad \partial_t U + (\mathcal{K}_1, \mathcal{K}_2) \cdot \nabla U = \mathbf{S}(U) \end{aligned}$$

$$\text{with} \quad \mathcal{K}_1 = \begin{pmatrix} 0 & 1 & 0 \\ gh - u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix} \quad \text{and} \quad \mathcal{K}_2 = \begin{pmatrix} 0 & 0 & 1 \\ -uv & v & u \\ gh - v^2 & 0 & 2v \end{pmatrix} \tag{6.3}$$

Spectrum of matrices $\mathcal{K}_1$ and $\mathcal{K}_2$ are respectively

$$Sp_1(u) = \{u, u + \sqrt{gh}, u - \sqrt{gh}\} \quad \text{and} \quad Sp_2(v) = \{v, v + \sqrt{gh}, v - \sqrt{gh}\}$$

We introduce now several notations: the celerity of the flow $c = \sqrt{gh}$, the free surface elevation $\eta = h + b$, the *specific total energy* $\epsilon_s = g\eta + k$ with k the kinetic energy $k = \frac{\|(u,v)^T\|^2}{2}$, the *discharge* $\vec{q} = h(u,v)^T$ and the Froude number $Fr = \frac{\|(u,v)^T\|}{c}$. The Froude number characterizes the ratio between the flow speed (or the kinetic energy) and the celerity (or the gravity potential energy). If $Fr < 1$ the flow is called a *subcritical flow*, if $Fr > 1$ the flow is characterized as *supercritical flow* and if $Fr \approx 1$ the flow is denoted as *critical flow*.

Another very important notion is the ***lake at rest*** which corresponds to the hydrostatic equilibrium. The ***lake at rest*** is characterized by the *discharge* $\vec{q} = h(u,v)^T = 0$ and $\nabla(g\frac{h^2}{2}) + gh\nabla b = gh\nabla(h+b) = 0$.
This is always satisfied by the physical steady state $(u,v)^T = \vec{0}$ and $\eta = \eta_0 = cst$.
To preserve the lake at rest, we use a different approach of the initial shallow-water formulation. The mono dimensional shallow-water frictionless system is written as follows

$$\begin{cases} \partial_t h + \partial_x(hu) & = 0, \\ \partial_t(hu) + \partial_x(hu^2 + g\frac{h^2}{2}) + gh\partial_x b & = 0. \end{cases} \tag{6.4}$$

However, considering the finite elements approximation of $h, u = 0$ and b, and the space of basis function $(\varphi_i)_{i \in T_K}$ of each elements K, the spatial derivative approximation becomes

$$\sum_{i \in K} \frac{h_i^2}{2} \partial_x \varphi_i + \sum_{j \in K} h_j \varphi_j \sum_{i \in K} b_i \partial_x \varphi_i \tag{6.5}$$

which is not necessarily equal to 0. Following Y. Xing and C.W. Shu [128], we can rewrite the formulation as

$$\textit{eq. (6.4)} \quad \Leftrightarrow \begin{cases} \partial_t h + \partial_x(hu) & = 0, \\ \partial_t(hu) + \partial_x(hu^2) + (gh\partial_x h - gb\partial_x b) + g(h+b)\partial_x b & = 0, \end{cases} \tag{6.6}$$

$$\Leftrightarrow \begin{cases} \partial_t h + \partial_x(hu) & = 0, \\ \partial_t(hu) + \partial_x(hu^2) + (g\partial_x \frac{h^2 - b^2}{2}) + g(h+b)\partial_x b & = 0. \end{cases} \tag{6.7}$$

The spatial derivative approximation becomes (knowing that $h + b = cst = \eta_0$)

$$\sum_{i\in K} \frac{h_i^2 - b_i^2}{2} \partial_x \varphi_i + \eta_0 \sum_{i\in K} b_i \partial_x \varphi_i \tag{6.8}$$

$$= \sum_{i\in K} \eta_0 \frac{h_i - b_i}{2} \partial_x \varphi_i + \eta_0 \sum_{i\in K} b_i \partial_x \varphi_i \tag{6.9}$$

$$= \eta_0 \left(\sum_{i\in K} \frac{h_i + b_i}{2} \partial_x \varphi_i \right) = \frac{\eta_0^2}{2} \left(\sum_{i\in K} \partial_x \varphi_i \right) = 0 \tag{6.10}$$

Similarly, eq. (6.2) becomes

$$\begin{cases} \partial_t U + \nabla \cdot \mathcal{F}^{WB}(U) = \mathbf{S}(U) \\ \mathcal{F}^{WB} = (F_1^{WB}, F_2^{WB}) \end{cases} \tag{6.11}$$

$$F_1^{WB}(U) = \begin{pmatrix} hu \\ hu^2 + g\frac{h^2-b^2}{2} \\ huv \end{pmatrix} \quad F_2^{WB}(U) = \begin{pmatrix} hv \\ huv \\ hv^2 + g\frac{h^2-b^2}{2} \end{pmatrix}$$

$$\text{and } \mathbf{S}(U) = \begin{pmatrix} 0 \\ -g(h+b)\partial_x b \\ -g(h+b)\partial_y b \end{pmatrix}$$

This last formulation eq. (6.11) preserves the lake at rest in the sense that for any high order spatial discretization and time integration scheme, if initial conditions are defined by $(h_0 + b_0) = \eta_0 = cst$ and $(u_0, v_0)^T = \vec{0}$, we obtain $\partial_t h = \partial_t(hu) = \partial_t(hv) = 0$. This property is commonly call the *well-balanced* property.

6.3 The lake at rest solution

In this section, we propose a comparison between the *well-balanced* and non *well-balanced* approaches. Two important things are considered: the lake at rest conservation and the accuracy of numerical methods. The first one will be tested using initial lake at rest condition with a smooth and a non-smooth topography. Then, the second one considering a perturbation of the lake at rest, by comparing our results with those obtain in [109]. The aim is to preserve the lake at rest, away from perturbation, and converge to a smooth solution.

We start by considering the lake at rest solution initial condition on two different bathymetries. The spatial domain $\Omega = [0, 2] \times [0, 1]$, and the bathymetries are defined by:

$$b(x, y) = b_0 e^{\psi(x,y)} \tag{6.12}$$

The first smooth bathymetry, which is generally used to verify the C-property (see e.g. [129, 114, 80] and references therein) is obtained with

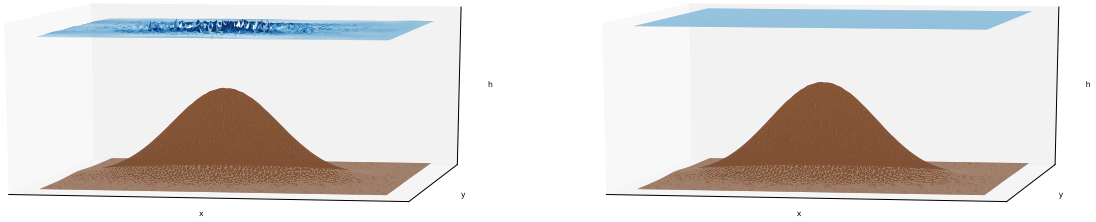
$$b_0 = 0.8, \quad \text{and } \psi = -5(x - 0.9)^2 - 50(y - 0.5)^2 \tag{6.13}$$

The second non-smooth bathymetry proposed in [109] is the same on which we add a discontinuity:

$$b_0 = 0.6, \quad \text{and } \psi = \begin{cases} \sqrt{(x-0.9)^2 + (y-0.5)^2} & \text{if } (x,y) \in [0.9,1.1] \times [0.3,0.7], \\ -5(x-0.9)^2 - 50(y-0.5)^2 & \text{otherwise} \end{cases} \tag{6.14}$$

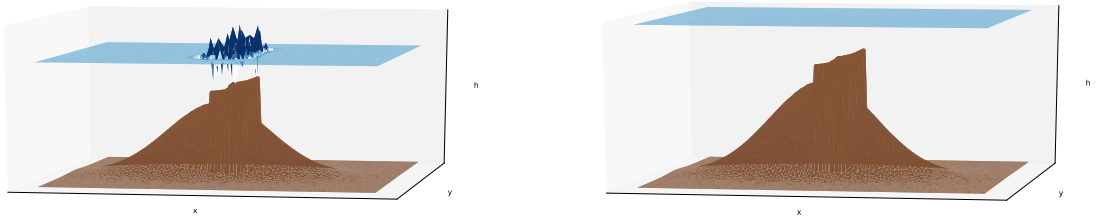
6.3.1 The lake at rest initial condition

In a first time, we consider the initial lake at rest solution $(\eta, q) = (1,0),\ \forall (x,y) \in \Omega$ using the Well-Balanced definition of methods and without. The numerical computation is done using an unstructured mesh. The mesh size used is $\Delta x_1 = 0.05$, $\Delta x_p = p \times \Delta x_1$, so ≈ 1000 nodes per mesh. An example of the *Well-Balanced* and non *Well-Balanced* behaviour is represented in fig. 6.1 and in fig. 6.2. The numerical method consider in this example is *Basic* $\mathbb{P}_1$ discretization, SSPRK time integration method and the OSS stabilization technique. The physical time of this test is $t_f = 1s$.



(a) not *Well-Balanced*. (b) *Well-Balanced*.

FIGURE 6.1: Lake at rest solution with a smooth topography, $t_f = 1s$ using Basic $\mathbb{P}_1$ discretization with the OSS stabilization technique and the SSPRK(3,2) scheme. Amplification factor of water instabilities = 100.



(a) not *Well-Balanced*. (b) *Well-Balanced*.

FIGURE 6.2: Lake at rest solution with a discontinuous topography, $t_f = 0.02s$ using Basic $\mathbb{P}_1$ discretization with the OSS stabilization technique and the SSPRK(3,2) scheme.

As expected, using the *Well-Balanced* formulation, the lake at rest is preserved even in the presence of discontinuity in the bathymmetry, see fig. 6.1(b) and fig. 6.2(b). However,

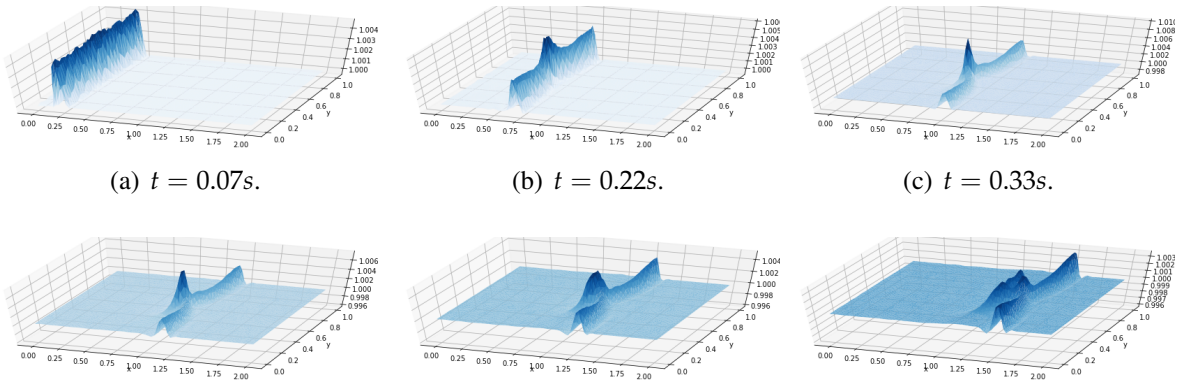
using a non *Well-Balanced* formulation does not allow to preserve the lake at rest. Indeed, small perturbations appear even with a smooth topography as it is shown in fig. 6.1(a), and high oscillations appear in presence of discontinuities in the bathymetry as we can see in fig. 6.2(a).

6.3.2 The Lake at rest with a perturbation

To visualize better this fact, we now add a perturbation of the lake at rest state defined by $(hu, hv) = 0$, and $\forall (x, y) \in \Omega$

$$\eta_0(x, y) = 1 + e^{-(x \times 30)^2} \times 0.01 \tag{6.15}$$

which corresponds to a gaussian function centered in 0, and we add a *wall* condition on the left boundary of the domain $x = 0$.
In [109] and references therein (i.e. [80, 109, 111, 114, 129]), they use a discontinuous initial solution ($\eta_0(x, y) = 1.01$ if $x \in [0.05, 0.15]$, 1 else), which is possible only if we are able to treat shock with entropy viscosity stabilization technique for example [58, 74].
We choose a mesh size $\Delta x_1 = 0.025m$, and so ≈ 4500 nodes for the CG case. Firstly, we show the 3d view of the free surface in fig. 6.3 to understand properly the global behaviour of the wave at different time.

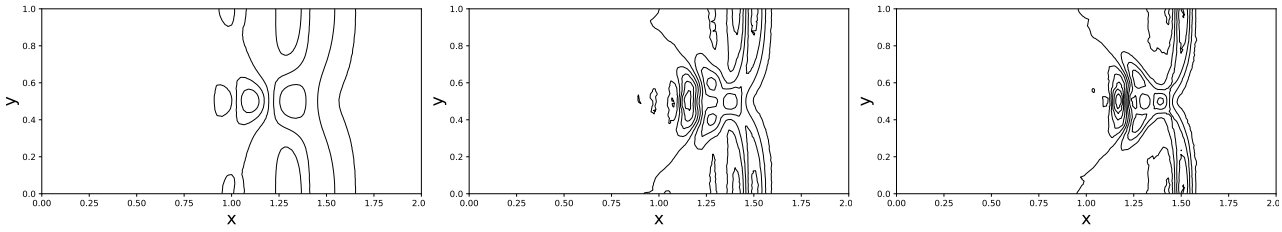


(a) $t = 0.07s$. (b) $t = 0.22s$. (c) $t = 0.33s$.

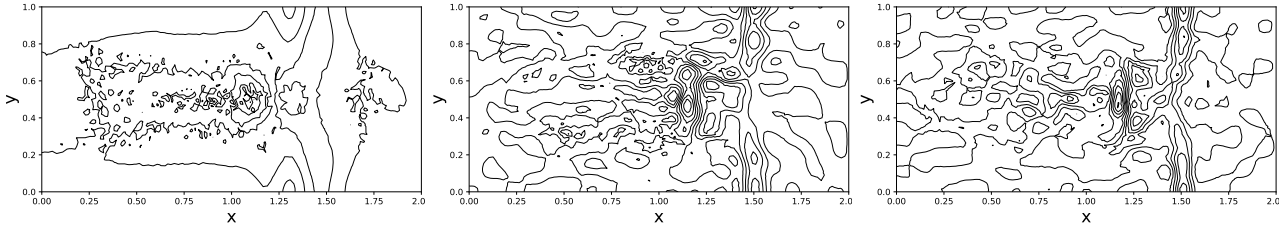
(d) $t = 0.37s$. (e) $t = 0.41s$. (f) $t = 0.48s$.

FIGURE 6.3: Propagation of the perturbation eq. (6.15) at different time step using *Basic* $\mathbb{P}_1$ elements with the OSS stabilization technique and the SSPRK(3,2) scheme.

The perturbation behaves uniformly in the $\vec{x}$ direction until $t \approx 0.20s$, then the wave is deformed during the propagation above the *hill*. This behaviour corresponds to what is observed in the literature. The lake at rest seems to be preserved away from the perturbation. To see better this fact, we plot the 2d view from the top with line contour of water elevation for the *Cubature* elements, with the OSS stabilization technique in fig. 6.4. In particular, we compare the approximated solution using and not using the well-balanced formulation. Comparison using *Basic* and the CIP stabilization technique are presented in Appendix F.1.1. The free surface elevation using the non-smooth topography and the well-balanced formulation is placed in Appendix F.1.2.



(a) *Cubature* $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ elements (from left to right) using the well-balanced formulation.



(b) *Cubature* $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ elements (from left to right) not using the well-balanced formulation.

FIGURE 6.4: 2D view of the free surface elevation at $t = 0.48s$ with the smooth topography eq. (6.13). All simulation are performed using *Cubature* elements, the OSS stabilization technique, and SSPRK schemes.

We observe the perfect preservation of the lake at rest away from the perturbation using the well-balanced formulation, see fig. 6.4(a). While, if we do not use a well-balanced formulation, oscillations appear upstream and downstream of the perturbation, see fig. 6.4(b). The preservation of the lake at rest away from the perturbation is also the case with a discontinuous topography, even if very small smooth oscillations appear behind the wave (see Appendix F.1.2). This result show in particular the well balancedness of the numerical methods.

6.4 The entropy viscosity technique

In this section, we consider the entropy viscosity proposed by J.L. Guermond et al. [101, 58], described in section 3.3. We already highlighted the benefit of using mass lumping with *Cubature* elements in the previous chapters using linear stabilization techniques. The aim is now to preserve this order of accuracy using the additional viscosity term for smooth solutions. We expect as in the previous chapters, to see a benefit in using mass lumping in term of cpu-time.

We consider the travelling vortex proposed in section 5.3.2 on the same meshes, and $t_f = 0.5s$. The initial free surface elevation is represented in fig. 6.5. Numerical methods compared in this study are SSPRK time integration methods, with the CIP and the OSS stabilization techniques, combined with *Basic* and *Cubature* elements. Convergence orders for all of these schemes are summarized in table 6.1. By default we use penalty coefficient from the two-dimensional analysis table 5.4 in the previous chapter. Then, taking in account CFL values in table 5.4, we choose to compare both spatial discretization with the same value: $CFL = 0.4$, 0.2 and 0.1 for respectively $p = 1, 2$ and 3. We summarize parameters

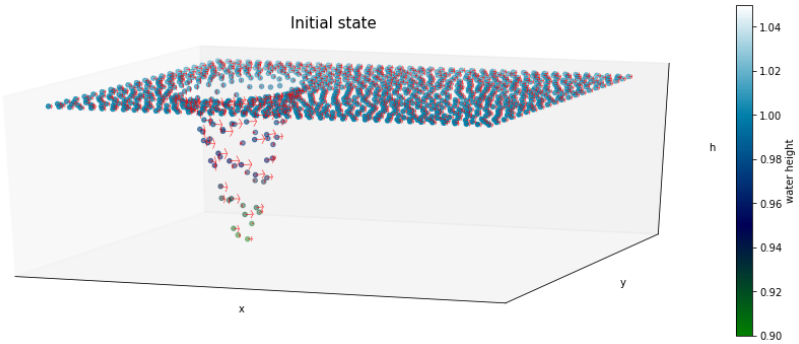


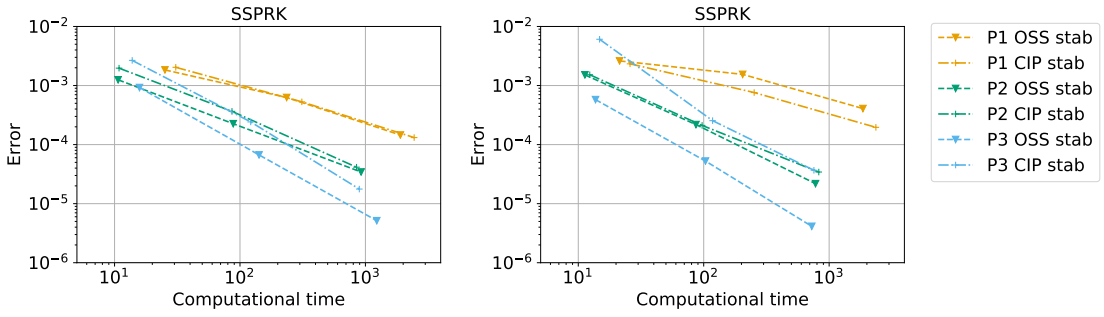
FIGURE 6.5: Initial water elevation and velocity field (red arrows) - Travelling vortex test case, $\Omega = [0,2] \times [0,1]$.

used for the convergence tests in table 6.2. We note that the viscosity parameter c_E (from section 3.3) was chosen after several tests.

Element & Time scheme	OSS			CIP		
	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic & SSPRK	1.6	2.4	3.8	1.8	2.5	3.6
Cub. & SSPRK	1.2	2.8	3.6	1.6	2.5	3.7

TABLE 6.1: Convergence order for Shallow Water problem (5.22) on unstructured mesh, $t_f = 0.5s$.

As we can see in table 6.1, for the majority of numerical methods, convergence orders are between $p + 1/2$ and $p + 1$. This first information is very promising. In particular, for the two fourth order of accuracy and mass matrix free methods, i.e. using *Cubature* $\tilde{\mathbb{P}}_3$ elements combined with the CIP or the OSS stabilization techniques. A disappointing result is the convergence order obtained with *Cubature* $\tilde{\mathbb{P}}_1$ elements with the OSS. This result was also observed in the previous chapter (see section 5.4.2). We now compare error and computational time for all methods presented above in fig. 6.6.



(a) *Basic* elements (b) *Cubature element*

FIGURE 6.6: Error for Shallow Water problem (5.22) with respect to computational time for all elements, the linear stabilization techniques and the additional viscosity term.

As expected, using the mass lumping, fastest methods are obtained using the mass lumping. When we now look at the ratio error/cpu-time, for elements of degree $p = 1$, as the

order of accuracy of *Cubature* $\tilde{\mathbb{P}}_1$ elements with the OSS is low, the advantage goes to *Basic* $\mathbb{P}_1$ elements and *Cubature* $\tilde{\mathbb{P}}_1$ with the CIP which are equivalent. Then, for elements of degree $p = 2$ the advantage goes to *Cubature* with the OSS and the CIP which are equivalent. And finally for elements of degree $p = 3$, the advantage goes clearly to *Cubature* elements, with the OSS.

Element & Time scheme	OSS		
	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic & SSPRK	0.4 (0.13, 0.05)	0.2 (0.05, 0.05)	0.1 (0.026, 0.05)
Cub. & SSPRK	0.4 (0.34, 0.05)	0.2 (0.08, 0.05)	0.1 (0.018, 0.001)
Element & Time scheme	CIP		
	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic & SSPRK	0.4 (0.012, 0.05)	0.2 (0.0008, 0.05)	0.1 (0.0005, 0.05)
Cub. & SSPRK	0.4 (0.048, 0.05)	0.2 (0.002, 0.05)	0.1 (0.0004, 0.5)

TABLE 6.2: CFL, penalty coefficient δ and viscosity term c_E in parenthesis used for convergence tests: CFL (δ, c_E)

6.5 The asymmetric break of dam on flat bathymetry

As we saw in section 5.4 using linear stabilization techniques, for smooth solution, all of our numerical schemes gives approximated solutions with the accuracy expected, i.e. order of accuracy $\approx p+1$ using elements $\mathbb{P}_p$. However, for coastal engineering applications, we need to use shock capturing techniques to deal with possible discontinuities. In section 6.4, we showed in particular that entropy viscosity method proposed in section 3.3 allows to conserve the order of accuracy expected in smooth region using *Basic* and *Cubature* elements with mass lumping. The aim of this work being to propose a mass matrix free, stabilized and shock capturing technique for Shallow Water equations, we now consider a discontinuous initial solution. This test case corresponds to a break of a dam. In this section, we simulate the propagation without using shock capturing technique in section 6.5.1 and then using shock capturing technique (from section 3.3) in section 6.5.2.

This test is taken from [84, 114]. It consists of the asymmetric break of dam separating two basins with two different water depths: 5 and 10 meters. The dam is contained in the computational domain $\Omega = [0,200]^2$, and the breaking is initially placed at $x = 95[m]$, and y from $84.5[m]$ to $179.5[m]$. A representation of the test case is shown in fig. 6.7 for the initial state and in fig. 6.8 for the final state using *Basic* $\mathbb{P}_1$ elements, the OSS stabilization technique and $\Delta x_1 = 2m$ ($\approx$ 13500 nodes and 26500 triangles in $\mathbb{P}_1$). Reflective boundary conditions are used on all boundaries, see [84, 114] for more details. This benchmark will show us in particular the benefit of using stabilization technique when the initial condition is not continuous.

For this test case, to be comparable with results obtained in [109] and [111], we now choose $\Delta x_1 = 2m$ which correspond to $\approx$ 13500 nodes (and 26500 triangles in $\mathbb{P}_1$) and we report the water elevation at $t = 7.2s$. We plot in a first time the 3d contour of the free surface elevation, view from the top to see the global free surface elevation over the domain. Then we plot the 1d water elevation along $y = 132m$ (equidistant to both walls, see the

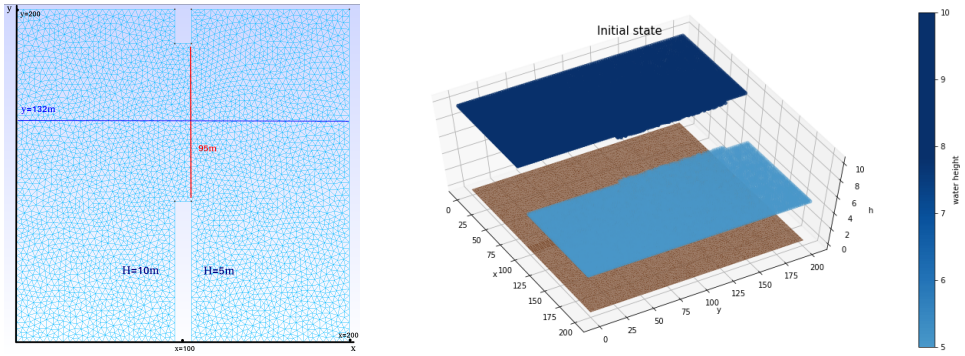


FIGURE 6.7: Asymmetric dam break. At left, visualisation of the domain Ω_h and initial conditions. At right: 3d view of the initial state.

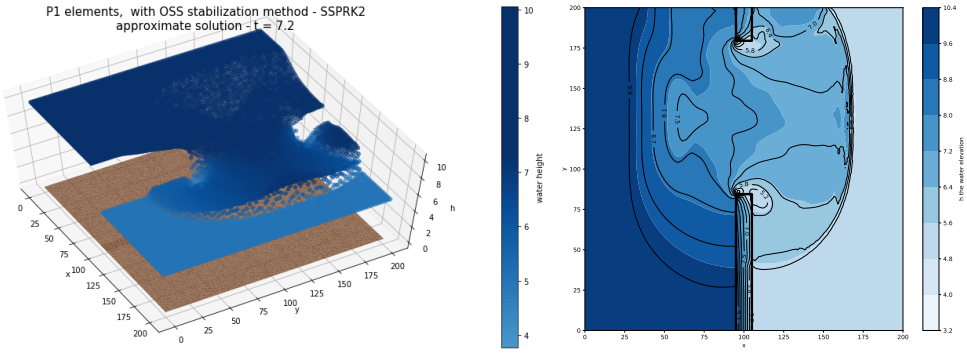


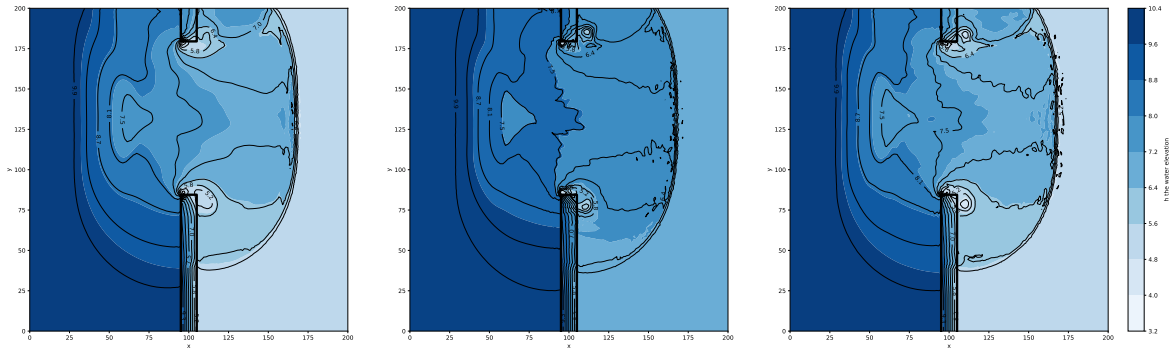
FIGURE 6.8: Asymmetric dam break. At left, 3d view of the final state at $t = 7.2$. At right: 2d view from the top of the final state, contour line of water elevations

blue line in fig. 6.7), which is the position of the trough due to the interaction of the corner rarefactions to see the approximated solution around discontinuities.

6.5.1 Without shock capturing technique

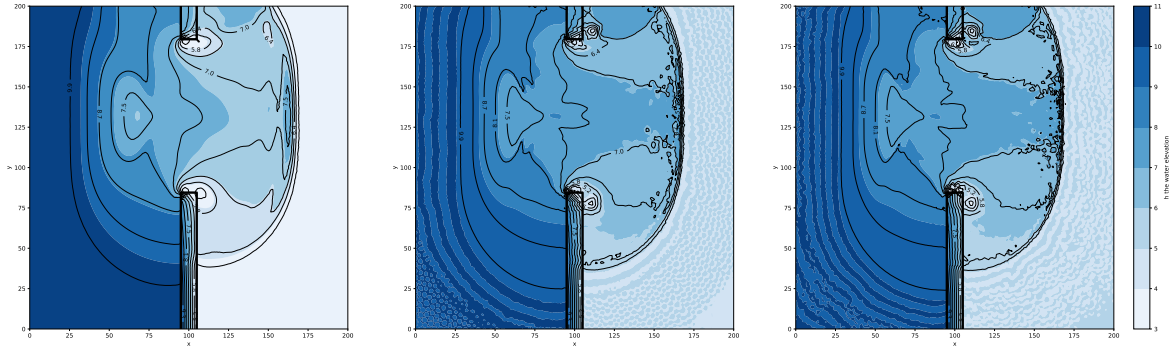
For this test case, we only show results using the OSS stabilization technique with the OSS because $\mathbb{P}_2$ and $\mathbb{P}_3$ *Basic* and *Cubature* elements using the CIP lead to an unstable scheme. This result is not surprising in the sense that these dispersion methods are not supposed to be stable in presence of a discontinuity.

As we can see in the 2d view from the top with line contour of water elevation (see fig. 6.9 for the *Basic* discretization and in fig. 6.10 for the *Cubature* discretization), all numerical schemes behave similarly around corner and discontinuity. The waves are propagated with the same speed and the surface elevations are quite similar. This behaviour is also equivalent to the behaviour observed in [109, sec. 5.1]. We now compare the free surface elevation at $y = 132m$ (see blue line in fig. 6.7) in fig. 6.11. Both spatial discretizations lead to the same dispersive behaviour that we already observed in 1d in fig. 4.15 for a linear advection case. The *Cubature* discretization seems to oscillate more than the *Basic* discretization around discontinuities. But the scheme stay stable and converge. Moreover, the steady state of the solution in the rest of the domain, away from the discontinuities, is preserved ($x < 25m$ and



(a) *Basic* $\mathbb{P}_1$. (b) *Basic* $\mathbb{P}_2$. (c) *Basic* $\mathbb{P}_3$.

FIGURE 6.9: 2D view of the free surface elevation at $t = 7.2s$, *Basic* elements with the OSS stabilization technique.



(a) *Cubature* $\tilde{\mathbb{P}}_1$. (b) *Cubature* $\tilde{\mathbb{P}}_2$. (c) *Cubature* $\tilde{\mathbb{P}}_3$.

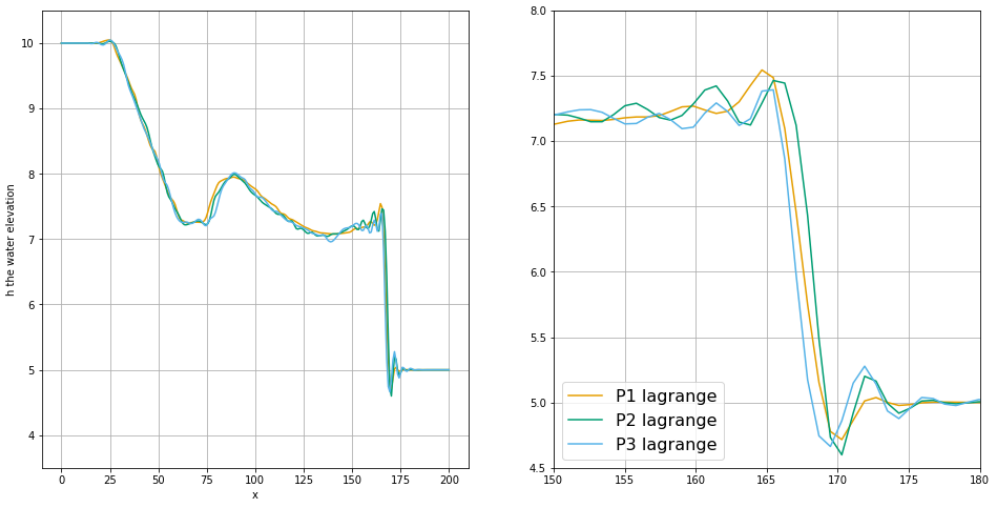
FIGURE 6.10: 2D view of the free surface elevation at $t = 7.s$, *Cubature* elements with the OSS stabilization technique.

$x > 175m$). The main difference with [109, 111] is the use of a diffusive scheme which smoothen shocks, instead of our schemes which are dispersive. Their results are reported in fig. 6.12 We can clearly see a difference between our dispersive schemes and the diffusive scheme from [109]. The solution is much smoother, however we can see that the global behaviour of the solutions are comparable.

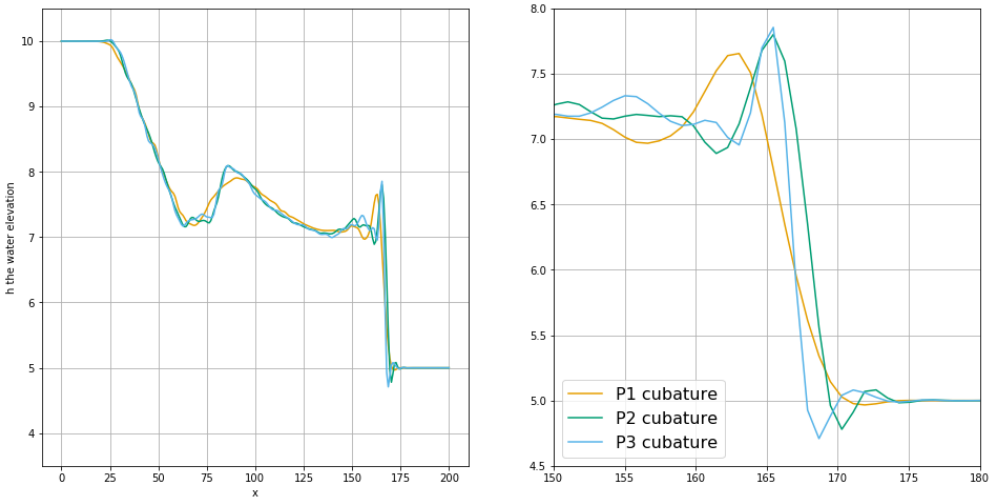
6.5.2 Using shock capturing technique

We use now the additional stabilization term describe in section 3.3 using the definition of parameters from [18], and we test different values for α in order to remove all spurious oscillations, $\kappa = 1/p$, $\tau = 1/10$ stabilized shocks and impose a minimal viscosity and see the global behaviour of our numerical test, $\beta = 0.057$. We still consider our both stabilization technique (OSS and CIP) and we add the diffusive term.

A first improvement is that all numerical schemes are stable and gives comparable results. Indeed, we obtain the same behaviour than in [109]. To understand the effect of the viscosity stabilization term, we plot the 1d water elevation at $y = 132m$, $t = 7.2s$ along the shock ($x \in [150, 180]$) using different value of α. We here consider the CFLs and penalty



(a) *Basic* elements. At left: $x \in [0, 200]$, at right $x \in [150, 180]$.



(b) *Cubature* elements. At left: $x \in [0, 200]$, at right $x \in [150, 180]$.

FIGURE 6.11: Asymmetric dam break: water height at time $t = 7.2s$. Data extracted along the line $y = 132m$. Use of the OSS stabilization technique.

coefficients from the analysis done previously, and we consider for the numerical scheme using *Cubature* $\tilde{\mathbb{P}}_3$ discretization and CIP stabilization (which were not stable in the previous chapter) coefficients from the *Basic* $\mathbb{P}_3$ discretization, CIP stabilization and SSPRK method by default. An important point is that if $\alpha\kappa$ is "too high", we have to decrease the CFL to be stable. In our test, we only plot the water elevation when the method is stable using the CFLs describe above.

We confirm in this test the fact that for a same method (time scheme, type of discretization and stabilization technique), μ must have a lower value for higher order methods, and so $\alpha\kappa$ has to be lower for higher polynomial degree. Moreover, as we can see in figures 6.13 and 6.15 using the OSS stabilization technique, the additional viscous term dumps spurious

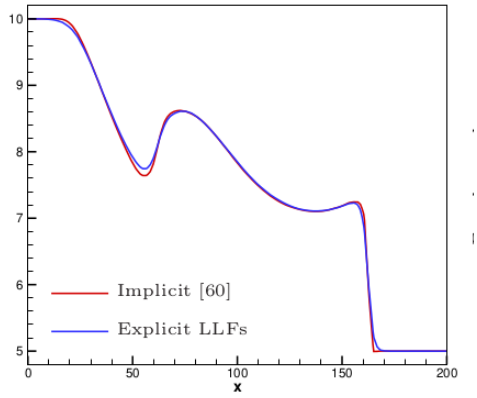
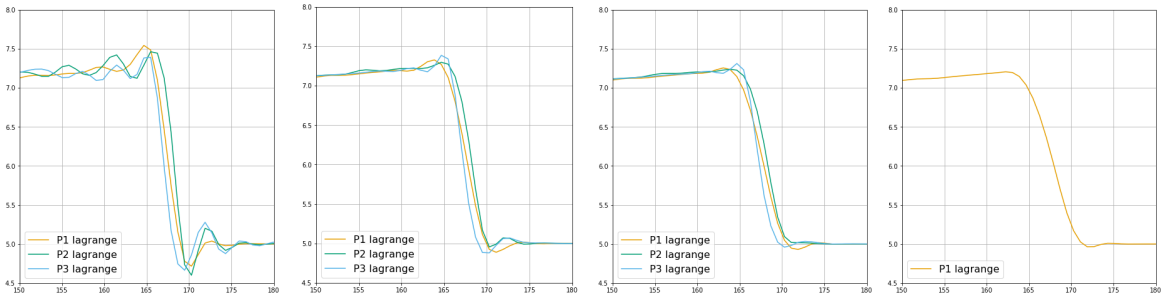


FIGURE 6.12: Asymmetric dam break: water height at time $t = 7.2s$. Data from [109], extracted along the line $y = 132m$



(a) $\alpha\kappa = 0 \times 1/p$ (b) $\alpha\kappa = 0.05 \times 1/p$ (c) $\alpha\kappa = 0.075 \times 1/p$ (d) $\alpha\kappa = 0.1 \times 1/p$

FIGURE 6.13: Asymmetric dam break: water height at time $t = 7.2s$. Data extracted along the line $y = 132m$. Use of the *Basic* discretization, the OSS and entropy stabilization technique.

oscillations until remove it (increasing the term $\alpha\kappa$). Also, we can see this fact in figures 6.14 and 6.16 with the CIP stabilization technique. We can also notice that the choice of coefficients seem very sensitive considering the numerical test.

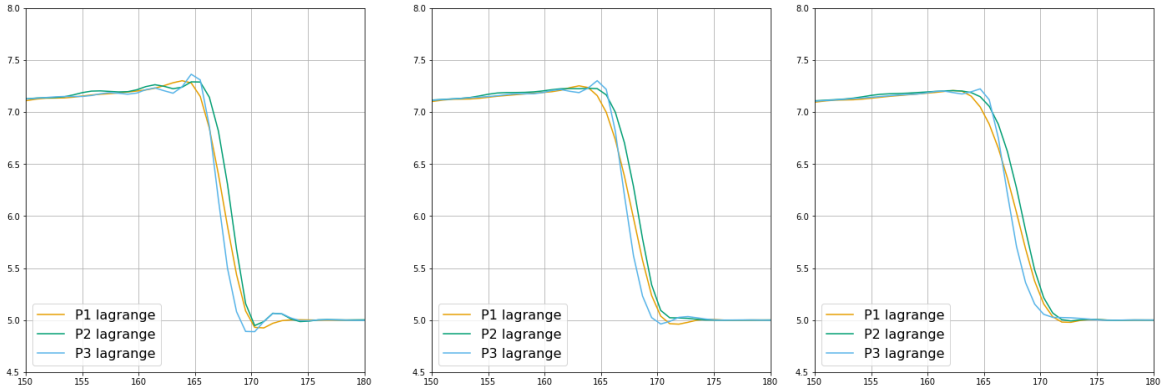
6.6 Circular discontinuous perturbation on a lake at rest

For this final numerical test case using the continuous Galerkin approach, we consider a perturbation of a lake at rest, which is propagated circularly on a domain $\Omega = [0,2] \times [0,2]$. The bathymetry is defined by

$$b(x,y) = b_0 \cos(\theta_{\frac{\pi}{4}}(x,y) \times \tau_0 \pi) \tag{6.16}$$

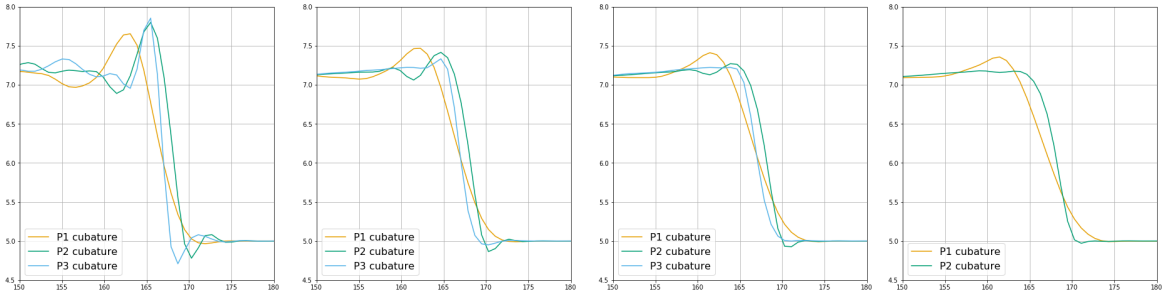
$\theta_{\frac{\pi}{4}}(x,y) = \cos(-\pi/4) \times x - \sin(-\pi/4) \times y$, which corresponds to a rotation of $\pi/4$, and $b_0, \tau_0 \in \mathbb{R}$.

In the first case, we choose $b_0 = 0.05$ and $\tau_0 = 6$ (see fig. 6.17(a)), the topography does no influence on the wave propagation while $b_0 \ll \eta_0$. This fact is not valid in the second case where $b_0 = 0.05$ and $\tau_0 = 6$ (see fig. 6.17(b)). The time of the simulation is $t_f = 0.18$. To stabilize numerical method, we add the viscous stabilization term (the entropy stabilization



(a) $\alpha\kappa = 0.05 \times 1/p$ (b) $\alpha\kappa = 0.075 \times 1/p$ (c) $\alpha\kappa = 0.1 \times 1/p$

FIGURE 6.14: Asymmetric dam break: water height at time $t = 7.2s$. Data extracted along the line $y = 132m$. Use of the *Basic* discretization, the CIP and entropy stabilization technique.



(a) $\alpha\kappa = 0 \times 1/p$ (b) $\alpha\kappa = 0.05 \times 1/p$ (c) $\alpha\kappa = 0.075 \times 1/p$ (d) $\alpha\kappa = 0.1 \times 1/p$

FIGURE 6.15: Asymmetric dam break: water height at time $t = 7.2s$. Data extracted along the line $y = 132m$. Use of the *Cubature* discretization, the OSS and entropy stabilization technique.

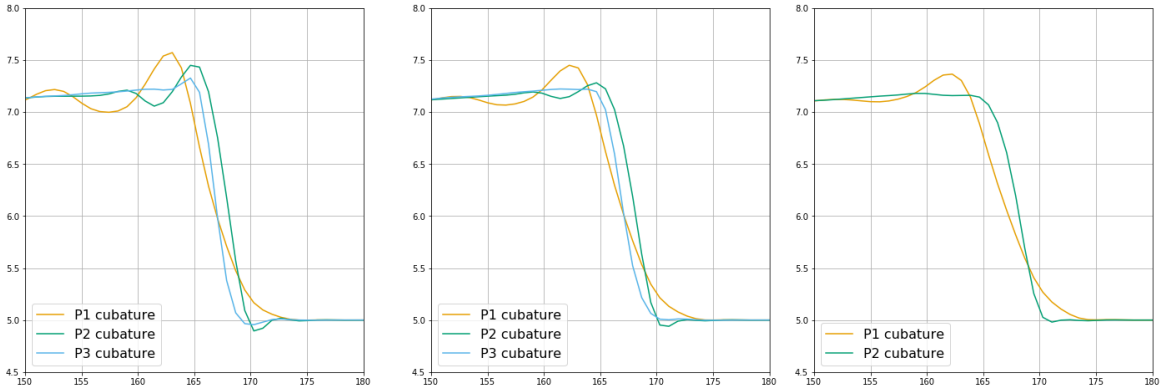
term) and we choose $\alpha = 0.05$. The initial water elevation is described $\forall (x,y) \in \Omega$ by

$$\eta_0(x,y) = \begin{cases} 0.7 & \text{if } d_c(x,y) < 0.12 \\ 0.5 & \text{else} \end{cases}, \quad d_c(x,y) = \sqrt{(x-1)^2 + (y-1)^2}. \tag{6.17}$$

η_0 is discontinuous, while $b_0 \in \mathcal{C}^\infty$. We add viscous stabilization term to smooth discontinuities.

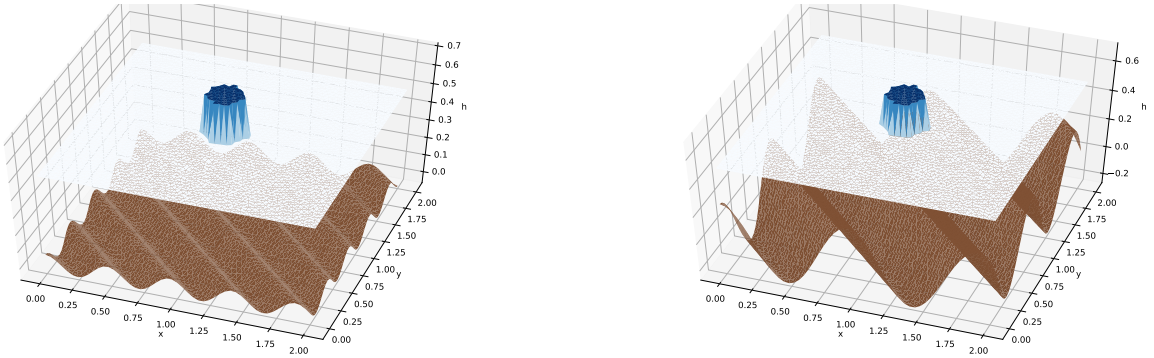
A 3d representation of the final state is shown in fig. 6.18(a) and fig. 6.18(b). As an example, we plot the 2d view from the top with line contour of water elevation (see fig. 6.19 using the OSS stabilization technique and *Basic* and *Cubature* discretizations) for the case 2.

Again, we observe that the lake at rest is preserved away from the perturbation and the perturbation is propagated smoothly. This is the case for base case 1 ($b_0 = 0.05$ and $\tau_0 = 6$) and case 2 ($b_0 = 0.05$ and $\tau_0 = 6$) for all numerical methods. These results show again the



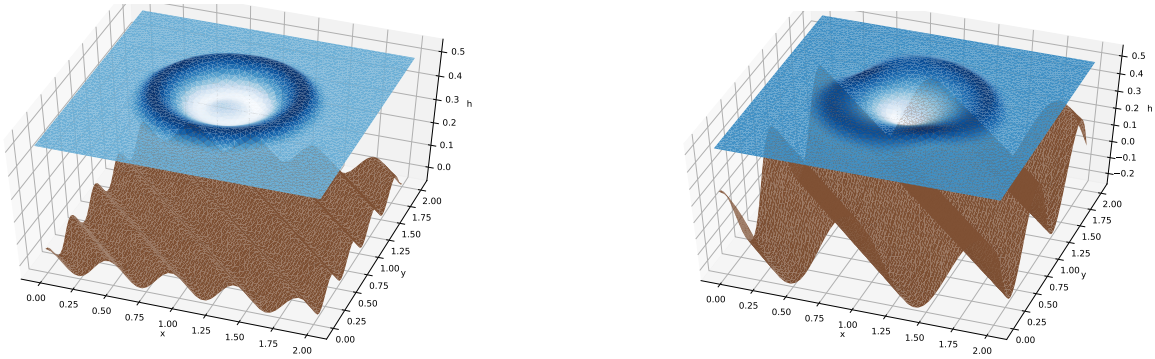
(a) $\alpha\kappa = 0.05 \times 1/p$ (b) $\alpha\kappa = 0.075 \times 1/p$ (c) $\alpha\kappa = 0.1 \times 1/p$

FIGURE 6.16: Asymmetric dam break: water height at time $t = 7.2s$. Data extracted along the line $y = 132m$. Use of the *Cubature* discretization, the CIP and entropy stabilization technique.



(a) Perturbation of a lake at rest, case 1. (b) Perturbation of a lake at rest, case 2.

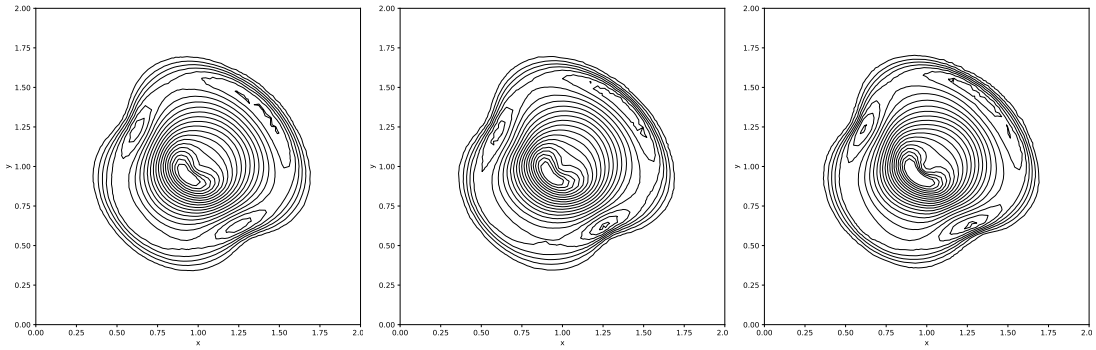
FIGURE 6.17: Perturbation of a lake at rest, initial solution.



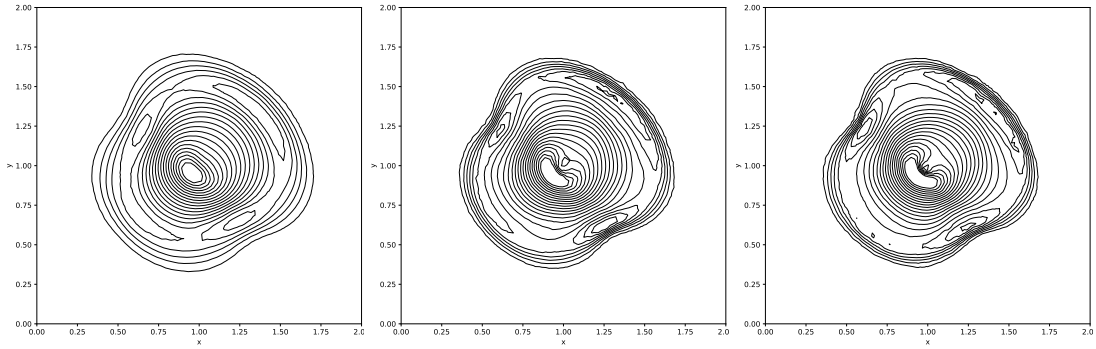
(a) Perturbation of a lake at rest, case 1. (b) Perturbation of a lake at rest, case 2.

FIGURE 6.18: Final state of the perturbation eq. (6.17) using *Basic* $\mathbb{P}_1$ elements with the OSS stabilization technique and the SSPRK(3,2) scheme.

well-balancedness and the entropy conservative character of our numerical methods. Moreover, we also shown these properties for numerical schemes which use the sparse matrix inversion (*Basic* discretization) and the mass-lumping (*Cubature* discretization).



(a) *Basic* $\mathbb{P}_1, \mathbb{P}_2, \mathbb{P}_3$ elements (from left to right) with the OSS stabilization technique.



(b) *Cubature* $\tilde{\mathbb{P}}_1, \tilde{\mathbb{P}}_2, \tilde{\mathbb{P}}_3$ elements (from left to right) with the OSS stabilization technique.

FIGURE 6.19: 2D view of the free surface elevation at $t = 0.18s$ with the oscillating topography eq. (6.16). All simulation are performed using SSPRK schemes.

6.7 Conclusion

To summarize this chapter, we showed the capacities of stabilized continuous Galerkin methods, using a well-balanced formulation to deal with the topography and adding a shock capturing term (viscosity term) to stabilize shocks and allow to propagate them. The main success of this chapter is the conservation of these properties using mass-lumping with *Cubature* elements. The extension of *Cubature* elements to hyperbolic equations if very promising. Indeed, for coastal engineering applications in operational codes, it should improve considerably the cpu-time. Moreover, these results are particularly interesting in the sense that we proposed a high order, fully explicit, well-balanced and shock capturing method to solve Shallow Water equations, which is mass matrix free. In the next chapter, we propose to evaluate the gain of cpu-time on three coastal engineering test cases in a operational code. These test cases will be perform using a well-balanced discontinuous Galerkin formulation, with the same shock capturing technique and SSPRK methods, for *Basic* and *Cubature* elements.

Chapter 7

Spherical and real benchmarks

Chapter Abstract
In this final chapter, we will mainly show the benefit of using mass lumping in an operational context, by brought light to the gain of time of computation. Only using the DG formulation (for a question of implementation), we compare the time of computation using *Basic* finite elements and *Cubature* elements. We start with two spherical test case from [5] in section 7.2. Then, we perform a submersion benchmark of the Région Nouvelle-Aquitaine in section 7.3.2. The first spherical test case is a steady state which does not use shock capturing technique. For this test case we perform convergence tests and compare numerical discretizations in terms of accuracy and time of computation. The second spherical test represents an unstable jet which travels around the globe. The complexity of this test case is that it creates high vorticity fields during the simulation.
We then finish with a real test case which models the submersion caused by the Xynthia storm (2010) in *"les Boucholeurs"* (Nouvelle-Aquitaine). The complexity of the test case is that it uses the entropy viscosity stabilization technique and has at wet/dry interface. All simulations are performed in the Aerosol code via the Uhaina platform.

Outline

7.1 Introduction to the 3d/2d-covariant formulation 118
7.1.1 Notations 118
7.1.2 The Well Balanced property 119
7.1.3 Fully diagonal mixed 3d/2d-covariant formulation 120
7.2 Global atmospheric tests 120
7.2.1 Steady-State 120
7.2.2 Unstable Jet 124
7.3 A real benchmark: Les Boucholeurs 127
7.3.1 The datas 127
7.3.2 The results 128
7.4 Conclusion 129

7.1 Introduction to the 3d/2d-covariant formulation

In this section firstly introduce the extension to spherical coordinates based on a projection from the cartesian plan to the sphere following the methodology from Arpaia et al. in [5]. Then, in the next section 7.2, we evaluate the proposed methods on standard global benchmarks for the Shallow Water equations on the sphere using *Basic* and *Cubature* discretization. In a first time, we consider a steady state in section 7.2.1, then a perturbation of a steady state in section 7.2.2, which creates high vorticity fields.

In the literature, there are three common approaches used for spherical problems such as the Earth. The first one is based on the two-dimensional parametrization of the sphere through a proper curvilinear coordinate system which means that all the differential operators are transformed in curvilinear coordinates also called the *curved manifold*. This traditional latitude-longitude parametrization is used particularly in regional models. However, it has a singular point in the Arctic Ocean at the North Pole, where the meridians converge [126]. The Jacobian of the coordinate transformation is not anymore defined and the singularity imposes a severe restriction on the maximum time step allowed for stability. The second approach consist to resolve the governing PDEs in a three-dimensional Cartesian framework and then adding a constraint to force the currents to remain tangent to the sphere [41, 54]. Using this approach, no special treatment is required in polar regions to preserve accuracy and to conserve global mass. The third approach combine the advantages of both methods [13]: momentum time derivative is written with respect to 2d components while the right-hand side is first expressed in 3d and then projected back onto the sphere surface by a simple scalar product with respect to the tangent basis. The advantage is that the number of unknown is kept at a minimum (water depth and two momentum components) and, at the same time, the right-hand side maintains Cartesian form, thus it is independent from the parametrization of the sphere, and moreover there is no need to transform differential operators. For example, in tsunami applications this could be laborious for depth averaged non-hydrostatic models with dispersive terms that involve mixed high order derivatives. If requested, Riemann solvers are formulated easily in 3d Cartesian framework, and then projected on the sphere surface along with the right-hand side.

We will mainly follow Arpaia et al. in [5], which proposes some improvements of the third approach.

7.1.1 Notations

Let us introduce the following notation: we consider the sphere $\mathcal{S}^2$ with the radius $\mathcal{R}$ described by curvilinear coordinates X^1, X^2 and othogonal (but not orthonormal) covariant basis g_1, g_2. The coordinate vector $\mathbf{x}$ writes:

$$\mathbf{x} = x^i e_i = x^1 e_1 + x^2 e_2 + x^3 e_3 \tag{7.1}$$

$$= X^\alpha g_\alpha = X^1 g_1 + X^2 g_2 \tag{7.2}$$

We define the spherical transformation $\mathbf{x} = G(X)$, the inverse $X = G^{-1}(\mathbf{x})$ and a *Jacobian*

$$J_G = \frac{\partial \mathbf{x}}{\partial X} \quad \text{and} \quad J_G^{-1} = \frac{\partial X}{\partial \mathbf{x}} \tag{7.3}$$

The covariant vectors define the tangent plane to the sphere. They can be obtained by differentiation as the columns of the Jacobian $g_i = \frac{\partial \mathbf{x}}{\partial X^i}$. We denote by $(g^*)_i$ the normalized basis. Note the $g_i^* \cdot g_j^* = \delta_{ij}$.

For us, X_1 corresponds to the longitude and X_2 the latitude.

Now, rewriting SW equations 6.1 in 3d cartesian coordinates

$$\begin{cases} \partial_t h + \nabla(h\mathbf{u}) & = 0 \\ \partial_t(h\mathbf{u}) + \nabla \cdot \mathbf{T} & = \mathbf{S} \end{cases} \tag{7.4}$$

The source term $\mathbf{S}$ includes the effects of bathymetry, Coriolis force, and meteorological forcing:

$$\mathbf{S} = gh\nabla b + \Omega\mathbf{k} \times h\mathbf{u} + \frac{gh}{\rho_0}\nabla p_{atm} + f_w \tag{7.5}$$

with Ω the Earth rotation rate, $\mathbf{k}$ the Earth rotation axis, p_{atm} the atmospheric pressure, ρ_0 the water density and f_w the wind forcing. The momentum vector $h\mathbf{u}$ can be expressed in both systems as:

$$h\mathbf{u} = hu^i e_i \tag{7.6}$$

$$= hu^\alpha g_\alpha = hu^\alpha \parallel g_\alpha \parallel \frac{g_\alpha}{\parallel g_\alpha \parallel} = hu^{*\alpha} g_\alpha^* \tag{7.7}$$

Note that $u^i = J_G^{i\alpha} u^\alpha$ and $u^i = J_G^{*,i\alpha} u^{*\alpha}$, with $J_G^{*,i\alpha} = J_G^{i\alpha} / \parallel g_\alpha \parallel$ the normalized *Jacobian*. Similarly, we write the flux tensor $\mathbf{T}$ as

$$\mathbf{T} = T^{ij} e_i e_j \tag{7.8}$$

$$= T^{\alpha,\beta} g_\alpha g_\beta = hu^\alpha \parallel g_\alpha \parallel \frac{g_\alpha}{\parallel g_\alpha \parallel} = hu^{*\alpha} g_\alpha^* \tag{7.9}$$

with Cartesian components $T^{ij} = huu^{ij} + P\delta_{ij}$ and curvilinear components $T^{\alpha\beta} = huu^{\alpha\beta} + GP$. The hydrostatic pressure is defined as $P = gh^2/2$ and G the determinant of the metric tensor constructed from the *Jacobian* matrix as $\mathbf{G} = J_G^T J_G$.

7.1.2 The Well Balanced property

As said before, we denote the free surface level $\eta = h + b = const = K = K_0$ and $h\mathbf{u} = 0$. In presence of atmospheric pressure forcing, a relevant state is the *inverted barometer* balance that is an exact solution of the SWEs in case of full adjustment of sea level to changes in barometric pressure:

$$h\mathbf{u} = 0, \quad K = \eta + \frac{p_{atm}}{g\rho_0} = const = K_1 \tag{7.10}$$

The numerical method to solve eq. (7.4) is said *Well-Balanced* if eq. (7.10) is also exact solution of the discrete equations. We write $\nabla \cdot PI = -gh\nabla\left(b + \frac{p_{atm}}{g\rho_0}\right)$.

7.1.3 Fully diagonal mixed 3d/2d-covariant formulation

$$\partial_t(h\mathbf{u}\cdot g_\alpha^*) + (\nabla\cdot\mathbf{T})\cdot g_\alpha^* = \mathbf{S}\cdot g_\alpha^* \tag{7.11}$$

See [5] for the formulation by components α. The main advantage of this formulation with respect to full 3d equations is that we keep the number of unknowns (h, hu^1, hu^2). Another attractive feature is that the flux function is in 3d form and does not depend on a particular transformation. This means that line integrals are defined intrinsically and mass/momentum is easily conserved circumventing implementation issue related to the use of composite meshes in the 2d approach.

The final variational formulation reads

$$\partial_t \int_\Theta h_h\varphi_i d\mathbf{x} + \int_{\partial\Theta} h\mathbf{u}_h'\varphi_i\cdot\mathbf{n}ds - \int_\Theta h\mathbf{u}_h\cdot\nabla\varphi_i d\mathbf{x} = 0 \tag{7.12}$$

$$\partial_t \int_\Theta h\mathbf{u}_h\cdot g_\alpha^*\varphi_i d\mathbf{x} + \int_{\partial\Theta}\mathbf{T}_h' g_\alpha^*\varphi_i\cdot\mathbf{n}ds - \int_\Theta \mathbf{T}_h : \nabla(g_\alpha^*\varphi_i)d\mathbf{x} = \int_\Theta \mathbf{S}_h\cdot g_\alpha^*\varphi_i d\mathbf{x} \tag{7.13}$$

with Θ the domain in the CG context, and $\Theta = \mathcal{K}$ a triangle in the DG context. T_h' denotes the numerical flux evaluated at the elements boundaries, and the symbole ":" correspond to the scalar product between second order tensors $\mathbf{A} : \mathbf{B} = A_{ij}B_{ij}$.

This method has been validated in [5] on academic benchmarks involving both smooth and discontinuous solutions. Following global atmospheric tests are taken from Williamson et al. [127] for the first and J. Galewsky et al. [51] for the second one. The numerical tests are performed using the *Aerosol* software.

7.2 Global atmospheric tests

7.2.1 Steady-State

It corresponds to a Global Steady-State. This exact steady geostrophic equilibrium allowing to measure the order of accuracy in presence of Earth rotation. We define the Eart radius $\mathcal{R} = 6371.22\times10^3 m$, the gravitationnal constant $g = 9.80616 m^2 s^{-1}$, the rotation rate parameter $\Omega = 7.295\times10^{-5}s^{-1}$ and the water density $\rho_0 = 1025 kg/m^3$. The velocity and height fields are initially given by:

$$h(\mathbf{x},0) = h_0 - \frac{1}{g}\left(\Omega\mathcal{R}u_0 + \frac{u_0^2}{2}\right)(-\cos\lambda_2\cos\lambda_1\sin\alpha + \sin\lambda_2\cos\alpha)^2 \tag{7.14}$$

$$u^{*1}(\mathbf{x},0) = u_0(\cos\lambda_2\cos\alpha + \cos\lambda_1\sin\lambda_2\sin\alpha) \tag{7.15}$$

$$u^{*2}(\mathbf{x},0) = -u_0\sin\lambda_1\sin\alpha \tag{7.16}$$

$$\text{with}\quad (\lambda_1,\lambda_2) = T_{Lat,Long}(\mathbf{x}),\ \alpha \text{ the rotation angle,} \tag{7.17}$$

where $T_{Lat,Long}$ the transformation from Cartesian to *Latitude/Longitude*, $gh_0 = 2.94\times10^4 m^2 s^{-2}$, $u_0 = \frac{2\pi\mathcal{R}}{12\times days}$ and $\alpha = 0$. A representation of the initial condition is given in fig. 7.5.

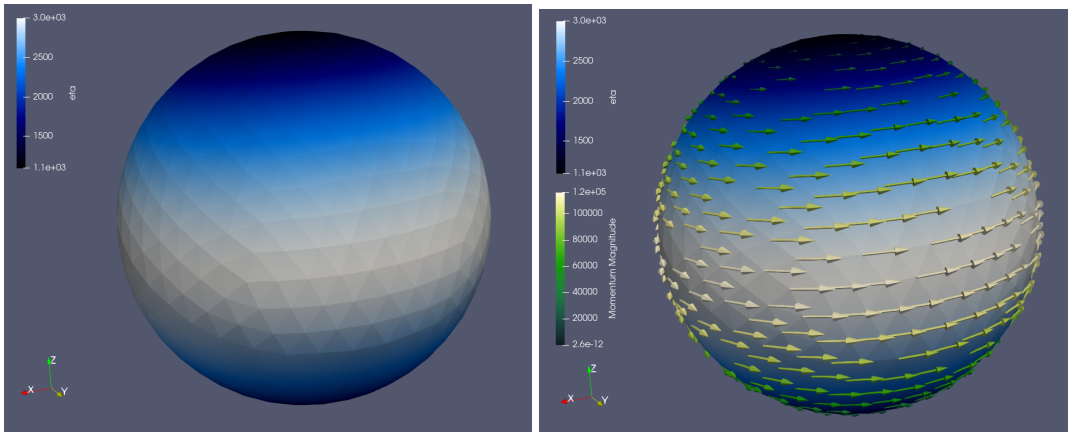
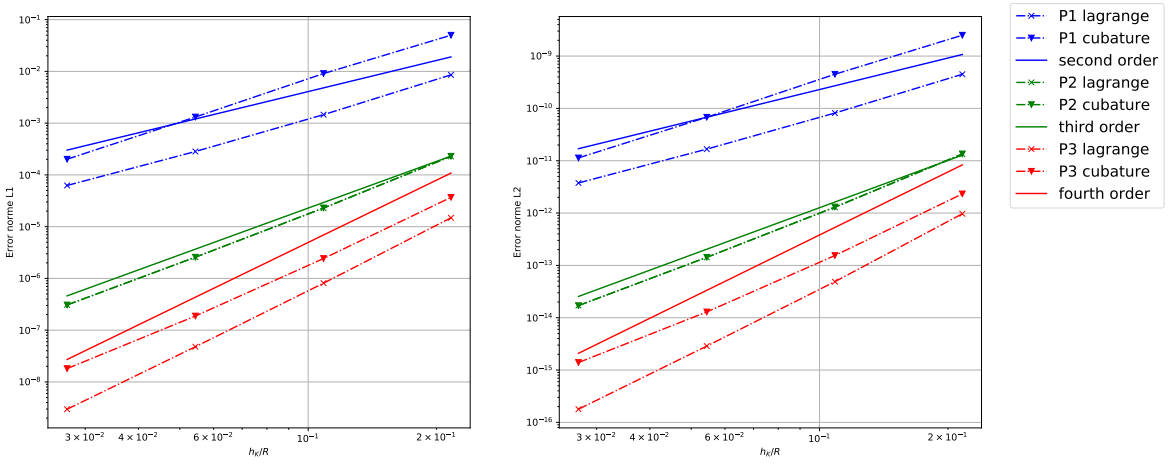


FIGURE 7.1: Initial state of the Global atmospheric test at left, representation of velocity field at right. The green scale represents the momentum magnitude.

This comparison will be performed using the DG formulation. The time of the simulation is 5 days, i.e. $t_f = 432000s$. To show the benefit of the use of *Cubature* elements and *Mass-Lumping*, we compare the error and the time of computation using *Basic* and *Cubature* elements. Grid convergence are defined by 3 levels: from $h_k = 446km$ to $h_k = 1785km$. Convergence curves are compared in fig. 7.2(a) for the L_1 error and fig. 7.2(b) for the L_2 error, then we summarize error, cpu-time and convergence order in table 7.1. The relative error is computed as follow: $e_p = \frac{\|h-h_{ex}\|_{L^p}}{\|h_{ex}\|_{L^p}}$.



(a) Using the L_1 error.

(b) Using the L_2 error.

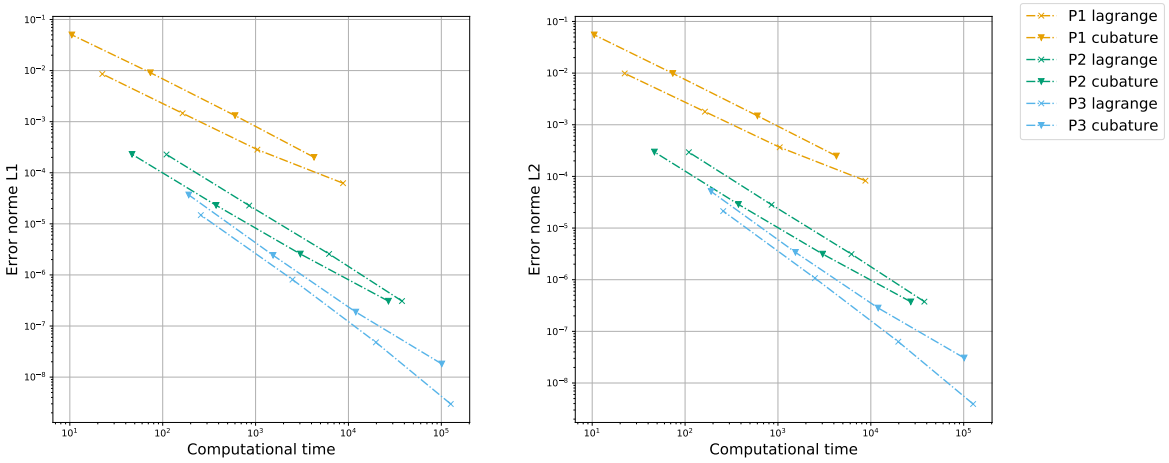
FIGURE 7.2: Global steady state simulation, convergence rates.

The convergence orders are relatively satisfying comparing to what we were expecting, i.e. order $p+1$ using $\mathbb{P}_p$ discretization. However, we can clearly see in fig. 7.2 and in table 7.1 that using $\tilde{\mathbb{P}}_1$ and $\tilde{\mathbb{P}}_3$ *Cubature* elements leads to a loose of accuracy for the same mesh length ($\approx$ factor 10). Results using the $\tilde{\mathbb{P}}_2$ *Cubature* discretization are very positive and promising for the future. Moreover, even the loose of accuracy for $\tilde{\mathbb{P}}_1$ and $\tilde{\mathbb{P}}_3$, we show in fig. 7.3(a) and fig. 7.3(b) an evaluation of the error relatively to the cpu-time in order to

Element & Time scheme		e_1 $\mathbb{P}_1$	cpu-time (s)	e_1 $\mathbb{P}_2$	cpu-time (s)	e_1 $\mathbb{P}_3$	cpu-time (s)
Basic SSPRK	$h_K = 1785km$	8.57e-3	22	2.28e-4	110	1.48e-5	257
	$h_K = 893km$	1.46e-3	164	2.28e-5	854	8.11e-7	2498
	$h_K = 446km$	2.83e-4	1045	2.57e-6	6189	4.79e-8	19772
	$h_K = 223km$	6.25e-5	8746	3.08e-7	37728	2.97e-9	126304
	convergence order	2.37		3.18		4.11	
Cub. SSPRK	$h_K = 1785km$	5.00e-2	10	2.30e-4	47	3.69e-5	191
	$h_K = 893km$	9.08e-3	74	2.30e-5	376	2.42e-6	1543
	$h_K = 446km$	1.31e-3	601	2.57e-6	3035	1.88e-7	12002
	$h_K = 223km$	2.01e-4	4262	3.07e-7	26960	1.81e-8	101206
	convergence order	2.67		3.19		3.68	

TABLE 7.1: Global steady state. Summary tab of numerical results. At left: the numerical schemes, at right: results.

show the benefit of mass-lumping in term of time of computation.



(a) Using the L_1 error.

(b) Using the L_2 error.

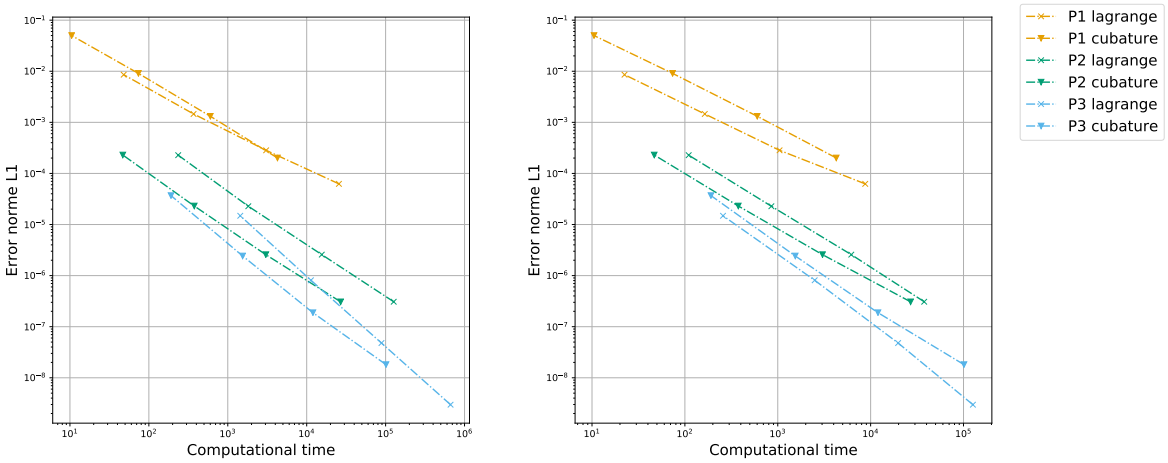
FIGURE 7.3: Global steady state simulation, error with respect to computational time.

As expecting, $\tilde{\mathbb{P}}_2$ discretization leads to a much better ratio error/cpu-time than $\mathbb{P}_2$ discretization. Moreover, the ratio error/cpu-time for $\mathbb{P}_1$ and $\mathbb{P}_3$ seems give the advantage to the *Basic* discretization because of the loose of accuracy using mass-lumping. This result is a bit disappointing comparing to CG results, but can be easily justified by the fact that for a DG approach, whatever the spatial discretization, we can solve the numerical system using a block-diagonal solver which allows to optimize considerably the cpu-time.

NB: The previous remark concerning the solver used and the gain of cpu-time can be brought to light by comparing cpu-time using the Petsc and the Block-diagonal solvers of Aerosol. This comparison is done in fig. 7.4. As before, with the Petsy solver, we can see in fig. 7.4(a) that the $\tilde{\mathbb{P}}_2$ discretization leads to a much better ratio error/cpu-time than $\mathbb{P}_2$

discretization. Then, using $\mathbb{P}_1$ and $\tilde{\mathbb{P}}_1$ discretization, the ratio seems equivalent because of the loose of accuracy using mass-lumping for $\tilde{\mathbb{P}}_1$. We can also notice that the *Cubature* $\tilde{\mathbb{P}}_1$ goes under the $\mathbb{P}_1$ curve at cpu-time $= 4 \times 10^3 s$ which means than using a thinner mesh, the *Cubature* $\tilde{\mathbb{P}}_1$ discretization should be better than $\mathbb{P}_1$.

In the same direction, even the loose of accuracy using $\tilde{\mathbb{P}}_3$ discretization, the ratio error/cpu-time is better than using $\mathbb{P}_3$ discretization in fig. 7.4(a).



(a) Using the Petsc solver.

(b) Using a Block-diagonal solver.

FIGURE 7.4: Global steady state simulation, L_1 error with respect to computational time. Comparison of the use of two different solvers.

To finish the study of this test case, we also give an approximation of proportion of time spent in the computation of the residual vector, and in the mass matrix inversion in table 7.2. The rest of the time is spent in post processing.

Element & Time scheme		Total cpu-time (s)	Residual computation (s)	Limit solution (s)	Mass matrix inversion (s)
Basic SSPRK	$\mathbb{P}_1$	8746	6130 (70.1%)	1707 (19.5 %)	34 (0.39 %)
	$\mathbb{P}_2$	37728	22945 (60.8 %)	9512 (25.2 %)	158 (0.42 %)
	$\mathbb{P}_3$	126304	64478 (51.0%)	33967 (26.9 %)	768 (0.61 %)
Cub. SSPRK	$\tilde{\mathbb{P}}_1$	4262	2768 (64.9 %)	940 (22.1 %)	10 (0.23 %)
	$\tilde{\mathbb{P}}_2$	26960	16429 (60.9%)	8185 (30.4 %)	57 (0.21 %)
	$\tilde{\mathbb{P}}_3$	101206	53140 (52.5 %)	41375 (40.9 %)	243 (0.24 %)

TABLE 7.2: Global steady state. Summary tab of cpu-time repartition for the mesh $h_k = 223km$, in parenthesis the pourcentage).

As expecting, regarding table 7.2, using the *Cubature* discretisation leads to a lower cpu-time. Indeed for a same order of discretization, the time spent in the residual computation are quite similar, this can be explained by the fact that *Cubature* elements have more DOF, but allows to use quadrature formulas using less quadrature points than the *Basic* discretization which uses high order quadrature formula. Moreover, even is the mass matrix inversion is

computed using a block diagonal solver for the *Basic* discretization, using the mass lumping allows to decrease considerably the time of computation by two using the *Cubature* discretization which is very promising considering the error obtained using $\tilde{\mathbb{P}}_2$. Furthermore, for all *Cubature* discretization, this result is very promising in an operational context where we work with a given mesh and we need fast and accurate results.

Remark: For both discretizations, a huge percentage of the cpu-time is spent in the limiting of the solution. This point will be discussed later.

7.2.2 Unstable Jet

For this last spherical test introduced in [51] and also used in [5], we consider a geostrophically mid-latitude jet, i.e. a small perturbation is added at the initial condition and we propagate it. The zonal velocity component u is a function of latitude:

$$u(\mathbf{x}) = u(\lambda_1) = \begin{cases} 0 & \text{for} \quad \lambda_1 \leq \psi_0 \\ \frac{u_{max}}{e_n} exp\left(\frac{1}{(\lambda_1-\psi_0)(\lambda_1-\psi_1)}\right) & \text{for} \quad \psi_0 < \lambda_1 < \psi_1 \\ 0 & \text{for} \quad \lambda_1 \geq \psi_1 \end{cases} \tag{7.18}$$

with $(\lambda_1, \lambda_2) = T_{Lat,Long}(\mathbf{x})$, u_{max} the maximum zonal velocity, ψ_1 the latitude of the northern boundary of the jet in radians, ψ_0 the latitude of the southern boundary of the jet in radians, and e_n a non-dimensional parameter that normalizes the magnitude of the jet to a value of u max at the jet's mid-point. From [51], we choose $u_{max} = 80ms^{-1}$, $\psi_0 = \pi/7$, $\psi_1 = \pi/2 - \psi_0$, $e_n = exp(-4(\psi_1 - \psi_0)^2)$ for which the jet's mid-point is located at $\lambda_1 = \pi/4$.

Considering the initial zonal flow from eq. (7.18), the water height h can be described by

$$gh(\mathbf{x}) = gh(\lambda_1) = gh_0 - \int^{\lambda_1} \mathcal{R}u(l)\left(f + \frac{\tan(l)}{\mathcal{R}}u(l)\right) dl \tag{7.19}$$

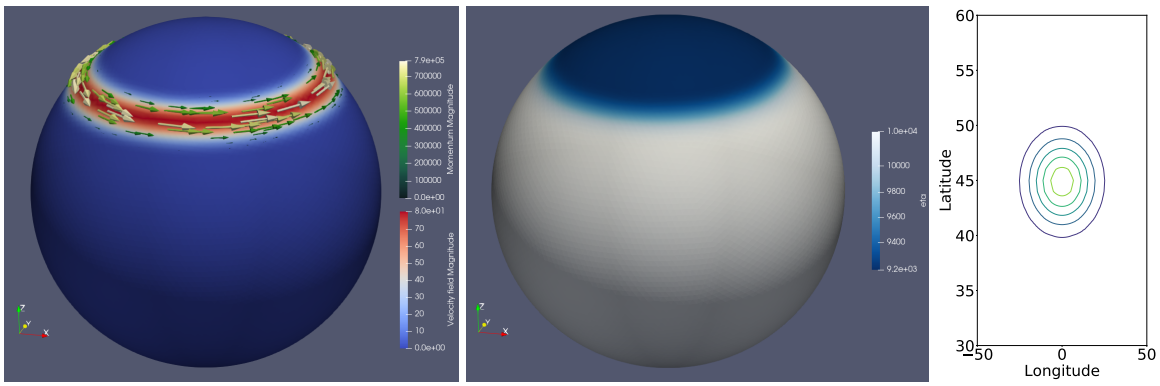
with h_0 is chosen so that the global mean layer depth is equal to $10km$. We now initiate the barotropic instability by adding a perturbation / localized bump:

$$h'(\mathbf{x}) = h'(\lambda_1, \lambda_2) = \begin{cases} \hat{h}\cos(\lambda_1)e^{-(\lambda_2/\alpha)^2}e^{-(\psi_2-\lambda_1)^2/\beta^2} \\ \text{for} \quad -\pi < \lambda_2 < \pi \end{cases} \tag{7.20}$$

where $\psi_2 = \pi/4$, $\alpha = 1/3$, $\beta = 1/15$ and $\hat{h} = 120$.

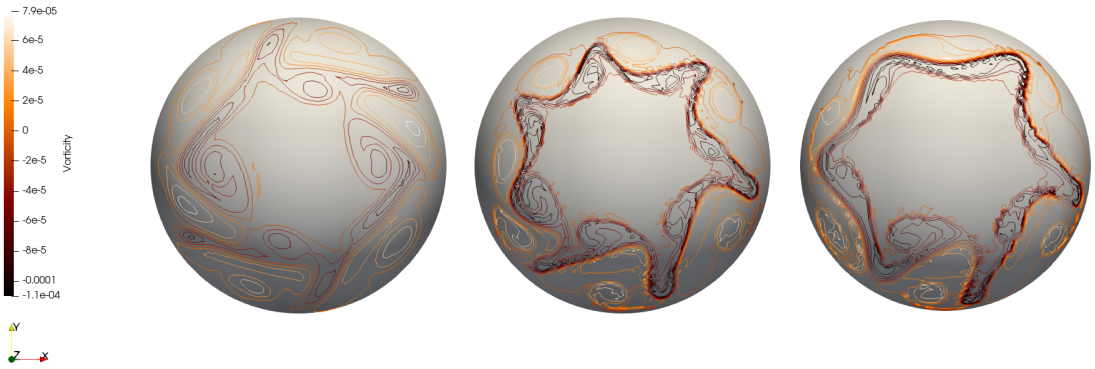
A representation of the initial state is shown in fig. 7.5 for the velocity field, the water elevation and the perturbation.

We look to the vorticity field (i.e. the local rotational motion) using *Basic* and *Cubature* $\mathbb{P}_p$ discretisations ($p = 1, 2, 3$) using the mesh size $h_k = 223km$.A representation of contour levels is showed in fig. 7.6, and compared with results from J. Galewsky et al. [51] in fig. 7.7.We then compare the cpu-time for all methods. The time of the simulation is 6 days

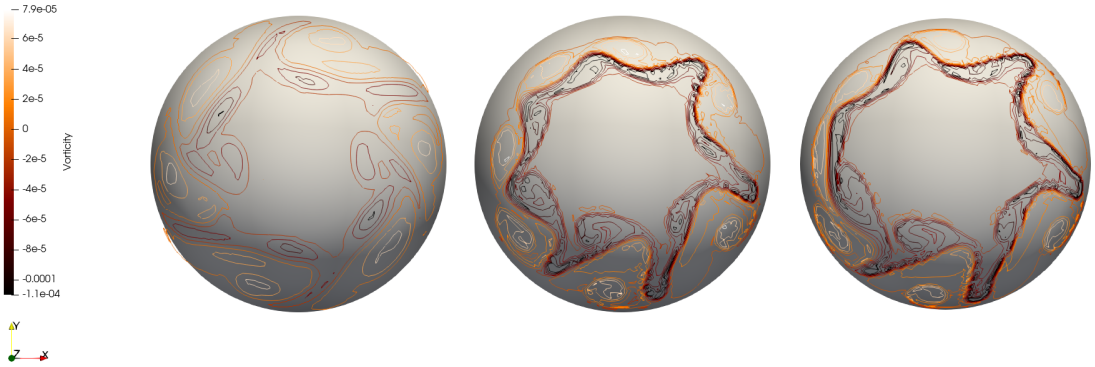


(a) Velocity field from eq. (7.18). (b) Water elevation from eq. (7.19). (c) Initial perturbation from eq. (7.20).

FIGURE 7.5: Initial state of the Unstable Jet test.



(a) *Basic* $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ elements (from left to right).



(b) *Cubature* $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ elements (from left to right).

FIGURE 7.6: Unstable Jet - 2D view from the northen pole of the vorticity field contour levels (from $-1.1e-4$ to $1.5e-4$) at $t_f = 6$ days. $h_K = 223km$

($t_f = 518400s$).

With a mesh size of $h_K = 223km$, we can see in fig. 7.6 that for both discretization, solution obtained with P_1 is not accurate enough to propagate the vortex evolution. But with

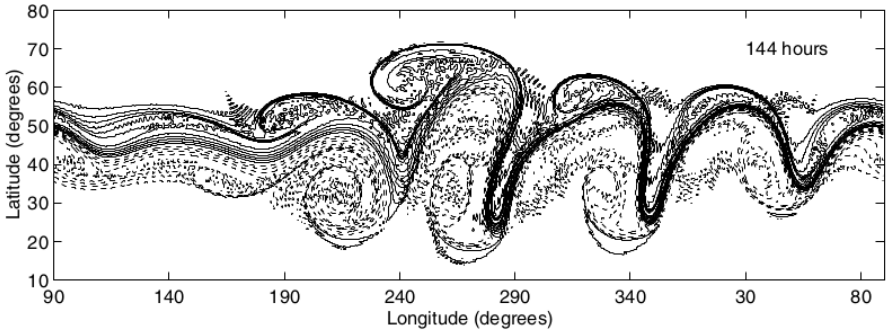


FIGURE 7.7: Approximated solution of the vorticity field after 6 days. Sources from J. Galewsky et al. [51].

$\mathbb{P}_2$ and $\mathbb{P}_3$ elements, we obtain the correct dynamic proposed in [51, 5] (see fig. 7.7). Moreover, the behaviour of the solution seems to be the same between the *Basic* and the *Cubature* discretization. We note that refining the mesh size to $h_k = 111km$ as it is done in [5], it allows to draw more properly contour line and approximate better the solution.

We now analyze the time of computation for both spatial discretizations in table 7.3.

Element & Time scheme		Total cpu-time (s)	Residual computation (s)	Limit solution (s)	Mass matrix inversion (s)
Basic SSPRK	$\mathbb{P}_1$	12701.6	8635.9 (70.0%)	2421.1 (19.0%)	99.4 (0.78%)
	$\mathbb{P}_2$	64081.9	40776.7 (63.6%)	17791.2 (27.8%)	318.4 (0.50%)
	$\mathbb{P}_3$	247554.0	146648 (59.2%)	84582.7 (34.2%)	1612.0 (0.65%)
Cub. SSPRK	$\tilde{\mathbb{P}}_1$	10686.3	6783.08 (63.5%)	2340.7 (21.9%)	24.6 (0.23%)
	$\tilde{\mathbb{P}}_2$	64890.2	39260.7 (60.5%)	19503.2 (30.1%)	132.3 (0.20%)
	$\tilde{\mathbb{P}}_3$	257276.7	138813 (54.0%)	100132 (38.9%)	619.3 (0.24%)

TABLE 7.3: Unstable Jet. Summary tab of cpu-time repartition for the mesh $h_k = 223km$, in parenthesis the percentage. $t_f = 6$ days.

Concerning the cpu-time, several points can be discussed. In particular:

- As we showed in the previous test case (see table 7.2), the time spent for the mass matrix inversion is approximately 2 times lower for *Cubature* elements with mass-lumping which is very encouraging. Then, we again obtain approximately the same cpu-time spent in the residual computation for *Basic* and *Cubature* element with a small advantage for *Cubature* elements which use less quadrature points.

- A second observation which is not advantageous for *Cubature* elements is the time spent in the *limit solution* computation. Indeed, for $\tilde{\mathbb{P}}_1$, only 21.9% against 30.1% for $\tilde{\mathbb{P}}_2$ and almost 40% for $\tilde{\mathbb{P}}_3$ elements. These percentages are very high comparing to *Basic* elements and are also observed for the smooth steady case in table 7.2. This point deserves to be explored and optimized in Aerosol. We can also notice that for lower physical time of simulation, even for the same cpu-time repartition, *Cubature* elements allow to reduce the time of computation. To illustrate this assumption, we re-do the same test in same conditions with $t_f = 96h$. Results are available in Appendix F.2.

This choice is made regarding results from [51] which show that the vorticity field appears after 4 days. The same results are also obtained previously in the steady case (see table 7.2)

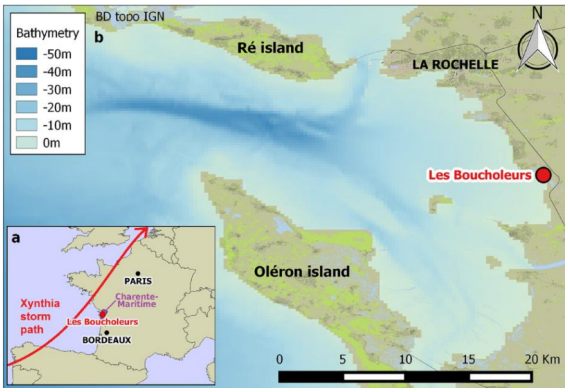
- Finally, for this test, the total cpu-time using element of degree 1 gives the advantage to *Cubature* elements. However, using elements of degree 2 and 3, the advantage goes to *Basic* discretization. This last point may depend on the final time of the simulation, for shorter times of simulation, the difference in terms of cpu-time between the resolution of the full mass matrix and *Cubature* approach being smaller (see table 7.1 and table F.1 for more comparisons).

7.3 A real benchmark: Les Boucholeurs

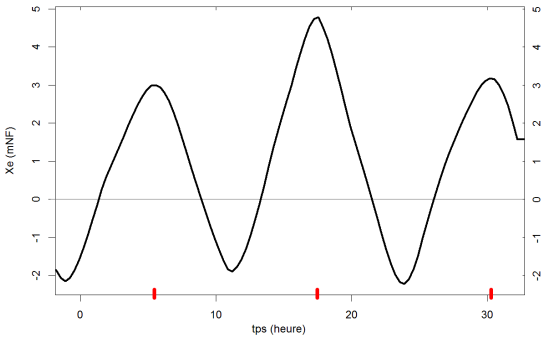
7.3.1 The datas

This benchmark corresponds to the submersion during the Xynthia storm, the 28^{th} of February 2010. A representation of the storm trajectory is shown in fig. 7.8(a). All data and results come from the report [103]:

1. The topography: 1m of resolution + probes *SHOM*[1].
2. Input: water height evaluate by harmonic analysis (see Pedreros & Paris (2012)) represented in fig. 7.8(b). The water elevation is introduced by the West boundary. The physical final time of the simulation is $t_f = 72000s = 20h$.
3. Domain of computation: $\approx 36km^2$, resolution $\approx 25m$ for the sea and $\approx 1m$ for road



(a) Location of the Boucholeurs and trajectoria of the Xynthia storm. Source Müller et al. 2016 [95].



(b) Temporal sea elevation introduced by the west boundary. Source Brgm. Red points on the temporal axis correspond to the high tides.

FIGURE 7.8: Representation of the test case: Les Boucholeurs (Nouvelle-Aquitaine).

NB: The high tides are at approximately 5h30, 18h and 31h30 (in the Atlantic face of France, the tide has a periodicity of approximately 12h30).

We are able to reproduce numerically particularly well the flooding, comparing to debris lines (seaweeds, sea muds). The snapshot views of the flooding are reported in fig. 7.9.

[1] https://www.shom.fr/

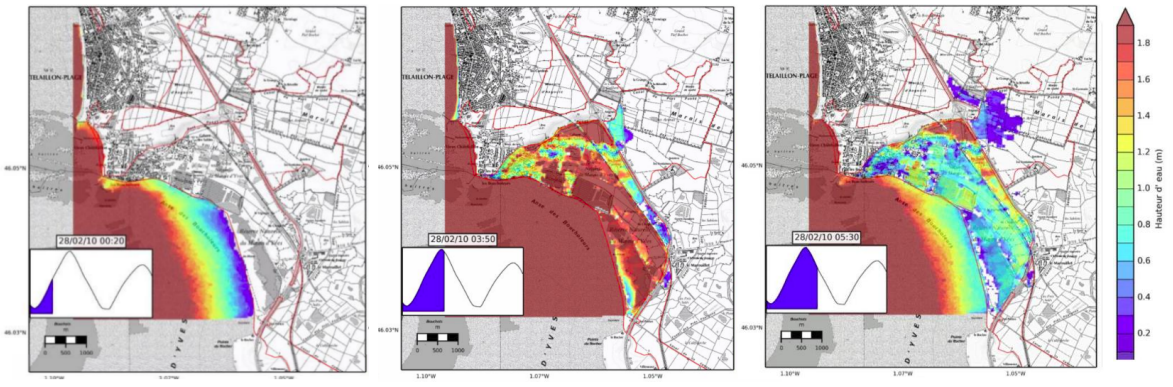


FIGURE 7.9: Snapshot views of the marine submersion in the Boucholeurs sector during the Xynthia storm at different hours: at left 00:20 am ($t_s \approx 14h$), on the middle 03:50 am ($t_s \approx 18h$), at right 05:30 am ($t_s \approx 20h$). Source Brgm, model MARS-2DH.

7.3.2 The results

The main objective of this study will be to compare the water elevation and the flooding using the mass lumping method for a very complex test case (complex topography, adaptive mesh, wet/dry interface, flooding, etc). Then, we compare the time of computation for this test case in order to see if the use of mass lumping allows to be faster without loosing in accuracy. This comparison will be performed using the DG entropy conservative formulation in the Aerosol code (defined in section 3.3), using 128 processors. Because of the complexity of the test case, we run it only using $\mathbb{P}_1$ elements. We report cpu-time for both simulation in table 7.4. The number of triangles is 194407.

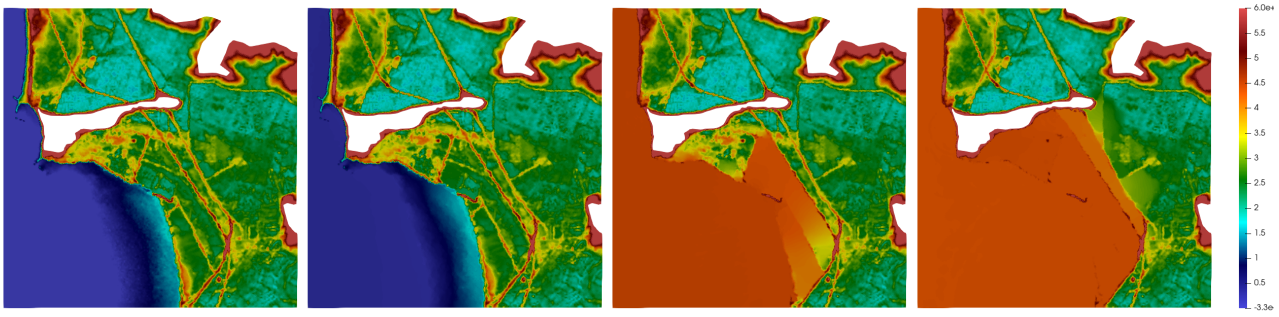
Element & Time scheme	Basic $\mathbb{P}_1$ SSPRK	Cub. $\tilde{\mathbb{P}}_1$ SSPRK
Mass-matrix inversion	1576.01 (1.30 %)	1572.44 (1.30 %)
Residual computation	57399.9 (47.3 %)	56739.8 (47.0 %)
Compute entropy viscosity terms	30393.8 (25.1 %)	30721.8 (25.4 %)
Limit the solution	20274.7 (16.7 %)	20144.1 (16.7 %)
Total cpu-time	121260	120840

TABLE 7.4: Summary tab of numerical results. At left: the numerical schemes, at right: results.

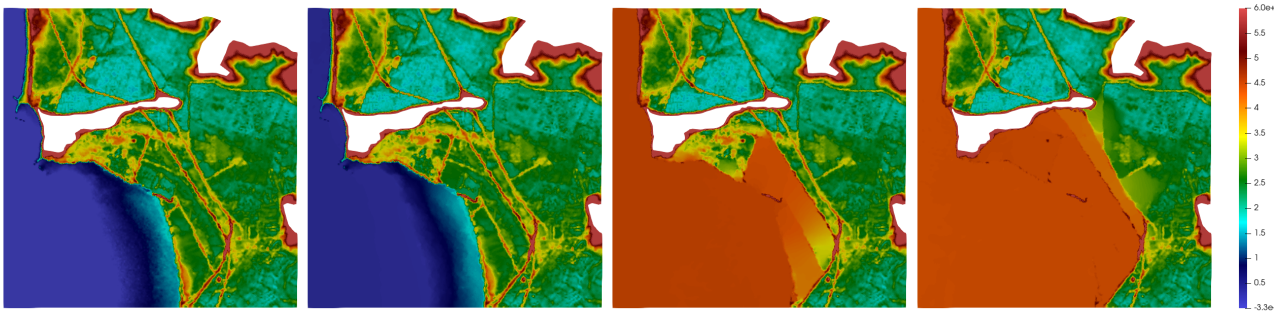
As we can see in table 7.4, using the mass-lumping allow to save $\approx 2500s$ for a simulation of $\approx 36h$. This is not as much as we were expecting, considering results obtain previously for the global atmospheric test (see table 7.1). Indeed, the benefit of using *Cubature* elements is only highlighted in the Residual and the viscous terms computations. There is no benefit from the inversion of the mass matrix. The benefice of the use of mass lumping can be dumped by two different points in the total cpu-time:
1/ the cpu-time spent in the "computation of the entropy viscosity terms" and "limit the solution" which is very high for both discretization ($\approx 40\%$) of the total cpu time.
2/ the cpu-time spent in the processor communication for the inversion of the mass lumping. Indeed, seeing the percentage of the cpu-time spent in the Mass-matrix inversion, comparing

to other simulation performed with the same code, we should be lower than 1% for the *Basic* discretization and lower than 0.25% for *Cubature* discretization.

We now look at the submersion at four different simulation times: $t_s \approx 0h$, $14h$, $18h$, and $20h$. A representation is shown in fig. 7.10(a) using *Basic* $\mathbb{P}_1$ elements and in fig. 7.10(b) using *Cubature* $\tilde{\mathbb{P}}_1$ elements.



(a) *Basic* $\mathbb{P}_1$, $t_s \approx 0h$, $14h$, $18h$, $20h$ (from left to right).



(b) *Cubature* $\tilde{\mathbb{P}}_1$, $t_s \approx 0h$, $14h$, $18h$, $20h$ (from left to right).

FIGURE 7.10: Les Boucholeurs simulation, Xynthia storm (2010) - 2D view from the top of the water elevation $\eta = h + b$ at different time steps.

As we can see in fig. 7.10, the submersion obtained with both discretizations is similar. Moreover, they reproduce the same behaviour than the one observed in fig. 7.9. We note that in fig. 7.9, a submersion appears at the top-right of the road (described in blue) and is not visible with our simulation in fig. 7.10. This submersion is possible thanks to the hydrolic connections which are implemented in the BRGM model MARS-2DH, but not in Aerosol. For this case, we can say that the use of mass-lumping allows to reduce the cpu-time, which is very promising for the future, knowing that the Aerosol code is not already optimized for *Cubature* elements. Indeed, the facts that quadrature nodes match with degree of freedom, and that $\phi_i(x_j) = \delta_{ij}$, numerous multiplication by zeros are hidden (in the residual computation as example), and can be removed.

7.4 Conclusion

In this chapter, we compare our spatial discretizations (*Basic* and *Cubature*) using the discontinuous Galerkin approach. We note that in a DG context, the numerical solver used for *Basic* discretizations is a block-diagonal solver which is already optimized compared to

sparse solver used in the CG formulation. Even considering the block-diagonal solver, we showed the benefice in using mass-lumping in an operational context. Indeed, we shown in section 7.2.1 an important reduction of the cpu-time by two for a long time (and spherical) smooth test cases, which allows to keep the order of accuracy. However, we also showed that the error becomes higher enough to consider the ratio *error/cpu-time* better using the classical discretization than using the *Cubature* one. Then, for a longer and unsteady simulation with the same mesh in section 7.2.2, the cpu-time repartition is the same. However, we observed disadvantage for *Cubature* $\tilde{\mathbb{P}}_2$ and $\tilde{\mathbb{P}}_3$ elements and mass-lumping. Indeed, the cpu-time spent in the limitation of the solution increase considerably with the degree of elements, and so for longer time simulation, *Cubature* elements do not allows to be faster anymore. An optimization of the actual limiter or a possibility to switch of limiters seems to be necessary for smooth cases. This could optimizes considerably the cpu-time. In particular, the actual limiter loops over elements and degree of freedoms four times:
1/ dry cells: cutOff correction on velocity,
2/ wet/dry detection,
3/ shock limiter (Barth and Jespersen Limiter [9]),
4/ Positivity Preserving Limiter (Zhang and Shu Limiter [131, 132]).

Finally, we proposed in section 7.3.2 a comparison of both spatial discretizations ($\mathbb{P}_1$ basic and $\tilde{\mathbb{P}}_1$ *Cubature* elements) for complex test cases with flooding (wet/dry interface) and with a complex topography. We showed again the benefit of using mass-lumping for *Cubature* elements in terms of cpu-time. However, when adding the shock capturing operator, and using positivity preserving limiter this benefit is masked from the cost of these two. We also saw that mass-lumping does not impact the submersion approximation and allow to obtain accurate results comparing to results obtained by the BRGM (numerical results and submersion observations). An optimization of the code could be interesting for these elements and easy to implement in the Aerosol code.

In conclusion, when considering the DG solver, when using lower order elements ($\mathbb{P}_2$ at most), there is an interest in using the *Cubature* approach in terms of cpu-time. This is not true anymore for higher polynomial degrees, especially when propagating smooth data for long times.

Chapter 8

Conclusions and outlook

Outline

8.1 The von Neuman numerical analysis 131
8.2 Shallow water equations and applications 132
8.3 Future investigations and developments 133

In this thesis we have studied several fully explicit numerical methods for hyperbolic equations. The underlying objective is to achieve a continuous finite element type method with a structure similar to discontinuous Galerkin. To this end, the use of *Cubature* elements combined with symmetric stabilization techniques is explored as a path toward fully explicit, mass matrix free high order methods.

Several elements play an important role in this construction, and in this PhD we have tried to account for the most important, namely:

1. the use of different stabilization approaches, using residual based strategies, or solution/gradient variations (chapter 3),
2. the use of different one-step time integration methods (RK or DeC - chapter 3),
3. the linear stability of all the possible combinations for an appropriate choice of the stabilization parameters (chapter 4 and chapter 5),
4. an extension to the nonlinear Shallow Water equations with variable topography guaranteeing the stable approximation of moving bores, as well as well balanced (chapter 6),
5. some applications to realistic coastal engineering simulations (chapter 7).

A summary of these contributions, with some perspectives is provided in this final chapter.

8.1 The von Neuman numerical analysis

The global work done for the von Neuman analysis applied to linear advection equation in chapter 4 and chapter 5 is to evaluate the stability and the dispersion of numerical methods considering different parameters (CFL,δ) and optimize them taking into account dispersion and solution errors. The Fourier analysis has been developed progressively as a spatial-eigenanalysis, a temporal-eigenanalysis and finally a spatio-temporal-eigenanalysis to optimize the CFLs and penalty coefficients δ. We proposed one methodology for the one dimensional case that we extended and improved for the two-dimensional space.

As a scientific contribution, this work has been performed comparing different time integration methods, spatial discretizations and stabilization techniques to provide reliable parameters and CFL conditions. Firstly in the mono-dimensional context and secondly extended to the two-dimensional approach using different triangular mesh configurations.

Based on the results of the analysis, we have performed a comparison of our numerical schemes in terms of accuracy and time of computation. In particular, we showed the potential of the mass matrix free approach proposed in minimizing the cpu-time for a given error level.

8.2 Shallow water equations and applications

Still in a Continuous Galerkin context, we have proposed in chapter 6 a well balanced extension to the Shallow Water equations, using an entropy viscosity operator to stabilize bores and hydraulic jumps. Moreover, we proposed a comparison of our numerical schemes in terms of accuracy and time of computation. As expected, *Cubature* elements using mass lumping showed very good results comparing with *Basic* elements. The validation performed for the mass matrix free *Cubature* continuous finite elements is very promising for, in general hyperbolic equations, and in particular Shallow Water equations.

In the last chapter 7 we investigate the benefit of using mass-lumping in discontinuous Galerkin context. In this case, the basic Lagrange approximation already provides a block diagonal mass matrix, which is relatively easy to invert. For smooth solution, *Cubature* elements allow to achieve a reduction of the computational time by a factor of 2. However, when comparing the ratio *error/cpu-time*, with regard *Cubature* $\tilde{\mathbb{P}}_1$ and $\tilde{\mathbb{P}}_3$ elements, the advantage goes to standard elements which allow to better approximate the numerical solution. Moreover, especially for realistic applications, the inversion of the mass matrix is not the computationally most demanding step (as e.g. compared to the evaluation of the entropy viscosity). So for long time simulations with higher order elements (at least three), and for realistic applications the *Cubature* DG approximation is not of great interest.

Finally, in section 7.3.2 we performed a comparison of both spatial discretizations ($\mathbb{P}_1$ *Basic* and $\tilde{\mathbb{P}}_1$ *Cubature* elements) for complex test cases with flooding (wet/dry interface) and with a complex topography. We showed again the benefit of using mass-lumping for *Cubature* elements in terms of computational time. In this case this is expected, as the cubature approximation uses the same space with a considerable reduction of quadrature points. We also saw that mass-lumping does not impact the submersion approximation and allow to obtain accurate results comparing to results obtained by the BRGM (numerical results and submersion observations). An optimization of the code could be interesting for these elements and easy to implement in the Aerosol code co-developed with BRGM via the Uhaina platform.

As a first conclusion, using the discontinuous Galerkin formulation in the Aerosol code, for benchmarks using elements of degree 1, *Cubature* elements seems always better to use. However, for long time simulation using high order elements, *Basic* discretization provides the lower cpu-time. Furthermore, in an operational context, the topography is generally not constant, the mesh is generally given, and discontinuities can appear in the solution during the simulation and force to go back to first or second order approximation. In this context, we showed in section 7.3.1 that using mass-lumping allows to reduce the cpu-time. This result

is very promising, because the Aerosol code is not already optimized for *Cubature* elements. Indeed, as we said before, the fact that quadrature nodes match with degree of freedom, and that $\phi_i(x_j) = \delta_{ij}$, numerous multiplication by zeros are hidden (in the residual computation as example), and can be removed.

As a second conclusion, using the continuous Galerkin formulation, we showed several results to see the benefit of *Cubature* elements for hyperbolic equations. It is clear that these elements allow to decrease considerably the cpu-time. All our von Neumann analysis presented in this manuscript allowed to characterize the stability of our numerical models, and so "tun" parameters to obtain very efficient schemes. This global study allowed in particular to propose a high order and mass matrix free method for complex test cases, such as for coastal engineering applications.

8.3 Future investigations and developments

This PhD belongs in the continuity of the CARDAMOM team researches in two aspects: the analysis of PDE and the coastal engineering applications through the development of the Aerosol code and the collaboration with the BRGM (via Uhaina).

Concerning the analysis of hyperbolic equations, several additional things can be done, in particular it would be interesting to evaluate the stability of the CIP using additional penalty terms on the jump of higher order derivatives as suggested in [26, 22, 107]. Then, the non-convergence for unsteady case of the DeC time integration scheme, combined with *Berntein* $\mathbb{B}$ elements, has to be clarified. Finally, in the continuity of the analysis, it could be interesting to generalize the analysis to higher degrees of approximations, based on the *Cubature* elements proposed by Mulder [93, 94], and see if the mass-lumping provides as good results as we obtained for elements of degree $p \leq 3$..

Concerning applications, we proposed in the two last chapters, two different axis. Firstly, a stable, shock capturing and well-balanced formulation using continuous *Cubature* elements. This formulation can be used for continuous and discontinuous Galerkin formulation. In this direction, it could be interesting to test different shock capturing techniques more optimal in terms of algorithm cost and implementation. Indeed, we saw that the viscous term computation is costly comparing to the total cpu-time (25% for $\mathbb{P}_1$ elements in Aerosol with the DG formulation), and need numerous parameters (α, β, κ, N, etc.. (see section 3.3)). In a second time, we proposed to highlight the gain of cpu-time using the mass-lumping procedure with *Cubature* elements in a DG context. In this direction, it seems obvious and necessary to optimize the computation of all terms for the *Cubature* discretization knowing that nodes of quadrature formula match with degree of freedom of elements, which hides numerous multiplications by zero. Then a change or an optimization of the actual limiter is necessary to do for smooth solutions.

Finally, the Aerosol code is written for discontinuous Galerkin formulation (the algorithmic is optimized for DG). This fact implies that it is not relevant to compare the continuous and the discontinuous methods in terms of cpu-time. Moreover, several additional implementations and tests has to be performed to validate the CG formulation in Aerosol before using it as reference. It is also possible to compare these CG formulations using another

operational codes written and optimized for CG, to validate our formulations on complex test cases.

Appendix A

Cubature elements, definition and construction

In this section we give a description of the *Cubature* finite elements [55, 39]. In fig. A.1 we show the $\tilde{\mathbb{P}}_3$ example comparing the Lagrangian nodes of *Basic* and *Cubature* elements. As defined in section 3.5.5, there are several requirements and optimization procedures in order to obtain the *Cubature* elements. These elements are very import in our study because they permit to obtain diagonal mass matrix, and so they decrease considerably the time of computation.
We re-write the requirements of this approach:

1. Obtain a quadrature which is highly accurate, at least $p + p' - 2$ order accurate [34];
2. Obtain positive quadrature weights $\omega_\alpha > 0$ for stability reasons [123];
3. Minimize the number of basis functions of $\tilde{\mathbb{P}}^p$;
4. The set of quadrature points has to be $\tilde{\mathbb{P}}^p$-unisolvent;
5. The number of quadrature points of edges as to be sufficient ensure the conformity of the finite element.

The optimization procedure that lead to these elements consists of several steps where the different goals are optimized one by one. We describe these steps for $p = 1, 2, 3$ in the following sections and give basis functions of these *Cubature* elements, using these notation: $\lambda_i(x, y)$ define the barycentric coordinates which are affine functions on $\mathbb{R}^2$, and verify the following relations

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

where $v_j = (x_j, y_j)$ are the vertices of the triangle and, with an abuse of notation, they can be written in barycentric coordinates as $v_j = (\delta_{1j}, \delta_{2j}, \delta_{3j})$.

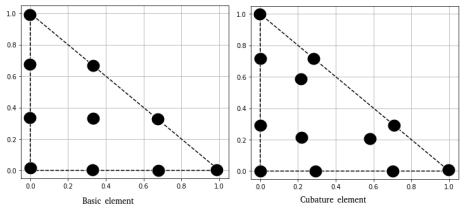


FIGURE A.1: Comparison of two element of degree three: at left the classical one $\mathbb{P}_3$, at right the *Cubature* one $\tilde{\mathbb{P}}_3$.

A.1 *Cubature* elements of degree 1

For element of degree 1, we start with $p' = p = 1$ and DOF are vertices of the triangle. We know that basis functions at the vertices $v_1 = (1,0,0)$, $v_2 = (0,1,0)$ and $v_3 = (0,0,1)$ are given by

$$\phi_{v_i}(\lambda) = \lambda_i, \qquad \text{for } i = 1,2,3.$$

And corresponding weights are $w_{v_i} = \frac{1}{3}$.
The corresponding integration quadrature formula is

$$I_K^{app}(f) = \frac{|K|}{3}(f(v_1) + f(v_2) + f(v_3)). \tag{A.2}$$

All condition are verified.

A.2 *Cubature* elements of degree 2

We start with $p' = p = 2$ and $\tilde{\mathbb{P}}_2$ is initialized as $\mathbb{P}_2$ elements. Unfortunately at mid-edge $e_{ij} = \frac{v_i+v_j}{2}$, $w_{e_{ij}} = \int_K \phi_{e_{ij}} = 0$, which implies that a term in the diagonal mass matrix is equal to zero, and so the mass matrix is not invertible.
We now consider $p' = 3$ and so $\tilde{\mathbb{P}}_2 = \mathbb{P}_2 + b\mathbb{P}_0$, $b \in \mathbb{R}$.

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

- At vertices of the triangle

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

- At edge midpoints

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

- At the centroid

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

The corresponding integration quadrature formula is

$$I_K^{app}(f) = |K| \left(w_v \sum_{j=1}^{3} f(v_j) + w_e \sum_{1 \le i \neq j \le 3} f(e_{ij}) + w_\beta f(G_\beta) \right). \tag{A.3}$$

All condition are verified ($p + p' - 2 = 3$), indeed the polynomial degree of basis function of $\tilde{\mathbb{P}}_2$ is three, and so we can approximate perfectly polynome of degree 3.

A.3 *Cubature* elements of degree 3

We start with $p' = p = 3$ and $\tilde{\mathbb{P}}_3$ is initialized as $\mathbb{P}_3$ elements. However, to integrate exactly polynomial function of degree 4 ($3+3-2$), we have to modify the initial $\tilde{\mathbb{P}}_3$. The only possibilty now (without adding DOF) is to modify the parameter α which evaluate edge mid-points $e_{ij}^{\alpha} = \alpha v_i + (1-\alpha)v_j$. We obtain a first value of $\alpha = \frac{3-\sqrt{3}}{6}$ which gives corresponding basis functions. Unfortunately we obtain $w_v = -1/60 < 0$, $w_\alpha = 1/10$ and $w_\beta = 9/20$.

Following [39, 55], we now consider $p' = 4$ and so $\tilde{\mathbb{P}}_3 = \mathbb{P}_3 + b\mathbb{P}_1$, $b \in \mathbb{R}$. The new space $\tilde{\mathbb{P}}_3$ contains 12 degrees of freedom: 3 vertices v_1, v_2 and v_3, 6 on edges: e_{ij}^{α} for $i, j \in [\![1,\dots,3]\!]$ with $i \neq j$ and three internal points G_i^{β} for $i \in [\![1,\dots,3]\!]$, with $G_i^{\beta} = \beta v_i + \frac{1-\beta}{2}(v_j + v_k)$.

$p' + p - 2 = 5$, so we have to integrate perfectly polynomial functions of degree 5.
1/ We consider the function $f : x \to \lambda_1\lambda_2\lambda_3(\lambda_1 - \frac{1-\beta}{2})(\lambda_1 - \lambda)$. The approximation of the integral is

$$I_K^{app}(f) = \sum_{l=1}^{1} 2w_l f(x_l) = 0. \tag{A.4}$$

But $\int_K f = \frac{1}{2\times 7!}(-2 + 28\beta - 42\beta^2)$. We obtain

$$I_K^{app}(f) = \int_K f \Leftrightarrow 0 = -1 + 14\beta - 21\beta^2 \tag{A.5}$$

$$\Rightarrow \beta_{\pm} = \frac{-14 \pm 4\sqrt{7}}{-42} = \frac{1}{3} \pm \frac{2\sqrt{7}}{21}. \tag{A.6}$$

And so $\beta = 1/3 + 2\sqrt{7}/21$ (β has to be positive).
Then, considering $f : x \to \lambda_1\lambda_2\lambda_3$, we obtain $w_\beta = \frac{21\sqrt{7}}{40(2\sqrt{7}+1)}$.

2/ We consider the function $f : x \to \lambda_1(\lambda_1 - \alpha)(\lambda_1 - (1-\alpha))(1-\lambda_1)$. The approximation of the integral is

$$I_K^{app}(f) = \sum_{l=1}^{1} 2w_l f(x_l) = \sum_{i=1}^{3} w_\beta f(G_i^{\beta}). \tag{A.7}$$

And $\int_K f = -\frac{1}{60} + \frac{\alpha}{12} - \frac{\alpha^2}{12}$. We obtain $\alpha_{\pm} = \frac{-15\sqrt{7}-21 \pm \sqrt{168+174\sqrt{7}}}{2(-15\sqrt{7}-21)}$ which are symmetric regarding $1/2$. We choose $\alpha = \frac{-15\sqrt{7}-21+\sqrt{168+174\sqrt{7}}}{2(-15\sqrt{7}-21)}$ as in [39, 55].
Then, considering $f : x \to \lambda_1(1-\lambda_1)$, we obtain $w_\alpha = \frac{287+115\sqrt{7}}{40(173+49\sqrt{7})}$.

3/ And finally $w_\beta = \frac{21\sqrt{7}}{40(2\sqrt{7}+1)}$.

All condition are verified, we now define basis functions. In order to simply the formulation of the basis functions, let us introduce some polynomials:

$$p_i(\lambda) := \lambda_i \left(\sum_{l=1}^{3} \lambda_l^2 - \frac{1-2\alpha+2\alpha^2}{\alpha(1-\alpha)} \lambda_i(\lambda_j+\lambda_k) + A_i \lambda_j \lambda_k \right), \qquad \text{with } j \neq i \neq k, \tag{A.8}$$

with (obtained solving $\int_K \phi_{v_i} = |K| w_v$)

$$A_i = \left(w_v - \frac{1}{10} - \frac{1}{15}\left(1 - \frac{1-2\alpha+2\alpha^2}{\alpha(1-\alpha)}\right) - \frac{1}{90}\frac{8}{\beta(1-\beta)^2(3\beta-1)}\left(\sum_{l=1}^{3} p_i(G_l)\right)\right) \times \frac{360}{6+\frac{8(1+\beta)}{\beta(1-\beta)(3\beta-1)}}; \tag{A.9}$$

$$p_{ij}(\lambda) := \frac{1}{\alpha(1-\alpha)(2\alpha-1)} \lambda_i \lambda_j (\alpha\lambda_i - (1-\alpha)\lambda_j + (1-2\alpha)\lambda_k), \text{ with } i \neq j \neq k \neq i. \tag{A.10}$$

We can then write the definition of the basis functions:

- At vertices of the triangle, for $i \in [\![1,\dots,3]\!]$

$$\phi_{v_i}(\lambda) = p_i(\lambda) - \frac{8}{\beta(1-\beta)^2(3\beta-1)} \left(\sum_{l=1}^{3} p_i(G_l) \left(\lambda_l - \frac{1-\beta}{2} \right) \right) \prod_{l=1}^{3} \lambda_l;$$

- At the nodes on edges, for $i \neq j \in [\![1,\dots,3]\!]$

$$\phi_{e_{ij}^\alpha}(\lambda) = p_{ij}(\lambda) - \frac{8}{\beta(1-\beta)^2(3\beta-1)} \left(\sum_{l=1}^{3} p_{ij}(G_l) \left(\lambda_l - \frac{1-\beta}{2} \right) \right) \prod_{l=1}^{3} \lambda_l;$$

- At the internal points, for $i \in [\![1,\dots,3]\!]$

$$\phi_{G_i^\beta}(\lambda) = \frac{8}{\beta(1-\beta)^2(3\beta-1)} \left(\lambda_i - \frac{1-\beta}{2} \right) \prod_{l=1}^{3} \lambda_l.$$

Appendix B

Butcher tab

In this appendix we introduce the time integration coefficients used in this work, to make the study fully reproducible. In table B.1 there are the RK coefficients, in table B.3 the SSPRK coefficients and in table B.2 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 B.1: Butcher Tableau of RK 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 B.2: DeC coefficients for equispaced subtimesteps.

SSPRK(3,2) by [117]	
γ	μ
$\begin{matrix} 1 & & \\ 0 & 1 & \\ \frac{1}{3} & 0 & \frac{2}{3} \end{matrix}$	$\begin{matrix} \frac{1}{2} & & \\ 0 & \frac{1}{2} & \\ 0 & 0 & \frac{1}{3} \end{matrix}$
CFL = 2.	

SSPRK(4,3) by [112, Page 189]	
γ	μ
$\begin{matrix} 1 & & & \\ 0 & 1 & & \\ \frac{2}{3} & 0 & \frac{1}{3} & \\ 0 & 0 & 0 & 1 \end{matrix}$	$\begin{matrix} \frac{1}{2} & & & \\ 0 & \frac{1}{2} & & \\ 0 & 0 & \frac{1}{6} & \\ 0 & 0 & 0 & \frac{1}{2} \end{matrix}$
CFL = 2.	

SSPRK(5,4) by [112, 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 B.3: Butcher Tableau of SSPRK methods

Appendix C

Backward Difference Formula

In this appendix we introduce the time integration coefficients of the Backward Difference Formula (BDF) used for the partial derivative evaluation of the entropy dissipation at the order $k = 1, 2, 3, 4$. The BDF is a family of implicit integration methods. Considering the linear system eq. (3.46):

$$d_t U = \mathtt{r}(t, U) \tag{C.1}$$

- BDF1 also called the backward Euler method:

$$U_{n+1} - U_n = \Delta t \mathtt{r}(t_{n+1}, U_{n+1}) \tag{C.2}$$

- BDF2

$$U_{n+2} - \frac{4}{3}U_{n+1} + \frac{1}{3}U_n = \frac{2}{3}\Delta t \mathtt{r}(t_{n+2}, U_{n+2}) \tag{C.3}$$

- BDF3

$$U_{n+3} - \frac{18}{11}U_{n+2} + \frac{9}{11}U_{n+1} - \frac{2}{11}U_n = \frac{6}{11}\Delta t \mathtt{r}(t_{n+3}, U_{n+3}) \tag{C.4}$$

- BDF4

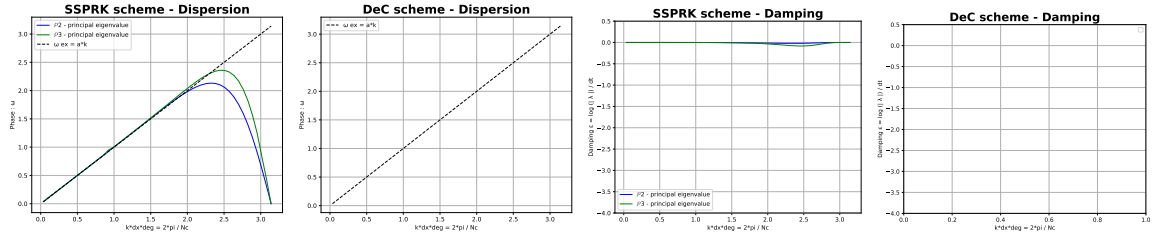
$$U_{n+4} - \frac{48}{25}U_{n+3} + \frac{36}{25}U_{n+2} - \frac{16}{25}U_{n+1} + \frac{3}{25}U_n = \frac{12}{25}\Delta t \mathtt{r}(t_{n+4}, U_{n+4}) \tag{C.5}$$

Appendix D

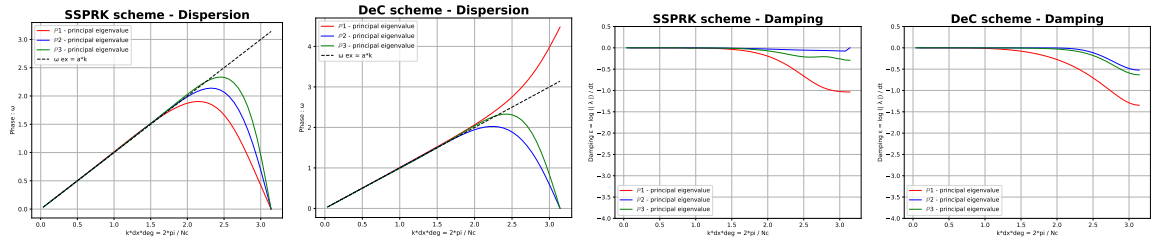
1D dispersion curves

In this appendix we present a summary of the fully discrete Fourier analysis of section 4.3.3, comparing different time schemes (SSPRK and DeC), discretizations (*Basic*, *Cubature*, *Bernstein* elements), and stabilization methods (OSS, 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 4.3. When the scheme was unstable we did not plot the mode. In fig. D.1 one finds the phase and the damping for *Basic* elements, in fig. D.2 for *Cubature* elements and in fig. D.3 for *Bernstein* elements. We remark that for *Cubature* elements in fig. D.2, Δx_3 is scaled differently with respect to the other orders because the point distribution is not equispaced.

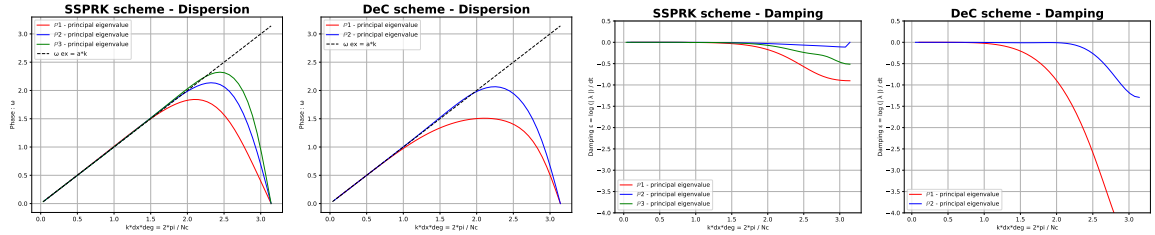
In general, we can observe that the phase error increases passing from full matrix SSPRK methods to diagonal one DeC. This is noticeable even more for *Bernstein* elements. *Cubature* elements, which are not affected by the mass lumping, do not show this behavior, and have a dispersion error which is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



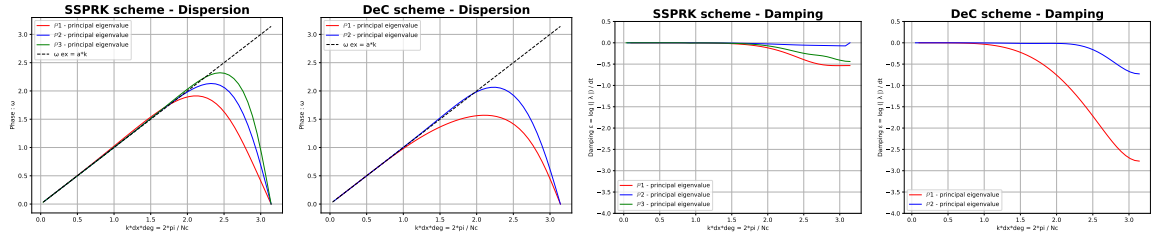
(a) Without any stabilization method



(b) Using the SUPG stabilization method

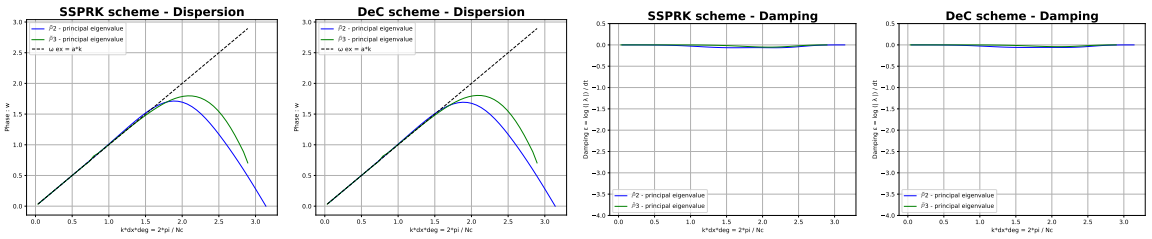


(c) Using the OSS stabilization method

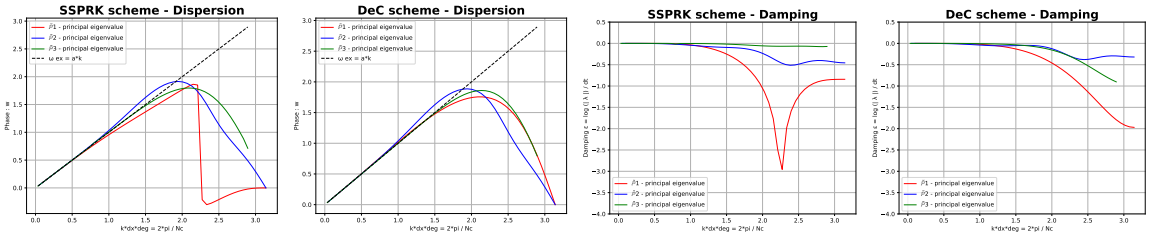


(d) Using the CIP stabilization method

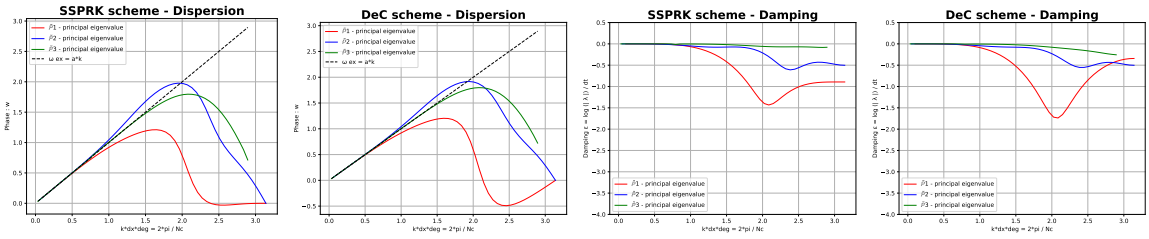
FIGURE D.1: Dispersion and damping coefficients for *Basic* elements, with DeC and SSPRK methods and all stabilization techniques



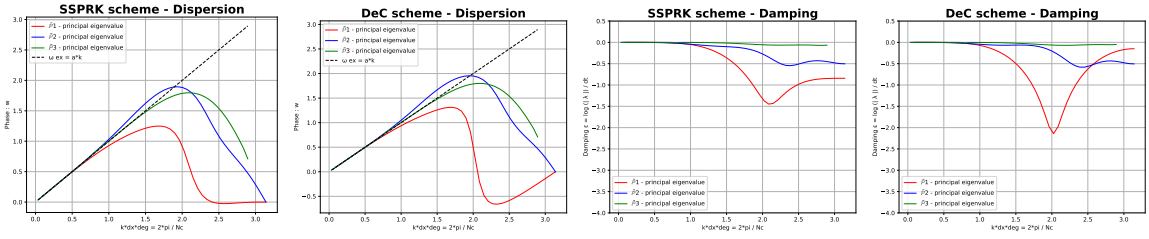
(a) Without any stabilization method



(b) Using the SUPG stabilization method

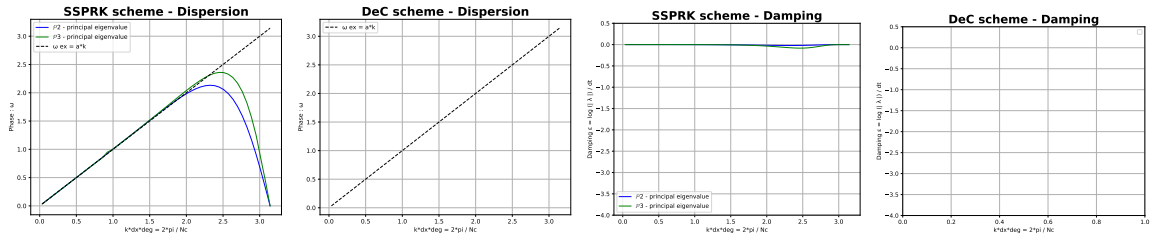


(c) Using the OSS stabilization method

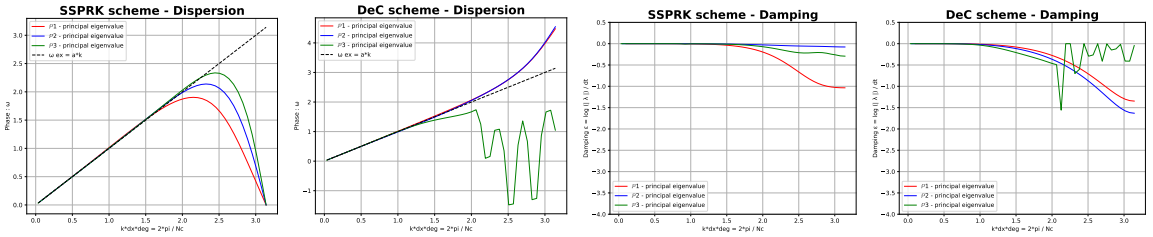


(d) Using the CIP stabilization method

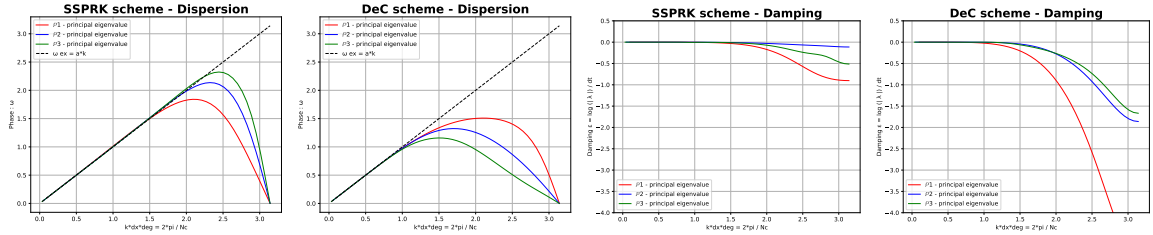
FIGURE D.2: Dispersion and damping coefficients for *Cubature* elements, with DeC and SSPRK methods and all stabilization techniques



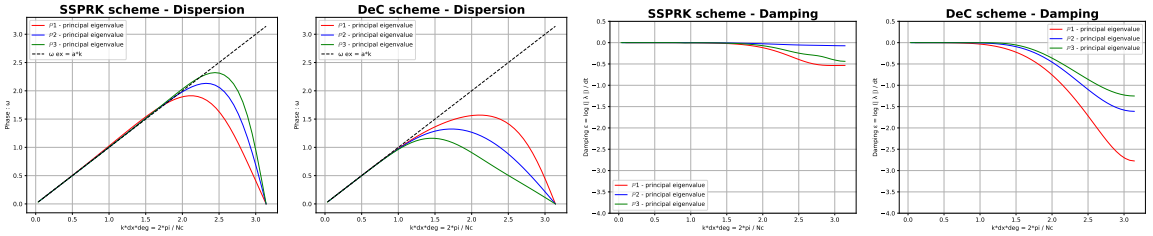
(a) Without any stabilization method



(b) Using the SUPG stabilization method



(c) Using the OSS stabilization method



(d) Using the CIP stabilization method

FIGURE D.3: Dispersion and damping coefficients for *Bernstein* elements, with DeC and SSPRK methods and all stabilization techniques

Appendix E

Fourier analysis, several 2D results

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

E.1 Mesh types and degrees of freedom

We represents in Figure E.1 the mesh configurations used in the Fourier analysis and the degrees of freedom of the elements of degree 3. The red square represents the periodic elementary unit that contains the degrees of freedom of interest for the Fourier analysis.

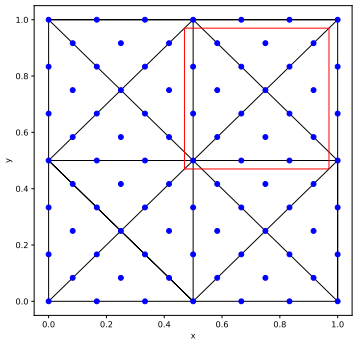
E.2 Fourier analysis results - Optimal Parameters

In this section, we put the optimal values of the stability analysis of Section 5.2.5 after the modification proposed in Section 5.2.6. In Table E.1 we show the parameters for the X mesh and in Table E.2 we show the parameters for the T mesh.

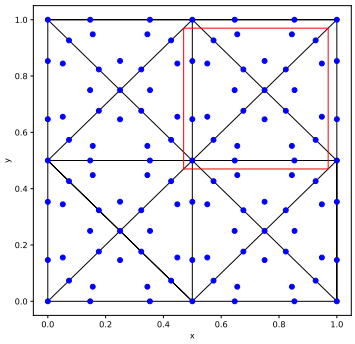
Element &		SUPG		
Time scheme		$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$
Basic	SSPRK	0.739 (0.127)	0.298 (0.058)	0.22 (0.026)
Cub.	SSPRK	1.062 (0.28)	0.1 (0.1)*	0.18 (0.04)*
	DeC	0.616 (0.28)	0.1 (0.04)*	0.144 (0.04)
Bern.	DeC	0.739 (0.298)	0.2 (0.2)*	0.2 (0.153)*

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

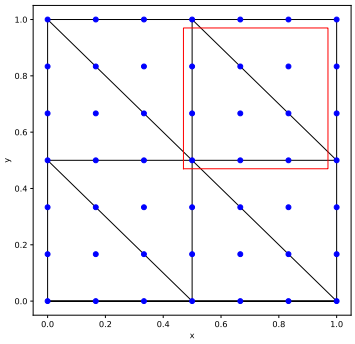
TABLE E.1: X mesh: Optimized CFL and penalty coefficient δ in parenthesis. The symbol "/" means that the fourier analysis for the scheme results always in instability. The values denoted by * are not the optimal one, but they lay in a safer region, see Section 5.2.6.



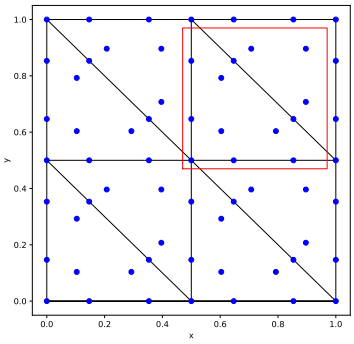
(a) X mesh, *Basic* elements



(b) X mesh, *Cubature* elements



(c) T mesh, *Basic* elements



(d) T mesh, *Cubature* elements

FIGURE E.1: Degrees of freedom and periodic unit for different mesh patterns and elements of degree 3

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

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

TABLE E.2: T mesh: Optimized CFL and penalty coefficient δ in parenthesis. The symbol "/" means that the fourier analysis for the scheme results always in instability.

E.3 Fourier analysis results - stability area

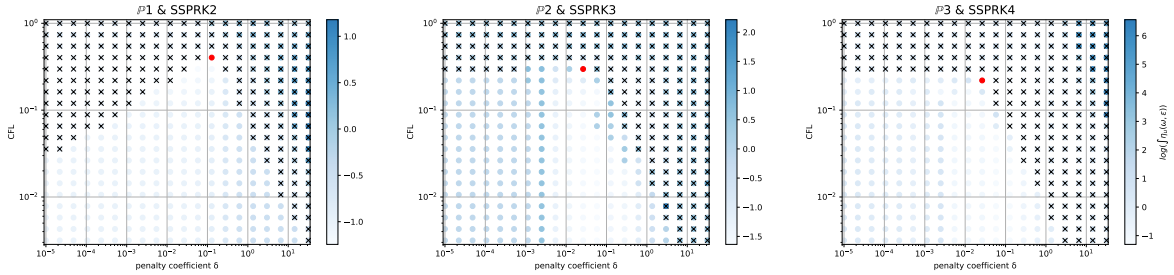
Finally, we present a comparison of stability area between the T and the X mesh. This comparison is performed as before, for all wave angle θ.

The comparison using *Basic* element, SSPRK time integration method and the OSS, the CIP and the SUPG stabilization techniques are represented in repectively fig. E.2, fig. E.3 and fig. E.4.

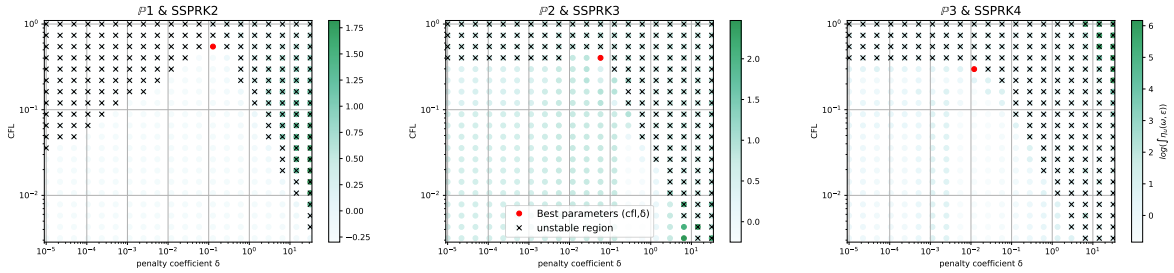
The comparison using *Cubature* element, SSPRK time integration method and the OSS, the CIP and the SUPG stabilization techniques are represented in repectively fig. E.5, fig. E.6 and fig. E.7.

The comparison using *Cubature* element, DeC time integration method and the OSS, the CIP and the SUPG stabilization techniques are represented in repectively fig. E.8, fig. E.9 and fig. E.10.

The comparison using *Bernstein* element, DeC time integration method and the OSS, the CIP and the SUPG stabilization techniques are represented in respectively fig. E.11 , fig. E.12 and fig. E.13.

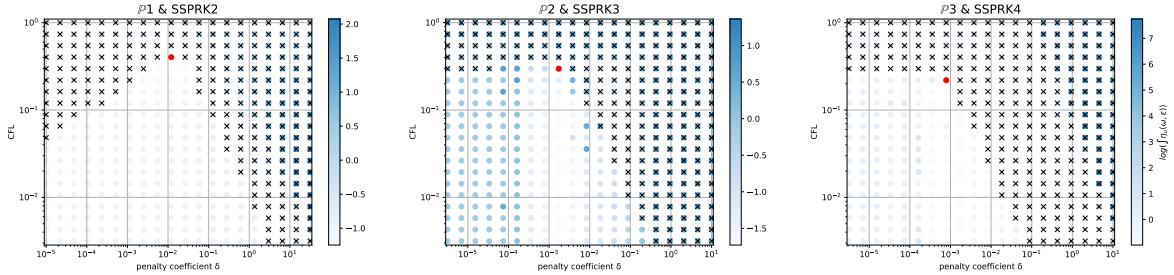


(a) X mesh

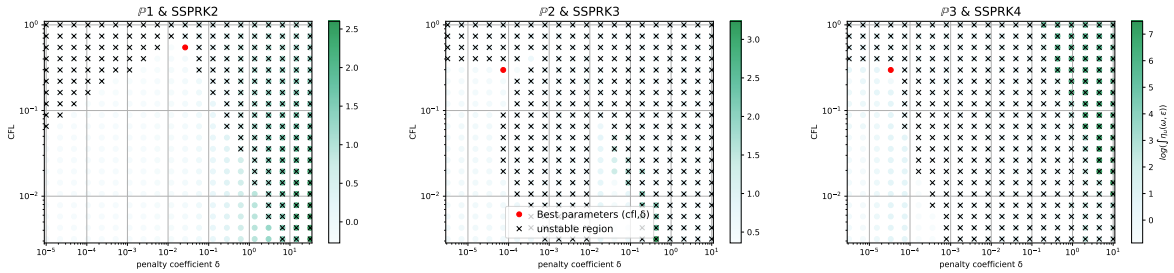


(b) T mesh

FIGURE E.2: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ *Basic* elements with SSPRK scheme and OSS stabilization.

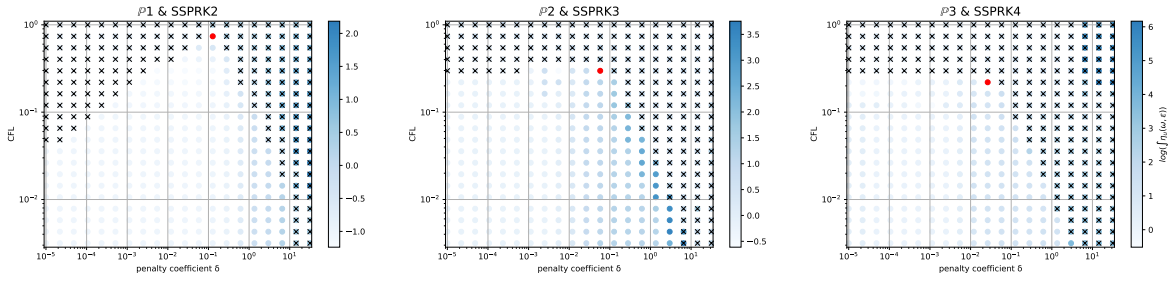


(a) X mesh

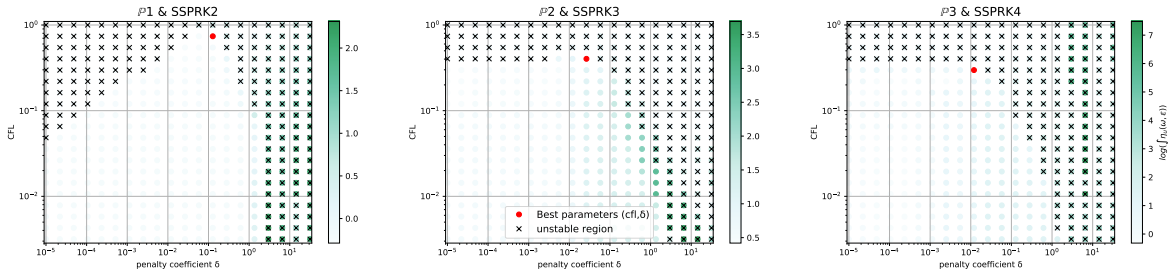


(b) T mesh

FIGURE E.3: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ *Basic* elements with SSPRK scheme and CIP stabilization.

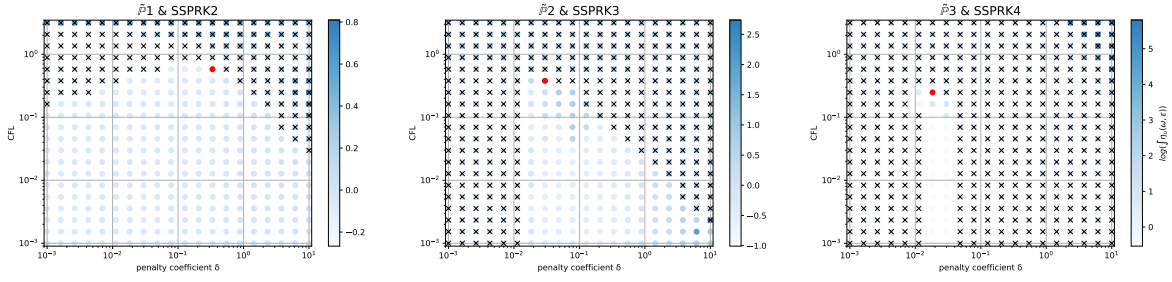


(a) X mesh

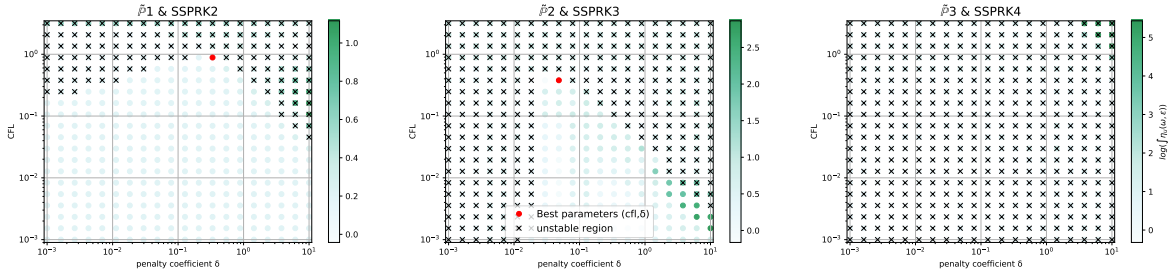


(b) T mesh

FIGURE E.4: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ *Basic* elements with SSPRK scheme and SUPG stabilization.

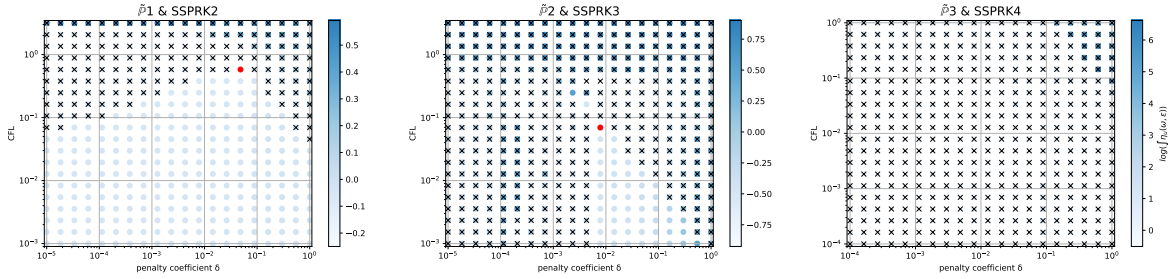


(a) X mesh

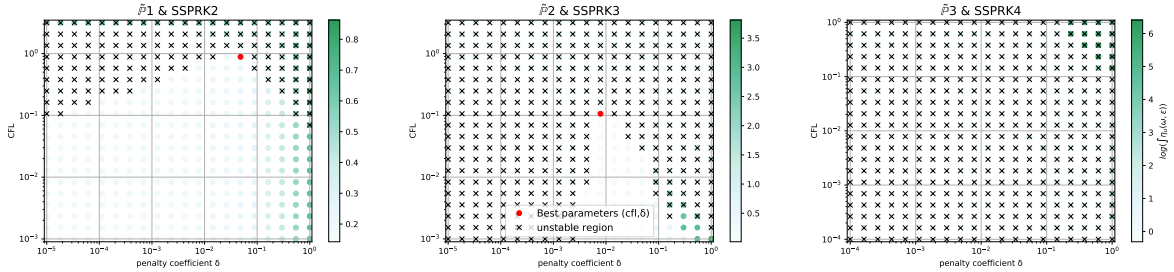


(b) T mesh

FIGURE E.5: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ *Cubature* elements with SSPRK scheme and OSS stabilization.

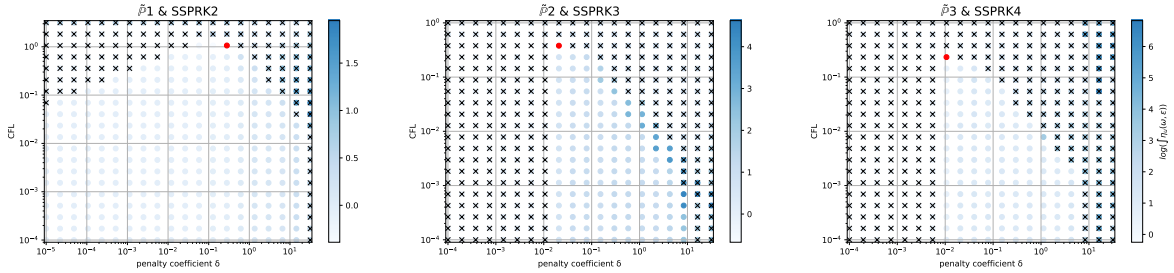


(a) X mesh

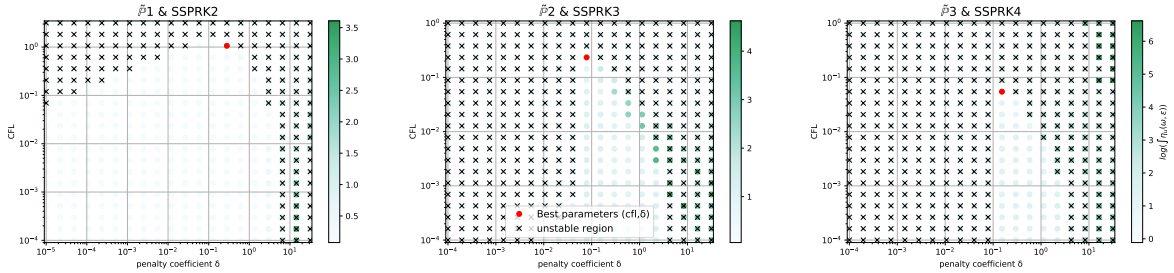


(b) T mesh

FIGURE E.6: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ *Cubature* elements with SSPRK scheme and CIP stabilization.

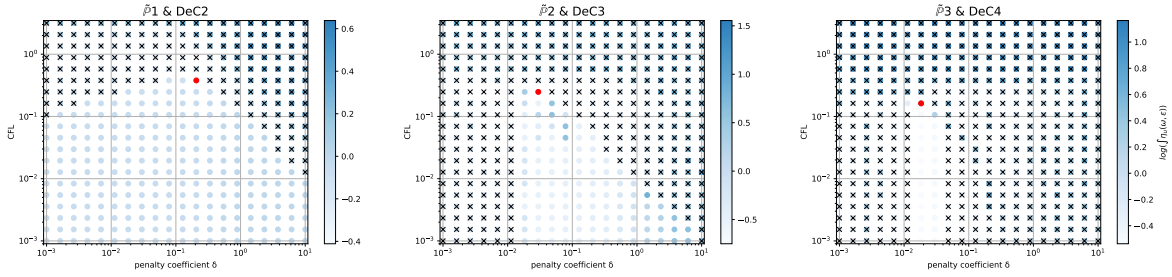


(a) X mesh

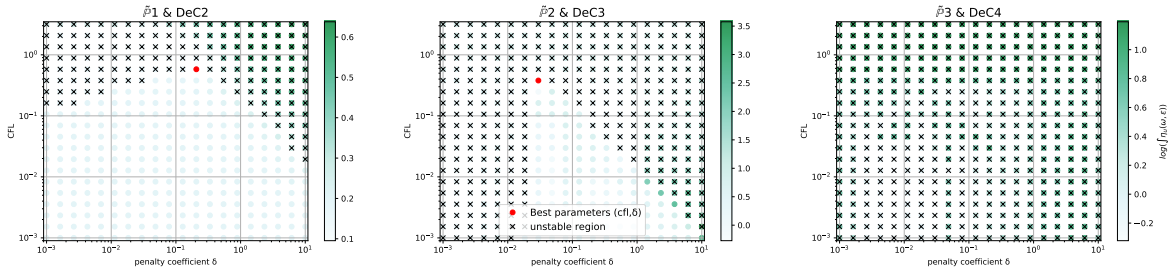


(b) T mesh

FIGURE E.7: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ *Cubature* elements with SSPRK scheme and SUPG stabilization.

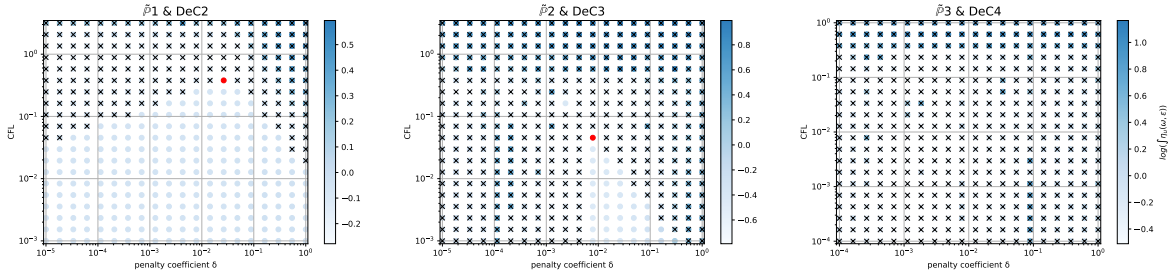


(a) X mesh

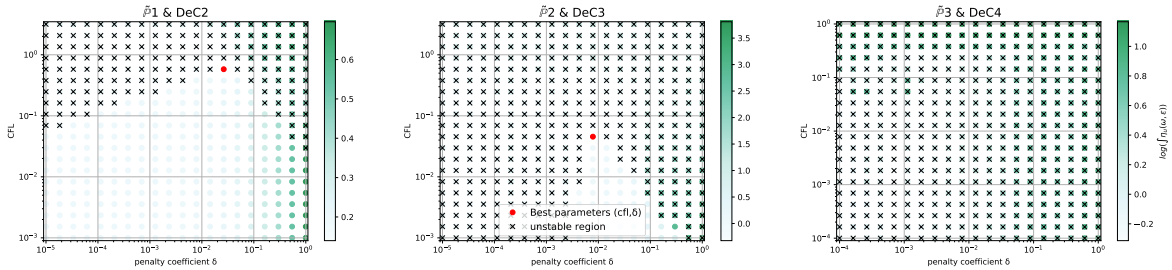


(b) T mesh

FIGURE E.8: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ *Cubature* elements with DeC scheme and OSS stabilization.

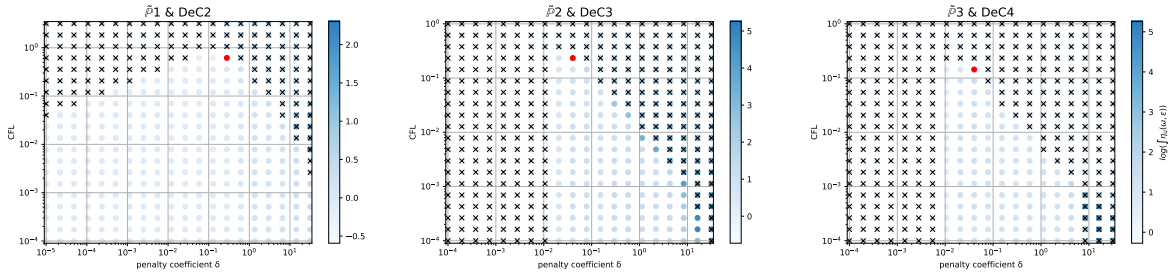


(a) X mesh

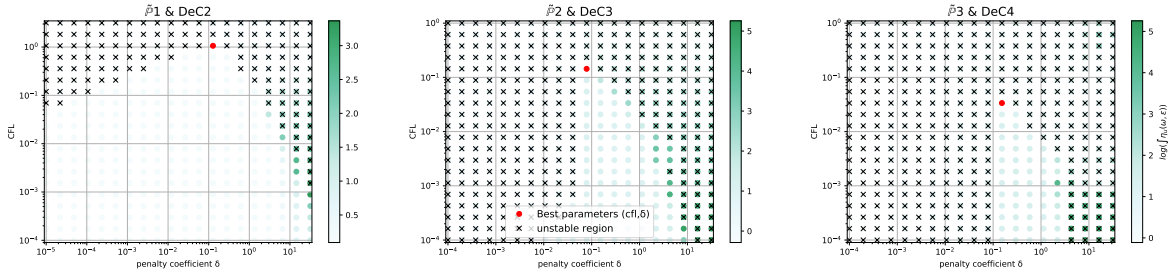


(b) T mesh

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

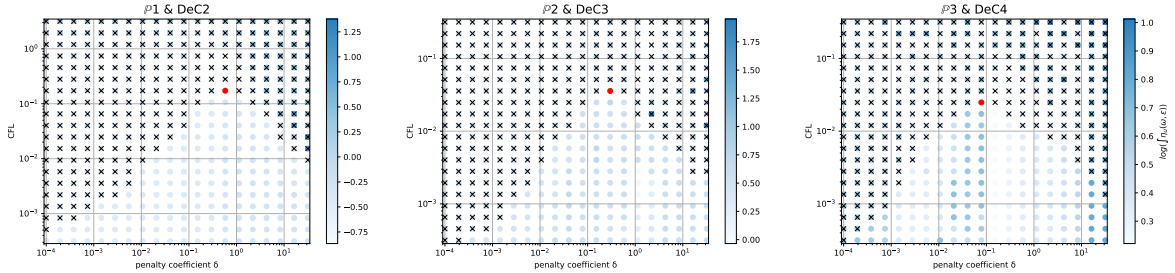


(a) X mesh

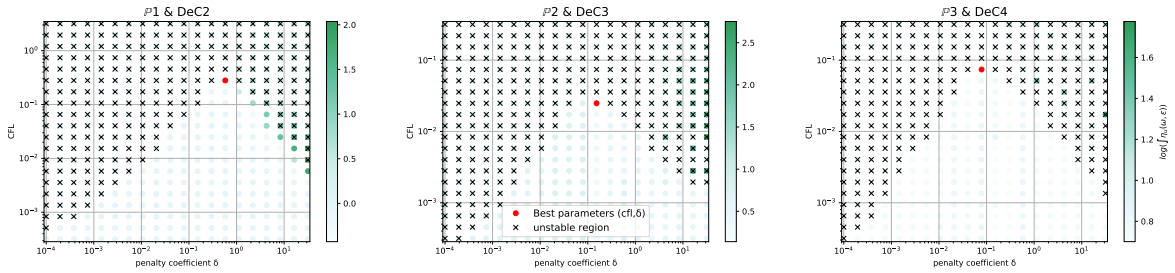


(b) T mesh

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

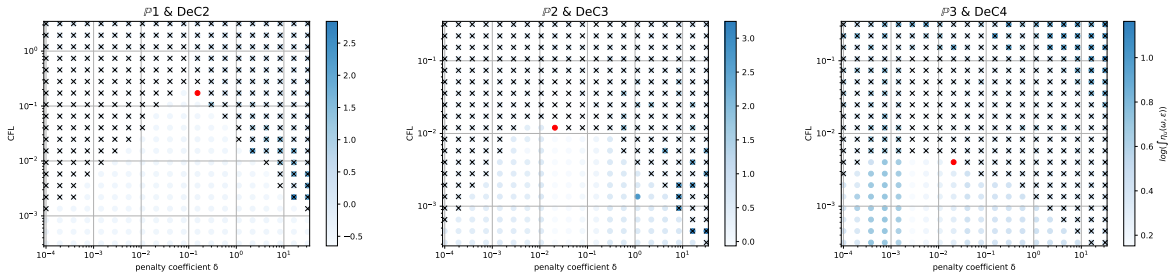


(a) X mesh

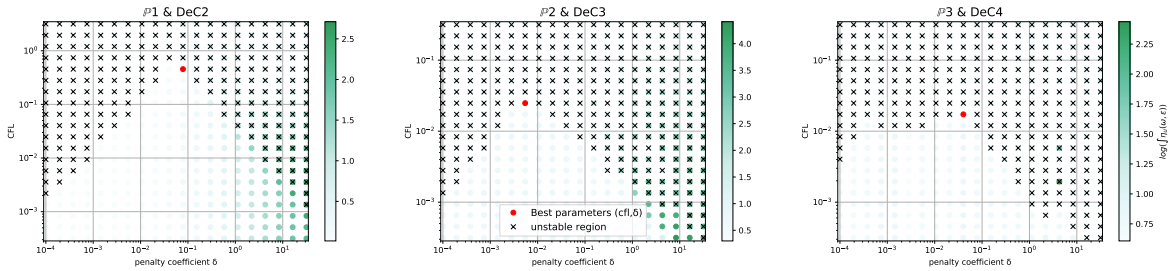


(b) T mesh

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

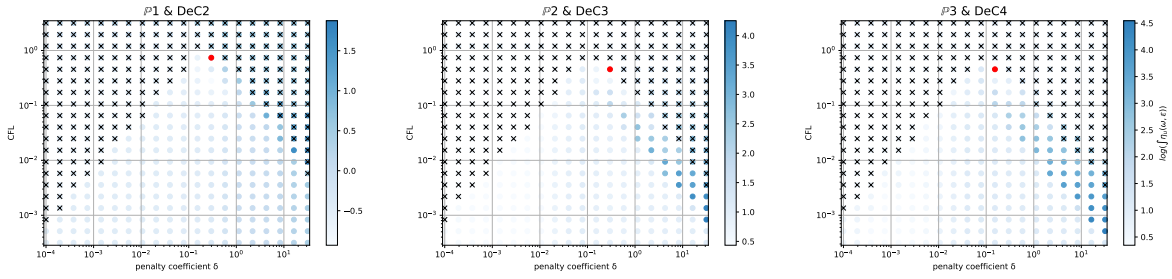


(a) X mesh

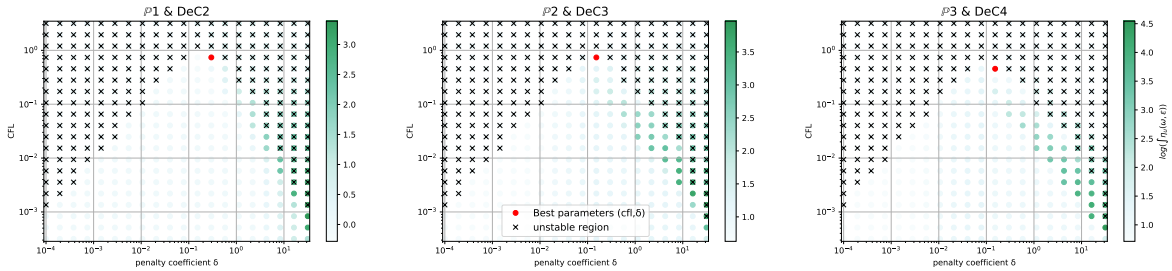


(b) T mesh

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



(a) X mesh



(b) T mesh

FIGURE E.13: $\log(\eta_u)$ values (blue scale) and stable area (unstable with black crosses), on (CFL, δ) plane. The red dot denotes the optimal value. From left to right $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ *Bernstein* elements with DeC scheme and SUPG stabilization.

Appendix F

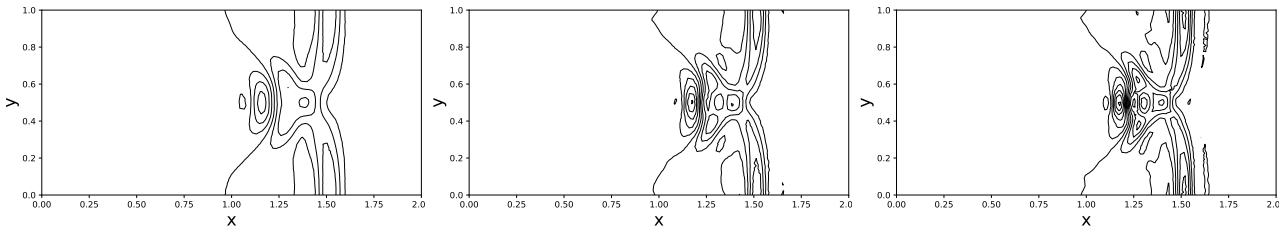
Shallow Water tests, additional results

F.1 Perturbation of the lake at rest

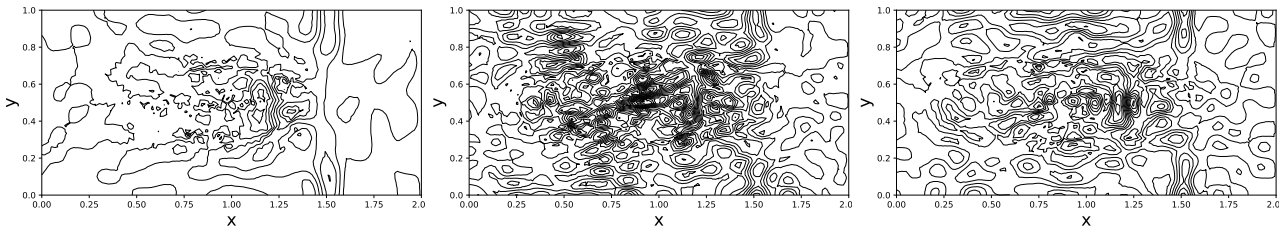
This section is complementary results of section 6.3.2 with a smooth and non-smooth bathymetry.

F.1.1 With a smooth topography

We remember the *Laket at rest* solution with the non-smooth topography in fig. 6.1 (in section 6.3.1). We use the perturbation defined by eq. (6.15) from section 6.3.2. Regarding the 2d view from the top with line contour of water elevation, we compare the approximated solution using and not using the well-balanced formulation. The Comparison using *Basic* elements and the OSS is represented in fig. F.1, *Basic* elements and the CIP in fig. F.2, and *Cubature* elements and the CIP in fig. F.3.

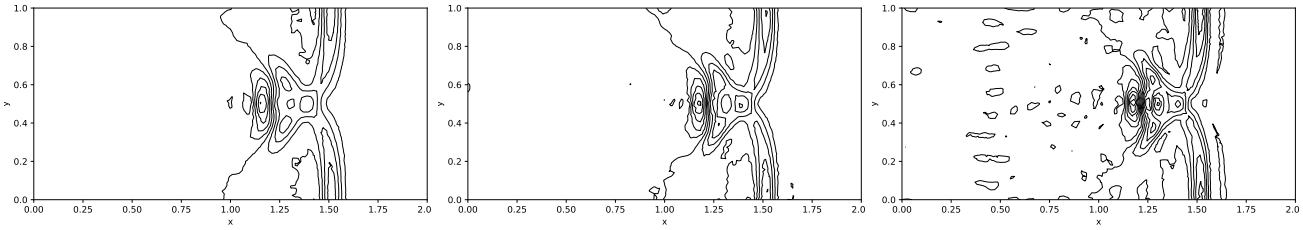


(a) *Basic* $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ elements (from left to right) using the well-balanced formulation.

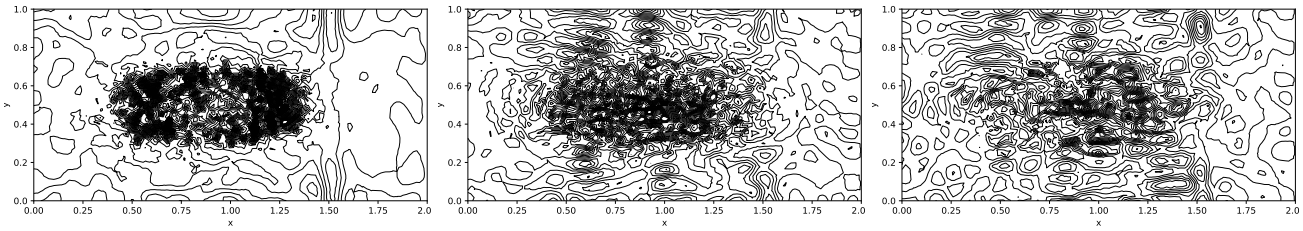


(b) *Basic* $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ elements (from left to right) not using the well-balanced formulation.

FIGURE F.1: 2D view of the free surface elevation at $t = 0.48s$ with the smooth topography eq. (6.13). All simulation are performed using *Basic* elements, the OSS stabilization technique, and SSPRK schemes.

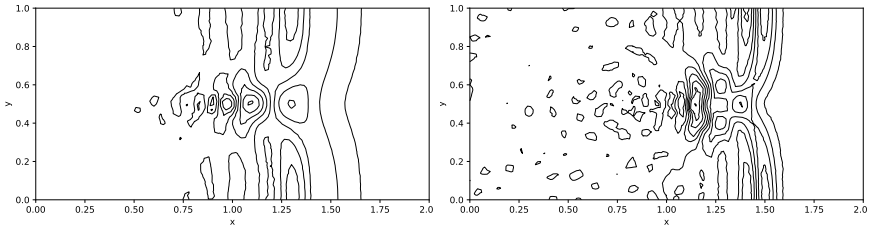


(a) *Basic* $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ elements (from left to right) using the well-balanced formulation.

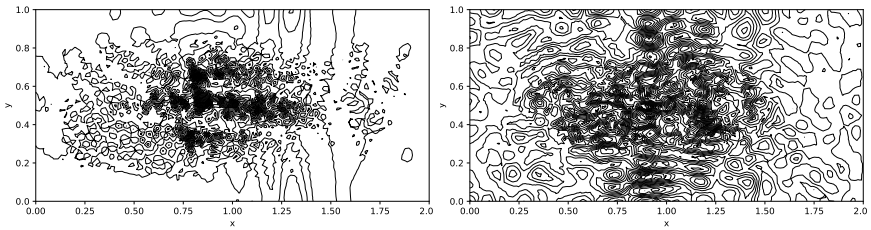


(b) *Basic* $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ elements (from left to right) not using the well-balanced formulation.

FIGURE F.2: 2D view of the free surface elevation at $t = 0.48s$ with the smooth topography eq. (6.13). All simulation are performed using *Basic* elements, the CIP stabilization technique, and SSPRK schemes.



(a) *Cubature* $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$ elements (from left to right) using the well-balanced formulation.



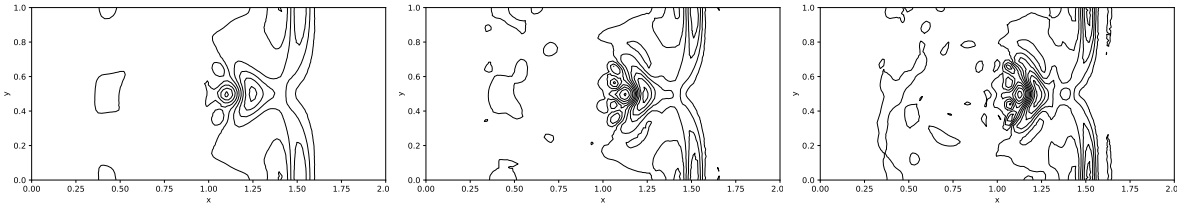
(b) *Cubature* $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$ elements (from left to right) not using the well-balanced formulation.

FIGURE F.3: 2D view of the free surface elevation at $t = 0.48s$ with the smooth topography eq. (6.13). All simulation are performed using *Cubature* elements, the CIP stabilization technique, and SSPRK schemes.

F.1.2 With a non-smooth topography

We remember the *Laket at rest* solution with the non-smooth topography in fig. 6.2 (in section 6.3.1). We use the perturbation defined by eq. (6.15) from section 6.3.2 and we show

Basic elements. At rigth $\mathbb{P}_1$*, on the middle* $\mathbb{P}_2$*, at left* $\mathbb{P}_3$ *elements.*



Cubature elements. At rigth $\tilde{\mathbb{P}}_1$*, on the middle* $\tilde{\mathbb{P}}_2$*, at left* $\tilde{\mathbb{P}}_3$ *elements.*

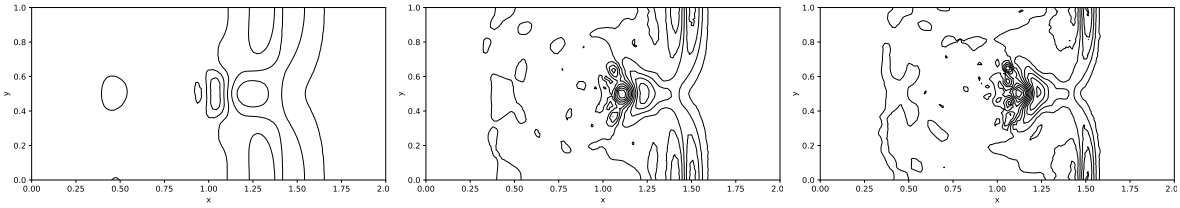
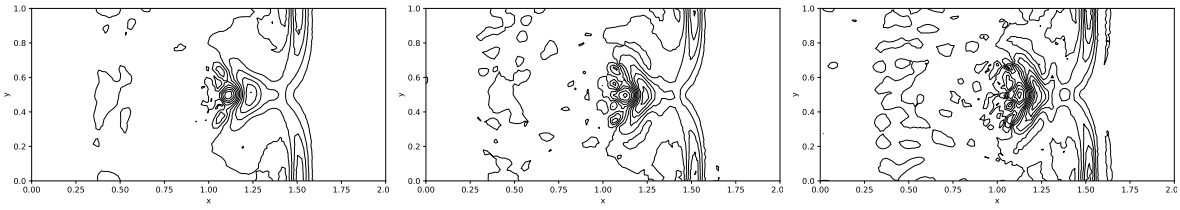


FIGURE F.4: 2D view of the free surface elevation at $t = 0.48s$ with the non-smooth topography eq. (6.14). Both discretization use the OSS stabilization technique and SSPRK schemes.

numerical results in fig. F.4 using the OSS stabilization technique and fig. F.5 using the CIP stabilization technique.

Basic elements. At rigth $\mathbb{P}_1$*, on the middle* $\mathbb{P}_2$*, at left* $\mathbb{P}_3$ *elements.*



Cubature elements .At rigth $\tilde{\mathbb{P}}_1$*, at left* $\tilde{\mathbb{P}}_2$*.*

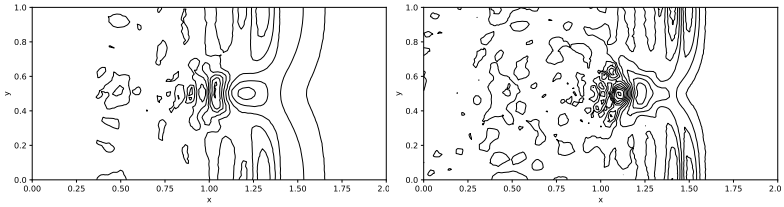


FIGURE F.5: 2D view of the free surface elevation at $t = 0.48s$ with the non-smooth topography eq. (6.14). Both discretization use the CIP stabilization technique and SSPRK schemes.

F.2 Global atmospheric tests - Unstable jet

This section is complementary results of section 7.2.2. We present the vorticity field after 96h (=4 days) from J. Galewsky et al. [51] in fig. F.6. A representation of contour levels for

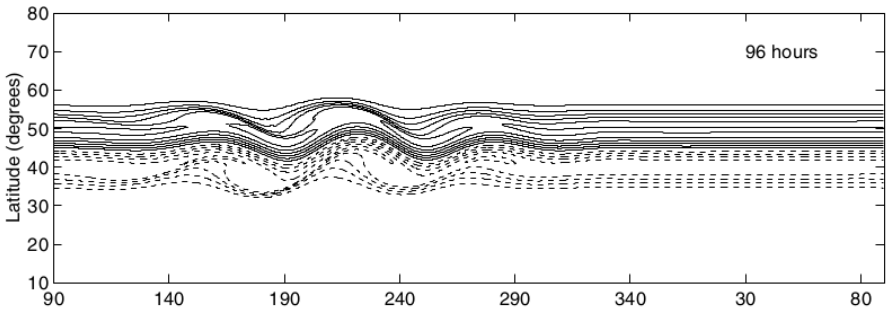


FIGURE F.6: Approximated solution of the vorticity field after 4 days. Sources from J. Galewsky et al. [51].

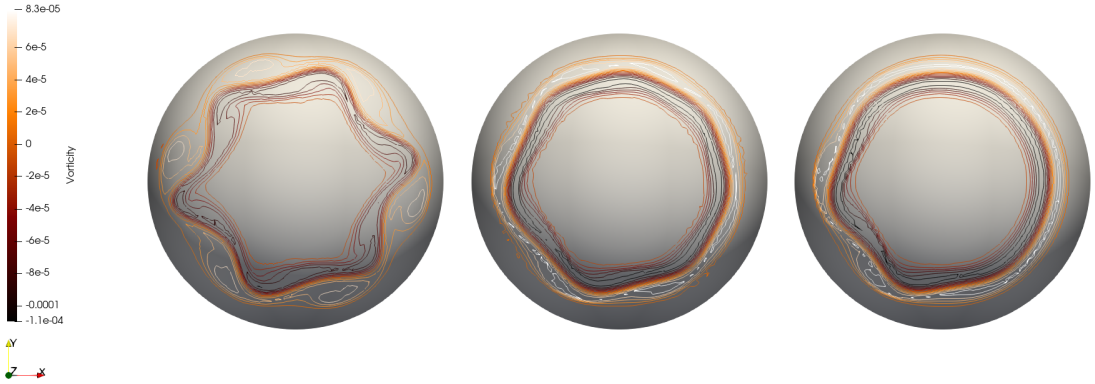
our simulation is showed in fig. F.7.

We summarize the cpu-time for all methods in table F.1. The time of the simulation is 4 days.

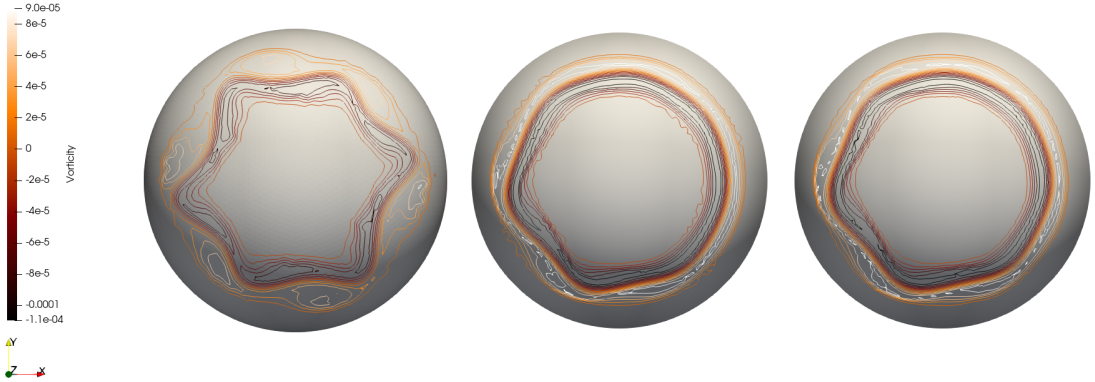
Element & Time scheme		Total cpu-time (s)	Residual computation (s)	Limit solution (s)	Mass matrix inversion (s)
Basic SSPRK	$\mathbb{P}_1$	8673	5882 (67.8%)	1615 (18.6%)	34 (0.39%)
	$\mathbb{P}_2$	48720	28718 (58.9%)	11737 (24.1%)	199 (0.41%)
	$\mathbb{P}_3$	157539	96944 (61.2%)	50406 (32.0%)	1138 (0.72%)
Cub. SSPRK	$\tilde{\mathbb{P}}_1$	7400	4642 (62.7%)	1606 (21.7%)	17 (0.23%)
	$\tilde{\mathbb{P}}_2$	48458	26657 (55.0%)	13127 (27.1%)	89 (0.18%)
	$\tilde{\mathbb{P}}_3$	154996	83666 (54.0%)	60141 (38.8%)	377 (0.24%)

TABLE F.1: Unstable Jet. Summary tab of cpu-time repartition for the mesh $h_k = 223km$, in parenthesis the pourcentage. $t_f = 4$ days.

In table F.1, we observe the same behaviour in the cpu-time repartition than in table 7.2 and table 7.3. However, *Cubature* elements with mass-lumping allow to decrease the cpu-time for elements of degree 1, 2 and 3. And the approximated solution is equivalent between the *Basic* and the *Cubature* discretization.



(a) *Basic* $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$ elements (from left to right).



(b) *Cubature* $\tilde{\mathbb{P}}_1$, $\tilde{\mathbb{P}}_2$, $\tilde{\mathbb{P}}_3$ elements (from left to right).

FIGURE F.7: Unstable Jet - 2D view from the northen pole of the vorticity field contour levels (from $-1.1e-4$ to $1.1e-4$) at $t_f = 4$ days. $h_K = 223km$

Bibliography

[1] R. Abgrall. “A general framework to construct schemes satisfying additional conservation relations. Application to entropy conservative and entropy dissipative schemes”. In: *Journal of Computational Physics* 372 (2018), pp. 640 –666. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2018.06.031. URL: http://www.sciencedirect.com/science/article/pii/S0021999118304091.

[2] R. Abgrall. “High Order Schemes for Hyperbolic Problems Using Globally Continos Approximation and Avoiding Mass Matrices”. In: *Journal of Scientific Computing* 73 (July 2017). DOI: 10.1007/s10915-017-0498-4.

[3] R. Abgrall and M. Ricchiuto. “High order methods for CFD”. In: *Encyclopedia of Computational Mechanics, Second Edition*. Ed. by Rene de Borst Erwin Stein and Thomas J.R. Hughes. John Wiley and Sons, 2017.

[4] Remi Abgrall and Mario Ricchiuto. *Hyperbolic balance laws: residual distribution, local and global fluxes*. 2021. arXiv: 2109.08491 [math.NA].

[5] Luca Arpaia et al. “An efficient 3d/2d-covariant formulation of the spherical shallow water equations: well balanced DG approximation and application to tsunami and storm surge”. working paper or preprint. Apr. 2021. URL: https://hal-brgm.archives-ouvertes.fr/hal-03207171.

[6] Emmanuel Audusse and Marie-Odile Bristeau. “A well-balanced positivity preserving “second-order” scheme for shallow water flows on unstructured meshes”. In: *Journal of Computational Physics* 206.1 (2005), pp. 311–333. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2004.12.016. URL: https://www.sciencedirect.com/science/article/pii/S0021999104005157.

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

[8] Santiago Badia and Ramon Codina. “Unified Stabilized Finite Element Formulations for the Stokes and the Darcy Problems”. In: *SIAM Journal on Numerical Analysis* 47 (Jan. 2009). DOI: 10.1137/08072632X.

[9] TIMOTHY BARTH and DENNIS JESPERSEN. “The design and application of upwind schemes on unstructured meshes”. In: *27th Aerospace Sciences Meeting*. DOI: 10.2514/6.1989-366. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.1989-366. URL: https://arc.aiaa.org/doi/abs/10.2514/6.1989-366.

[10] T.J. Barth. “Numerical Methods for Gasdynamic Systems on Unstructured Meshes”. In: *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws*. Ed. by Kröner, Ohlberger, and Rohde. Vol. 5. Lecture Notes in Computational Science and Engineering. Heidelberg: Springer-Verlag, 1998, pp. 195–285.

[11] Roland Becker, Erik Burman, and Peter Hansbo. "A finite element time relaxation method". In: *Comptes Rendus Mathematique* 349.5 (2011), pp. 353–356. ISSN: 1631-073X. DOI: https://doi.org/10.1016/j.crma.2010.12.010. URL: https://www.sciencedirect.com/science/article/pii/S1631073X10003894.

[12] Alfredo Bermudez and Ma Elena Vazquez. "Upwind methods for hyperbolic conservation laws with source terms". In: *Computers & Fluids* 23.8 (1994), pp. 1049–1071. ISSN: 0045-7930. DOI: https://doi.org/10.1016/0045-7930(94)90004-3. URL: https://www.sciencedirect.com/science/article/pii/0045793094900043.

[13] Paul-Emile Bernard et al. "High-order discontinuous Galerkin schemes on general 2D manifolds applied to the shallow water equations". In: *J. Comput. Phys.* 228 (2009), pp. 6514–6535.

[14] L. Bos, Mark Taylor, and Beth Wingate. "Tensor product Gauss-Lobatto points are Fekete points for the cube". In: *Math. Comput.* 70 (Oct. 2001), pp. 1543–1547. DOI: 10.1090/S0025-5718-00-01262-X.

[15] P. Brufau and Pilar Garcia-Navarro. "Unsteady free surface flow simulation over complex topography with a multidimensional upwind technique". In: *Journal of Computational Physics* 186 (Apr. 2003), pp. 503–526. DOI: 10.1016/S0021-9991(03)00072-X.

[16] P. Brufau, P. García-Navarro, and M. E. Vázquez-Cendón. "Zero mass error using unsteady wetting–drying conditions in shallow flows over dry irregular topography". In: *International Journal for Numerical Methods in Fluids* 45.10 (2004), pp. 1047–1082. DOI: https://doi.org/10.1002/fld.729. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.729. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.729.

[17] P. Brufau, M. E. Vázquez-Cendón, and P. García-Navarro. "A numerical model for the flooding and drying of irregular domains". In: *International Journal for Numerical Methods in Fluids* 39.3 (2002), pp. 247–275. DOI: https://doi.org/10.1002/fld.285. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.285. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.285.

[18] Sébastien de Brye. *Uhaina report*. Rapport technique - Uhaina. Aug. 2018.

[19] E. Burman, A. Ern, and M.A. Fernandez. "Explicit Runge-Kutta schemes and finite elements with symmetric stabilization for first-order linear PDE systems". In: *SIAM J. Numer. Anal.* 48.6 (2010), pp. 2019–2042.

[20] E. Burman, A. Quarteroni, and B. Stamm. "Interior Penalty Continuous and Discontinuous Finite Element Approximations of Hyperbolic Equations". In: *J.Sci.Comp.* 43.3 (2010), pp. 293–312.

[21] Erik Burman. "Consistent SUPG-method for transient transport problems: Stability and convergence". In: *Computer Methods in Applied Mechanics and Engineering* 199 (Mar. 2010), pp. 1114–1123. DOI: 10.1016/j.cma.2009.11.023.

[22] Erik Burman. "Weighted error estimates for transient transport problems discretized using continuous finite elements with interior penalty stabilization on the gradient jumps". In: *arXiv preprint arXiv:2104.06880* (2021).

[23] Erik Burman, Alexandre Ern, and Miguel Fernández. "Explicit Runge-Kutta Schemes and Finite Elements with Symmetric Stabilization for First-Order Linear PDE Systems". In: *SIAM Journal on Numerical Analysis* 48 (Jan. 2010). DOI: 10.1137/090757940.

[24] Erik Burman and Peter Hansbo. "Edge stabilization for Galerkin approximations of convection–diffusion problems". In: *Computer Methods in Applied Mechanics and Engineering* 193 (Apr. 2004), pp. 1437–1453. DOI: 10.1016/j.cma.2003.12.032.

[25] Erik Burman and Peter Hansbo. "The edge stabilization method for finite elements in CFD". In: *Numerical mathematics and advanced applications*. Springer, 2004, pp. 196–203.

[26] Erik Burman, Peter Hansbo, and Mats G. Larson. "A cut finite element method for a model of pressure in fractured media". In: *Numerische Mathematik* 146.4 (2020), pp. 783–818. ISSN: 0945-3245. DOI: https://doi.org/10.1007/s00211-020-01157-5.

[27] Erik Burman, Alfio Quarteroni, and Benjamin Stamm. "Interior Penalty Continuous and Discontinuous Finite Element Approximations of Hyperbolic Equations". In: *Journal of Scientific Computing* 43 (June 2010), pp. 293–312. DOI: 10.1007/s10915-008-9232-6.

[28] Erik Burman, Alfio Quarteroni, and Benjamin Stamm. "Stabilization Strategies for High Order Methods for Transport Dominated Problems". In: *Bolletino dell'Unione Matematica Italiana* 1 (Feb. 2008).

[29] J. C. Butcher. "Numerical Differential Equation Methods". In: John Wiley & Sons, Ltd, 2016. Chap. 2, pp. 55–142. ISBN: 9781119121534. DOI: https://doi.org/10.1002/9781119121534.ch2. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119121534.ch2. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119121534.ch2.

[30] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Ltd, 2008. ISBN: 9780470753767. DOI: 10.1002/9780470753767.fmatter. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470753767.fmatter. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470753767.fmatter.

[31] Valerio Caleffi, Alessandro Valiani, and Andrea Zanni. "Finite volume method for simulating extreme flood events in natural channels". In: *Journal of Hydraulic Research* 41.2 (2003), pp. 167–177. DOI: 10.1080/00221680309499959. eprint: https://doi.org/10.1080/00221680309499959. URL: https://doi.org/10.1080/00221680309499959.

[32] MANUEL J. CASTRO, JOSÉ M. GONZÁLEZ-VIDA, and CARLOS PARÉS. "NUMERICAL TREATMENT OF WET/DRY FRONTS IN SHALLOW FLOWS WITH A MODIFIED ROE SCHEME". In: *Mathematical Models and Methods in Applied Sciences* 16.06 (2006), pp. 897–931. DOI: 10.1142/S021820250600139X. eprint: https://doi.org/10.1142/S021820250600139X. URL: https://doi.org/10.1142/S021820250600139X.

[33] M.J. Castro et al. "The numerical treatment of wet/dry fronts in shallow flows: application to one-layer and two-layer systems". In: *Mathematical and Computer Modelling* 42.3 (2005), pp. 419–439. ISSN: 0895-7177. DOI: https://doi.org/10.1016/j.mcm.2004.01.016. URL: https://www.sciencedirect.com/science/article/pii/S0895717705002852.

[34] Philippe G . Ciarlet. *The Finite Element Method for Elliptic Problems*. Studies in mathematics and its applications 4. North-Holland, 1978. ISBN: 9780080875255,9780444850287,9780

[35] Bernardo Cockburn, Suchung Hou, and Chi-Wang Shu. "The Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws. IV: The Multidimensional Case". In: *Mathematics of Computation* 54.190 (1990), pp. 545–581. ISSN: 00255718, 10886842. URL: http://www.jstor.org/stable/2008501.

[36] Bernardo Cockburn, George E. Karniadakis, and Chi-Wang Shu. "The Development of Discontinuous Galerkin Methods". In: *Discontinuous Galerkin Methods*. Ed. by Bernardo Cockburn, George E. Karniadakis, and Chi-Wang Shu. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 3–50. ISBN: 978-3-642-59721-3.

[37] Ramon Codina. "Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods". In: *Computer Methods in Applied Mechanics and Engineering* 190.13 (2000), pp. 1579–1599. ISSN: 0045-7825. DOI: https://doi.org/10.1016/S0045-7825(00)00254-1. URL: https://www.sciencedirect.com/science/article/pii/S0045782500002541.

[38] Ramon Codina and Jordi Blasco. "A finite element formulation for the Stokes problem allowing equal velocity-pressure interpolation". In: *Computer Methods in Applied Mechanics and Engineering* 143.3 (1997), pp. 373–391. ISSN: 0045-7825. DOI: https://doi.org/10.1016/S0045-7825(96)01154-1. URL: https://www.sciencedirect.com/science/article/pii/S0045782596011541.

[39] Gary Cohen et al. "Higher Order Triangular Finite Elements with Mass Lumping for the Wave Equation". In: *Siam Journal on Numerical Analysis* 38 (Jan. 2001). DOI: 10.1137/S0036142997329554.

[40] Jeffrey Connors and William Layton. "On the accuracy of the finite element method plus time relaxation". In: *Math. Comput.* 79 (Apr. 2010), pp. 619–648. DOI: 10.1090/S0025-5718-09-02316-3.

[41] J. Coté. "A Lagrange multiplier approach for the metric terms of semi-Lagrangian models on the sphere". In: *Quarterly Journal of the Royal Meteorological Society* 114.483 (1988), pp. 1347–1352. DOI: https://doi.org/10.1002/qj.49711448310. eprint: https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1002/qj.49711448310. URL: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.49711448310.

[42] R. Courant, K. O. Friedrichs, and H. Lewy. "Über die partiellen Differenzengleichungen der mathematischen Physik". In: *Math. Ann.* 100 (1928), pp. 32–74.

[43] H. Deconinck and Mario Ricchiuto. "Residual Distribution Schemes: Foundations and Analysis". In: Oct. 2007. ISBN: 9780470091357. DOI: 10.1002/0470091355.ecm054.

[44] Argiris Delis, Maria Kazolea, and Nikolaos Kampanis. "A robust high resolution finite volume scheme for the simulation of long waves over complex domain". In: *International Journal for Numerical Methods in Fluids* 56 (Feb. 2008), pp. 419 – 452. DOI: 10.1002/fld.1537.

[45] *Direction Générale de Prevention des Risques. Guide méthodologique: Plan de prévention des risques littoraux*. Rapport DGPR. 2014.

[46] Jim Douglas and Todd Dupont. "Interior Penalty Procedures for Elliptic and Parabolic Galerkin Method". In: vol. 58. Springer, Aug. 2008, pp. 207–216. DOI: 10.1007/BFb0120591.

[47] Alok Dutt, Leslie Greengard, and Vladimir Rokhlin. "Spectral deferred correction methods for ordinary differential equations". English (US). In: *BIT Numerical Mathematics* 40.2 (June 2000), pp. 241–266. ISSN: 0006-3835.

[48] Alexandre Ern and Jean-Luc Guermond. *Finite Elements III*. Springer, Mar. 2021. DOI: 10.1007/978-3-030-57348-5. URL: https://hal.archives-ouvertes.fr/hal-03226051.

[49] Alexandre Ern, Serge Piperno, and Karim DJADEL. "A well-balanced Runge–Kutta Discontinuous Galerkin method for the Shallow-Water Equations with flooding and drying". working paper or preprint. June 2007. URL: https://hal.archives-ouvertes.fr/hal-00153788.

[50] Leopold Fejér. "Bestimmung derjenigen Abszissen eines Intervalles, für welche die Quadratsumme der Grundfunktionen der Lagrangeschen Interpolation im Intervalle ein möglichst kleines Maximum besitzt". In: *Annali della Scuola Normale Superiore di Pisa-Classe di Scienze* 1.3 (1932), pp. 263–276.

[51] J. Galewsky, RICHARD SCOTT, and Lorenzo Polvani. "An initial-value problem to test numerical models of the shallow-water equations". In: *Tellus Series A-dynamic Meteorology and Oceanography - TELLUS A-DYN METEOROL OCEANOG* 56 (Oct. 2004), pp. 429–440. DOI: 10.1111/j.1600-0870.2004.00071.x.

[52] Thierry Gallouët, Jean-Marc Hérard, and Nicolas Seguin. "Some approximate Godunov schemes to compute shallow-water equations with topography". In: *Computers and Fluids* 32.4 (2003), pp. 479–513. URL: https://hal.archives-ouvertes.fr/hal-01290889.

[53] Jean-Frédéric Gerbeau and Benoît Perthame. *Derivation of Viscous Saint-Venant System for Laminar Shallow Water; Numerical Validation*. Research Report RR-4084. Projet M3N. INRIA, 2000. URL: https://hal.inria.fr/inria-00072549.

[54] Francis Giraldo, Jan Hesthaven, and Tim Warburton. "Nodal High-Order Discontinuous Galerkin Methods for the Spherical Shallow Water Equations". In: *Journal of Computational Physics* 181 (Sept. 2002), pp. 499–525. DOI: 10.1006/jcph.2002.7139.

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

[56] S. Gottlieb and S.J. Ruuth. "Optimal Strong-Stability-Preserving Time-Stepping Schemes with Fast Downwind Spatial Discretizations". In: *J. Sci. Comput.* 27(1-3) (2006), pp. 289–303.

[57] J. M. Greenberg and A. Y. Leroux. “A Well-Balanced Scheme for the Numerical Processing of Source Terms in Hyperbolic Equations”. In: *SIAM Journal on Numerical Analysis* 33.1 (1996), pp. 1–16. DOI: 10.1137/0733001. eprint: https://doi.org/10.1137/0733001. URL: https://doi.org/10.1137/0733001.

[58] Jean-Luc Guermond, Richard Pasquetti, and Bojan Popov. “Entropy viscosity method for nonlinear conservation laws”. In: *Journal of Computational Physics* 230.11 (May 2011), pp. 4248–4267. DOI: 10.1016/j.jcp.2010.11.043.

[59] Guillermo Hauke. “A symmetric formulation for computing transient shallow water flows”. In: *Computer Methods in Applied Mechanics and Engineering* 163.1 (1998), pp. 111–122. ISSN: 0045-7825. DOI: https://doi.org/10.1016/S0045-7825(98)00007-3. URL: https://www.sciencedirect.com/science/article/pii/S0045782598000073.

[60] Jan Hesthaven and Tim Warburton. “Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications”. In: vol. 54. Jan. 2007.

[61] Jan S. Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods*. 1st ed. Texts in Applied Mathematics. Springer, New York, NY.

[62] Tuong Hoang et al. “Skeleton-stabilized immersogeometric analysis for incompressible viscous flow problems”. In: *Computer Methods in Applied Mechanics and Engineering* 344 (2019), pp. 421–450. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2018.10.015. URL: https://www.sciencedirect.com/science/article/pii/S0045782518305188.

[63] M.E. Hubbard and P. Garcia-Navarro. “Flux Difference Splitting and the Balancing of Source Terms and Flux Gradients”. In: *Journal of Computational Physics* 165.1 (2000), pp. 89–125. ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.2000.6603. URL: https://www.sciencedirect.com/science/article/pii/S0021999100966038.

[64] Thomas J. R. Hughes, Guglielmo Scovazzi, and Tayfun E. Tezduyar. “Stabilized Methods for Compressible Flows”. In: *Journal of Scientific Computing* 43.3 (2010), pp. 343–368. ISSN: 1573-7691. DOI: 10.1007/s10915-008-9233-5. URL: https://doi.org/10.1007/s10915-008-9233-5.

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

[66] T.J.R. Hughes, L.P. Franca, and M. Mallet. “A new finite element formulation for CFD I: symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics”. In: *Comp. Meth. Appl. Mech. Engrg.* 54 (1986), pp. 223–234.

[67] T.J.R. Hughes, G. Scovazzi, and T. Tezduyar. “Stabilized Methods for Compressible Flows”. In: *J. Sci. Comp.* 43 (2010), pp. 343–368.

[68] T.J.R. Hughes and T.E. Tezduyar. “Development of time-accurate finite element techniques for first order hyperbolic systems with emphasys on the compressible Euler equations”. In: *Comp. Meth. Appl. Mech. Engrg.* 45.1-3 (1984), pp. 217–284.

[69] Sébastien Jund and Stéphanie Salmon. "Arbitrary high-order finite element schemes and high-order mass lumping". In: *International Journal of Applied Mathematics and Computer Science* 17.3 (2007), p. 375.

[70] Dimitri Komatitsch et al. "Wave propagation in 2-D elastic media using a spectral element method with triangles and quadrangles". In: *Journal of Computational Acoustics* 9 (June 2001), pp. 703–718. DOI: 10.1142/S0218396X01000796.

[71] Dmitri Kuzmin. *Entropy stabilization and property-preserving limiters for discontinuous Galerkin discretizations of nonlinear hyperbolic equations*. 2020. arXiv: 2004.03521 [math.NA].

[72] Dmitri Kuzmin. "Monolithic convex limiting for continuous finite element discretizations of hyperbolic conservation laws". In: *Computer Methods in Applied Mechanics and Engineering* 361 (2020), p. 112804. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2019.112804. URL: https://www.sciencedirect.com/science/article/pii/S0045782519306966.

[73] Dmitri Kuzmin and Manuel Quezada de Luna. "Algebraic entropy fixes and convex limiting for continuous finite element discretizations of scalar hyperbolic conservation laws". In: *Computer Methods in Applied Mechanics and Engineering* 372 (2020), p. 113370. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2020.113370. URL: https://www.sciencedirect.com/science/article/pii/S0045782520305557.

[74] Dmitri Kuzmin and Manuel Quezada de Luna. "Entropy conservation property and entropy stabilization of high-order continuous Galerkin approximations to scalar conservation laws". In: *Computers & Fluids* 213 (2020), p. 104742. ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2020.104742. URL: https://www.sciencedirect.com/science/article/pii/S0045793020303121.

[75] Dmitri Kuzmin and Manuel Quezada de Luna. "Subcell flux limiting for high-order Bernstein finite element discretizations of scalar hyperbolic conservation laws". In: *Journal of Computational Physics* 411 (2020), p. 109411. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2020.109411. URL: http://www.sciencedirect.com/science/article/pii/S0021999120301856.

[76] Mats G Larson and Sara Zahedi. "Stabilization of high order cut finite element methods on surfaces". In: *IMA Journal of Numerical Analysis* 40.3 (Apr. 2019), pp. 1702–1745.

[77] S. Le Roy et al. "Coastal flooding of urban areas by overtopping: dynamic modelling application to the Johanna storm (2008) in Gâvres (France)". In: *Natural Hazards and Earth System Sciences* 15.11 (2015), pp. 2497–2510. DOI: 10.5194/nhess-15-2497-2015. URL: https://nhess.copernicus.org/articles/15/2497/2015/.

[78] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002. DOI: 10.1017/CBO9780511791253

[79] R. J. LeVeque. *Numerical Methods for Conservation Laws*. 2nd ed. Lectures in Mathematics ETH Zürich. Birkhäuser, Basel, 1992.

[80] Randall J. LeVeque. “Balancing Source Terms and Flux Gradients in High-Resolution Godunov Methods: The Quasi-Steady Wave-Propagation Algorithm”. In: *Journal of Computational Physics* 146.1 (1998), pp. 346–365. ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1998.6058. URL: https://www.sciencedirect.com/science/article/pii/S0021999198960582.

[81] Tao Liu et al. “Dispersion analysis of the spectral element method using a triangular mesh”. In: *Wave Motion* 49 (June 2012), pp. 474–483. DOI: 10.1016/j.wavemoti.2012.01.003.

[82] Youshan Liu et al. “Higher-order triangular spectral element method with optimized cubature points for seismic wavefield modeling”. In: *Journal of Computational Physics* 336 (2017), pp. 458 –480. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2017.01.069. URL: http://www.sciencedirect.com/science/article/pii/S0021999117300852.

[83] Julie Llobell, Sebastian Minjeaud, and Richard Pasquetti. “High Order CG Schemes for KdV and Saint-Venant Flows”. In: Feb. 2020, pp. 341–352. ISBN: 978-3-030-30704-2. DOI: 10.1007/978-3-030-30705-9_30.

[84] R. Abgrall M. Ricchiuto and H. Deconinck. “Application of conservative residual distribution schemes to the solution of the shallow water equations on unstructured meshes”. In: *Journal of Computational Physics* 222 (Mar. 2007), pp. 287–331. DOI: 10.1016/j.jcp.2006.06.024.

[85] D. Torlo M. Ricchiuto. “Analytical travelling vortex solutions of hyperbolic equations for validating very high order schemes”. In: *arXiv preprint arXiv:2109.10183* (2021).

[86] S. Michel et al. *Stability analysis of several FEM methods 2D: results*. https://gitlab.inria.fr/simichel/stability-analysis-of-several-fem-methods-in-2d.-results. 2021.

[87] S. Michel et al. *Stability analysis of several FEM methods: results and code*. https://gitlab.inria.fr/dtorlo1/stability-analysis-of-several-fem-methods-results-and-code.git. 2021.

[88] Sixtine Michel et al. “Spectral Analysis of Continuous FEM for Hyperbolic PDEs: Influence of Approximation, Stabilization, and Time-Stepping”. In: *Journal of Scientific Computing* 89.2 (2021), p. 31. ISSN: 0885-7474. DOI: 10.1007/s10915-021-01632-7.

[89] John Miller. “On the Location of Zeros of Certain Classes of Polynomials with Applications to Numerical Analysis”. In: *Journal of the Institute of Mathematics and its Applications* 8 (Dec. 1971). DOI: 10.1093/imamat/8.3.397.

[90] Michael Minion. “Semi-Implicit Spectral Deferred Correction Methods For Ordinary Differential Equations”. In: *Communications in Mathematical Sciences* 1 (Nov. 2003). DOI: 10.4310/CMS.2003.v1.n3.a6.

[91] Rodrigo C. Moura et al. “Spatial eigenanalysis of spectral/hp continuous Galerkin schemes and their stabilisation via DG-mimicking spectral vanishing viscosity for high Reynolds number flows”. In: *Journal of Computational Physics* 406 (2020), p. 109112. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2019.109112. URL: http://www.sciencedirect.com/science/article/pii/S0021999119308174.

[92] A. Nicolae Lerma R. Pedreros B. Ayache M. Garcin T. Bulteau A. Hoareau Mugica F. Apris. *Caractérisation de l'aléa submersion marine dans le cadre des PPRL du Bassin d'Arcachon.* BRGM/RP-64807-FR. 2016.

[93] W. Mulder. "Higher-order mass-lumped finite elements for the wave equation". In: *Journal of Computational Acoustics* 14 (Jan. 2001), pp. 671–680.

[94] W. Mulder. "New triangular mass-lumped finite elements of degree 6 for wave propagation". In: *Progress In Electromagnetics Research* 141 (Jan. 2013), pp. 671–692. DOI: 10.2528/PIER13051308.

[95] Héloïse Muller et al. "Storm Impact on a French coastal dune system: morphodynamic modeling using X-beach". In: *Infra-gravity waves : from driving mechanisms to impacts*. La Rochelle, France, Mar. 2016. URL: https://hal-brgm.archives-ouvertes.fr/hal-01275253.

[96] Ioannis Nikolos and Argiris Delis. "An unstructured node-centred finite volume scheme for shallow water flows with wet/dry fronts over complex topography". In: *Computer Methods in Applied Mechanics and Engineering* 198 (Oct. 2009), pp. 3723–3750. DOI: 10.1016/j.cma.2009.08.006.

[97] Sebastian Noelle, Yulong Xing, and Chi-Wang Shu. "High-order well-balanced finite volume WENO schemes for shallow water equation with moving water". In: *Journal of Computational Physics* 226 (Sept. 2007), pp. 29–58. DOI: 10.1016/j.jcp.2007.03.031.

[98] Sebastian Noelle, Yulong Xing, and Chi-Wang Shu. "High order well-balanced schemes". In: 2010.

[99] Sebastian Noelle et al. "Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows". In: *Journal of Computational Physics* 213 (Apr. 2006), pp. 474–499. DOI: 10.1016/j.jcp.2005.08.019.

[100] Philipp Öffner and Davide Torlo. "Arbitrary high-order, conservative and positivity preserving Patankar-type deferred correction schemes". In: *Applied Numerical Mathematics* 153 (2020), pp. 15 –34. ISSN: 0168-9274. DOI: https://doi.org/10.1016/j.apnum.2020.01.025. URL: http://www.sciencedirect.com/science/article/pii/S016892742030026X.

[101] Richard Pasquetti. "Viscous stabilizations for high order approximations of Saint-Venant and Boussinesq flows". In: *Lecture Notes in computational Science and Engineering*. Ed. by Marco Bittencourt, Ney Dumont, and Jan S. Hesthaven. Vol. 119. Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016. Springer, 2017, pp. 519–531. DOI: 10.1007/978-3-319-65870-4_37. URL: https://hal.univ-cotedazur.fr/hal-01657795.

[102] Richard Pasquetti and Francesca Rapetti. "Cubature points based triangular spectral elements: An accuracy study". In: *Journal of Mathematical Study* 51.1 (2018), pp. 15–25. DOI: 10.4208/jms.v51n1.18.02. URL: https://hal.univ-cotedazur.fr/hal-01954133.

[103] Rodrigo Pedreros. *État des connaissances sur la dynamique et la cinétique de la submersion marine et des méthodologies d'évaluation*. Rapport final - BRGM. Mar. 2016.

[104] D. Torlo R. Abgrall. *High Order Asymptotic Preserving Deferred Correction Implicit-Explicit Schemes for Kinetic Models*. 2020. DOI: 10.1137/19M128973X. eprint: https://doi.org/10.1137/19M128973X. URL: https://doi.org/10.1137/19M128973X.

[105] Jan P. Öffner S. Tokareva R. Abgrall J. Nordström. "Analysis of the SBP-SAT Stabilization for Finite Element Methods Part II: Entropy Stability". In: *Commun. Appl. Math. Comput.* (2021), pp. 2661–8893.

[106] P. Öffner S. Tokareva R. Abgrall J. Nordström. "Analysis of the SBP-SAT Stabilization for Finite Element Methods Part I: Linear Problems". In: *Journal of Scientific Computing* 85.43 (2020), pp. 1573–7691.

[107] S. Tokareva R. Abgrall P. Bacigaluppi. "High-order residual distribution scheme for the time-dependent Euler equations of fluid dynamics". In: *Computers & Mathematics with Applications* 78.2 (2018), pp. 274–297. ISSN: 0898-1221. DOI: https://doi.org/10.1016/j.camwa.2018.05.009. URL: http://www.sciencedirect.com/science/article/pii/S0898122118302712.

[108] E. Burman S. Sherwin R. Moura A. F. De Castro da Silva. *Eigenanalysis of gradient-jump penalty (GJP) stabilisation for CG*. Feb. 2020. DOI: 10.13140/RG.2.2.32887.85924.

[109] Mario Ricchiuto. "An explicit residual based approach for shallow water flows". In: *Journal of Computational Physics* 280 (2015), pp. 306–344. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2014.09.027. URL: https://www.sciencedirect.com/science/article/pii/S0021999114006639.

[110] Mario Ricchiuto and Andreas Bollermann. "Stabilized residual distribution for shallow water simulations". In: *J. Comput. Physics* 228 (Mar. 2009), pp. 1071–1115. DOI: 10.1016/j.jcp.2008.10.020.

[111] Mario Ricchiuto and Andreas Bollermann. "Stabilized residual distribution for shallow water simulations". In: *Journal of Computational Physics* 228.4 (2009), pp. 1071–1115. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2008.10.020. URL: https://www.sciencedirect.com/science/article/pii/S0021999108005391.

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

[113] U. Skre Fjordholm S. Mishra and R. Abgrall. *Numerical methods for conservation laws and related equations*. 2016. URL: www.math.uzh.ch/index.php?id=ve_vo_det&key1=0&key2=2659&key3=487&semId=32.

[114] Mohammed Seaid. "Non-oscillatory relaxation methods for the shallow-water equations in one and two space dimensions". In: *International Journal for Numerical Methods in Fluids* 46 (Oct. 2004), pp. 457 –484. DOI: 10.1002/fld.766.

[115] S.J. Sherwin and G.E. Karniadakis. "A triangular spectral element method; applications to the incompressible Navier-Stokes equations". In: *Computer Methods in Applied Mechanics and Engineering* 123.1 (1995), pp. 189 –229. ISSN: 0045-7825. DOI: https://doi.org/10.1016/0045-7825(94)00745-9. URL: http://www.sciencedirect.com/science/article/pii/0045782594007459.

[116] Spencer Sherwin. "Dispersion Analysis of the Continuous and Discontinuous Galerkin Formulations". In: *Discontinuous Galerkin Methods* 11 (Aug. 1999). DOI: 10.1007/978-3-642-59721-3_43.

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

[118] R.J. Spiteri and S.J. Ruuth. "A new class of optimal high-order strong-stability-preserving time discretization methods". In: *SIAM J. Numer. Anal.* 40(2) (2002), pp. 469–491.

[119] Joanna Szmelter and Piotr K. Smolarkiewicz. "An edge-based unstructured mesh discretisation in geospherical framework". In: *Journal of Computational Physics* 229.13 (2010), pp. 4980–4995. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2010.03.017. URL: https://www.sciencedirect.com/science/article/pii/S0021999110001270.

[120] Joanna Szmelter and Piotr K. Smolarkiewicz. "An edge-based unstructured mesh framework for atmospheric flows". In: *Computers & Fluids* 46 (2011), pp. 455–460.

[121] Eitan Tadmor and Weigang Zhong. "Energy-Preserving and Stable Approximations for the Two-Dimensional Shallow Water Equations". In: Oct. 2008, pp. 67–94. ISBN: 978-3-540-68848-8. DOI: 10.1007/978-3-540-68850-1_4.

[122] M. A. Taylor, B. A. Wingate, and R. E. Vincent. "An Algorithm for Computing Fekete Points in the Triangle". In: *SIAM J. Numer. Anal.* 38.5 (Oct. 2000), pp. 1707–1720. ISSN: 0036-1429. DOI: 10.1137/S0036142998337247. URL: https://doi.org/10.1137/S0036142998337247.

[123] Nathalie Tordjman. "Éléments finis d'ordre élevé avec condensation de masse pour l'équation des ondes". Thèse de doctorat dirigée par Cohen, Gary Chalom Mathématiques appliquées à l'ingénierie Paris 9 1995. PhD thesis. Université Paris VI, 1995, 1 vol. (300 p.) URL: http://www.theses.fr/1995PA090002.

[124] Eleuterio Toro. "Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction". In: Jan. 2009. DOI: 10.1007/b79761.

[125] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[126] David Williamson. "The Evolution of Dynamical Cores for Global Atmospheric Models". In: *J. Royal Met. Soc. Japan* 85B (July 2007), pp. 241–269. DOI: 10.2151/jmsj.85B.241.

[127] David L. Williamson et al. "A standard test set for numerical approximations to the shallow water equations in spherical geometry". In: *Journal of Computational Physics* 102.1 (1992), pp. 211–224. ISSN: 0021-9991. DOI: https://doi.org/10.1016/S0021-9991(05)80016-6. URL: https://www.sciencedirect.com/science/article/pii/S0021999105800166.

[128] Yulong Xing and Chi-Wang Shu. "High order finite difference WENO schemes with the exact conservation property for the shallow water equations". In: *Journal of Computational Physics* 208.1 (2005), pp. 206–227. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2005.02.006. URL: https://www.sciencedirect.com/science/article/pii/S002199910500094X.

[129] Yulong Xing and Chi-Wang Shu. "High order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms". In: *Journal of Computational Physics* 214.2 (2006), pp. 567–598. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2005.10.005. URL: https://www.sciencedirect.com/science/article/pii/S0021999105004626.

[130] Yulong Xing, Chi-Wang Shu, and Sebastian Noelle. *On the advantage of well-balanced schemes for moving-water equilibria of the shallow water equations*. 2015. arXiv: 1502.00800 [math.NA].

[131] Xiangxiong Zhang. "On positivity-preserving high order discontinuous Galerkin schemes for compressible Navier–Stokes equations". In: *Journal of Computational Physics* 328 (2017), pp. 301–343. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2016.10.002. URL: https://www.sciencedirect.com/science/article/pii/S0021999116304958.

[132] Xiangxiong Zhang and Chi-Wang Shu. "On maximum-principle-satisfying high order schemes for scalar conservation laws". In: *Journal of Computational Physics* 229.9 (2010), pp. 3091–3120. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2009.12.030. URL: https://www.sciencedirect.com/science/article/pii/S0021999109007165.