Extension of the Spectral Difference method to simplex cells and hybrid grids

Adèle Veilleux

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Institut National Polytechnique de Toulouse (INP Toulouse)*

Présentée et soutenue le *24 mars 2021* **par :**

Adèle VEILLEUX

Extension of the Spectral Difference method to simplex cells and hybrid grids

---

JURY

Chunlei LIANG	Professeur - Clarkson University	Rapporteur
Georg MAY	Professeur Agrégé - VKI	Rapporteur
Hélène BARUCQ	Directrice de Recherche - INRIA	Présidente
Eusebio VALERO	Professeur - UPM	Examinateur
Hugues DENIAU	Ingénieur de Recherche - ONERA	Invité
Jean-Luc ESTIVALEZES	Professeur Agrégé - INP Toulouse	Invité
Guillaume PUIGT	Maitre de Recherche - ONERA	Directeur
Guillaume DAVILLER	Chercheur Senior - CERFACS	Co-Directeur

École doctorale et spécialité :

MEGEP : Dynamique des fluides

Unités de Recherche :

ONERA – Département Multi-Physique pour l'Énergétique, unité Instabilités, Transition, contrôle ACoustique (DMPE/ITAC)

CERFACS – Équipe Computational Fluid Dynamics (CFD)

Directeur de Thèse :

Guillaume PUIGT

Rapporteurs :

Chunlei LIANG et *Georg MAY*

Extension of the Spectral Difference method to simplex cells and hybrid grids

---

Abstract This thesis examines the extension of the Spectral Difference (SD) method on unstructured hybrid grids involving simplex cells (triangles, tetrahedra) and prismatic elements. The Spectral Difference method is part of high-order spectral discontinuous numerical methods. These methods rely on piecewise continuous polynomial approximation to obtain high-order accuracy with a good parallel efficiency. The standard SD scheme is first presented in the one-dimensional case and then for tensor-product elements (quadrilaterals and hexahedra). The treatment of simplex cells using Raviart-Thomas elements is detailed for triangles (in 2D) and tetrahedra (in 3D), followed by the implementation for prismatic elements.

The linear stability of the Spectral Difference method using Raviart-Thomas elements (SDRT) is studied on triangles and tetrahedra. The SDRT scheme stability is strongly dependent on the interior flux points location. On triangles, the SDRT implementation based on interior flux points located at Williams-Shunn-Jameson quadrature points is found stable up to the fourth-order of accuracy but shown as spatially unstable for higher orders. Nevertheless, it is shown that this implementation can be stabilized for fifth- and sixth-order schemes using suitable temporal integration schemes. This approach being submitted to strict conditions, an optimization of the interior flux points location is conducted to determine spatially stable SDRT formulations for orders higher than four. The optimization process leads to spatially stable schemes up to the sixth-order of accuracy. Finally, the stability analysis on tetrahedra proves that the SDRT scheme based on the interior flux points located at Shunn-Ham quadrature points is stable up to the third-order.

The SD/SDRT numerical method is validated on several academic cases for first and second-order Partial Differential Equations (linear advection equation, Euler equations, Navier-Stokes equations). Both proposed implementations (based either on Williams-Shunn-Jameson quadrature points or optimization points) are used. Numerical experiments involve grids composed of quadratic triangles, linear tetrahedral elements as well as 2D hybrid meshes.

Keywords: High-order discontinuous method, Spectral Difference, Hybrid grid, Simplex cells, Stability analysis

---

CERFACS

ONERA
THE FRENCH AEROSPACE LAB

Remerciements

Les travaux menés pendant ces trois années de thèse ont été rendus possible grâce à l'aide et le soutien d'une multitude de personnes. Je profite de ces quelques lignes pour leur adresser mes remerciements.

Je remercie d'abord les membres du jury pour avoir accepté d'évaluer mes travaux de thèse et pour leur participation à la soutenance, bien que dans un contexte peu habituel. La soutenance à distance m'a permis d'échanger avec un jury extrêmement compétent sur le sujet. J'exprime toute ma reconnaissance à Chunlei Liang et Georg May, qui ont accepté de rapporter cette thèse et dont les remarques et suggestions m'ont beaucoup apporté. Merci à Hélène Barucq d'avoir présidé ce jury et à Eusebio Valero et Jean-Luc Estivalezes d'en avoir fait partie : les échanges scientifiques que nous avons eu ont été très intéressants et contribueront certainement à orienter les futures thématiques de recherche sur Jaguar.

Cette thèse a été cofinancée par deux laboratoires, dans lesquels j'ai été (plus que) bien accueillie.
Côté Onera, j'ai intégré l'unité ITAC au sein du DMPE ; merci à Pierre Millan et Jérôme Anthoine pour leur accueil et merci à Estelle Piot pour son accompagnement et son suivi. Je remercie également Valérie, Corinne et Chrystelle pour leur aide précieuse au secrétariat, ainsi que l'ensemble du personnel du poste de garde. Merci aussi à tous les permanents du DMPE avec entre autres Thomas B., Fabien, Julien, Maxime, François, avec qui j'ai pu échanger autant autour du travail (comme aux JDD) que d'une bière (comme aux *repas* JDD) pendant ces trois années.
Côté Cerfacs, je remercie Jean-François Boussuge et Thierry Poinsot pour m'avoir permis d'évoluer au sein de l'équipe CFD. Le quotidien de l'équipe CFD (et de tout le Cerfacs d'ailleurs) est en particulier rythmé par le soutien et la bonne humeur de l'équipe admin avec Marie, Nicole, Jade, Michèle, Lydia et bien sûr Chantal ! Je voudrais aussi remercier l'équipe CSG - Fred, Isabelle, Fabrice, Gérard, Patrick, Nicolas -, pour toutes les fois où ils m'ont sauvée d'heures d'arrachage de cheveux devant mon mac, même à distance pendant les confinements. Un grand merci à Fred pour son aide le jour de la soutenance. Merci à l'ensemble des permanents, en particulier à Corentin pour son aide en début de thèse, et à Eléonore pour son suivi en tant que marraine de thèse.

Mes remerciements s'adressent ensuite évidemment à mes encadrants de thèse. Hugues et Guillaume, vos compétences et caractères à chacun ont formé à l'Onera une équipe d'encadrants de rêve ! Vos explications, encouragements, conseils et suggestions m'ont énormément apporté. Ça a été un plaisir d'être votre thésarde, autant sur le plan scientifique que sur toutes les discussions plus légères qu'on a pu avoir à côté. Vous avez été hyper présents et disponibles, y compris pendant les confinements, et je suis ravie qu'on continue à travailler ensemble et à échanger pour l'année qui vient. Côté Cerfacs, merci à Guillaume D pour sa lecture attentive de ce manuscrit et plus généralement pour ses remarques pertinentes sur le contenu scientifique produit. Merci aussi à Jeff de m'avoir souvent sorti la tête du guidon en m'apportant un regard plus applicatif (et moins mathématique !) sur Jaguar.

Je voudrais ensuite remercier ceux avec qui j'ai partagé le bureau J13 et qui ont dû me supporter pendant trois ans. Laurent, merci pour toutes nos discussions, pour l'aide que tu m'as apportée à chaque fois que tu en as eu l'occasion. Tu vas bientôt partir vers de nouveaux horizons mais n'oublie pas que Philippe, je sais où te caches ! Marco, merci pour tes précieux conseils pendant la thèse (le 'adele.py' continue à me servir en post-doc...), et merci d'avoir complété ton boulot de chercheur par un travail de garderie dans le bureau J13. C'était un plaisir de venir bosser pour vous retrouver tous les deux !

J'adresse également une mention spéciale à la team des dresseurs, Marc, Thomas M, Alexia et Rémi : l'équipe jaja a permis de toujours avoir quelqu'un vers qui se tourner en cas de session de débogage interminable (et d'en profiter pour se vanner un peu sur slack). Thomas, je t'envoie toutes mes bonnes ondes pour la fin de ta thèse, tu tiens le bon bout ! Merci à toi pour nos appels à rallonge à propos du félin (dont les sujets de discussion ont souvent dérivé !).

Ces trois années auront certes été rythmées par des heures de travail laborieux mais elles ont aussi (surtout !) vu défiler un certain nombre de stagiaires, doctorants et post-doctorants qui ont fait de cette thèse une période inoubliable.

Je pense d'abord à ceux qui ont débuté cette aventure avec moi à l'Onera : Julien, Loïc, Damien et Gaétan. J'ai découvert Toulouse avec vous (surtout en ce qui concerne l'Arty et la Paillotte) et malgré une présence sur site de ma part qui s'est légèrement raréfiée au fil du temps, vous avez été là pour tous les moments importants (coucou le 38-37 et la Tantina). Merci à Virgile et Pierre d'avoir complété l'OST dans ses aventures au ski ou au bar ! Je remercie aussi la team des anciens, avec entre autres Rémi, Loïc J, Loïc VG, Anthony D, les "nouveaux" arrivants, dont Boris, Nicolas P, Thomas A, Romain, et l'ensemble des doctorants et post-doc avec qui j'ai pu échanger à l'Onera.

La suite de la thèse au Cerfacs a donné lieu à un nombre (très) important de belles rencontres. Merci d'abord à Danny, qui m'a intégrée à la team LBM à mon arrivée au Cerfacs (pour le meilleur ou pour le pire, ça reste à voir !). Le groupe LBM a bien sûr égayé les journées au Cerfacs avec sa bonne humeur mais n'en est pas resté là. Thomas, merci pour ta motivation sans limite pour organiser les sessions kite et tes conseils précieux pour le backloop. Tu es sûrement celui qui m'a le plus vu me prendre des boîtes ! Florian, merci d'avoir partagé avec nous tes talents de chanteur, ton interprétation d'IAM restera dans les mémoires ! Nos thèses se sont terminées en même temps et ça a été vraiment rassurant de partager ça avec toi. Merci aussi à Bastien pour ses encouragements pendant le permis moto et à Gauthier pour son intérêt pour mes problèmes d'analyse de stabilité linéaire (et pour les ragots...).

Cette équipe s'est étendue à la team kite, d'abord avec Luc et Romain puis David et Christopher. Merci à vous pour nos voyages à l'autre bout du monde et pour tous les moments qu'on a partagés (la bonne époque du CSG se fait regretter). Un immense merci à David, mon éternel voisin ; de nos footings occasionnels à nos soirées cocktail, ton amitié et ta présence ces dernière années me sont devenues indispensable.

Un seul groupe de copains au Cerfacs aurait pu être suffisant, mais j'ai eu ration double : merci à l'équipe combustion/escalade/pause-de-18h avec Etienne, Aurel, Victor, Simon, Paul, Quentin, Julien, Théo, Antoine, Guillaume, Wilca et évidemment Thomas L. Thomas, merci pour toutes nos pauses café, nos sessions wake, et aussi pour ta présence dans les moments moins faciles. Tu fais partie de ces amitiés qui ne s'arrêteront pas avec cette thèse ! Plus largement, merci à tous les autres doctorants et post-doc que j'ai pu croiser, avec entre autres Valentin, Robin, Pamphil, Mathieu, Maxou, Johan...

Un immense merci aussi à la team Time Toulouse, avec Thomas M, Paul, Camille, Alexis et Enora, Nico, Thibault, Mag et Minh. Vous avez été une bouffée d'air frais pendant la période de rédaction ! Je ne risque pas d'oublier la soirée de mon envoi aux rapporteurs et les bougies sur St Nectaire qui l'ont accompagnée... Enfin, merci à Thomas d'avoir partagé avec moi ces périodes de confinement et de fin de thèse qui resteront de super moments.

J'ai finalement une pensée pour les amis qui sont présents depuis bien longtemps et qui ont suivi cette affaire d'un peu plus loin que Toulouse. Merci aux Marseillais avec qui j'ai eu la chance de partager les belles années que sont celles d'école d'ingé : Pauline, Hugo, Quentin, Anna, Conrad, Lorenz, Sam, Fabish, Matthieu, Pierre, Océ, Vincent et tous les autres. La distance rend les retrouvailles plus compliquées mais toujours régulières. Merci aussi à tous ceux qui m'ont encouragée à fond pendant la thèse, merci à Vanessa, à Ashkan, à Idriss, à mes copines bordelaises, Manon et Valentine, et merci à Marie, qui m'écoute et m'apaise depuis bientôt dix ans.

Mes derniers remerciements s'adressent à ma famille. Merci à vous, papa, maman, Salomé, de m'avoir supportée tour à tour en confinement, de vous être chacun intéressés à ce sujet pourtant bien loin de vos préoccupations habituelles et d'avoir patiemment écouté mes tentatives d'explications. Je suis très fière d'être votre fille, et Sasa ta sœur, et je sais que vous étiez très fiers de me voir soutenir ! Merci pour votre soutien inconditionnel et vos encouragements.

Merci à tous.

Contents

Abstract iii

Remerciements v

Nomenclature xiii

Résumé Étendu en Français xix

1. Introduction 1
- 1.1. Computational Fluid Dynamics . . . 2
  - 1.1.1. Mesh Requirements . . . 2
  - 1.1.2. Numerical Methods . . . 3
  - 1.1.3. Computational Efficiency . . . 4
- 1.2. Turbulence Modeling and Computation . . . 5
  - 1.2.1. Direct Numerical Simulation . . . 6
  - 1.2.2. Reynolds-Averaged Navier–Stokes Approach . . . 6
  - 1.2.3. Large Eddy Simulation . . . 7
- 1.3. High-Order Discontinuous Methods for Unstructured Grid Based on the Integral Form 9
  - 1.3.1. Discontinuous Galerkin Method . . . 9
  - 1.3.2. Spectral Volume Method . . . 10
- 1.4. High-Order Discontinuous Methods for Unstructured Grid Based on the Strong Form 11
  - 1.4.1. Flux Reconstruction Method . . . 11
  - 1.4.2. Spectral Difference Method . . . 12
- 1.5. Motivations for the Thesis . . . 14
- 1.6. Outline . . . 14
- 1.7. Publications . . . 15
  - 1.7.1. Papers . . . 15
  - 1.7.2. Presentations . . . 15

2. The Spectral Difference Method 17
- 2.1. One-Dimensional Formulation . . . 18
- 2.2. Tensor Product Elements Formulation . . . 21
- 2.3. Triangular and Tetrahedral Elements Formulation . . . 25
  - 2.3.1. Algebraic Framework . . . 25
    - 2.3.1.1. Polynomial Expansion on Simplices . . . 25
    - 2.3.1.2. Proriol-Koornwinder-Dubiner Basis . . . 26
  - 2.3.2. General Principle . . . 27
  - 2.3.3. Raviart-Thomas Basis . . . 28
  - 2.3.4. The Spectral Difference Scheme Using Raviart-Thomas Elements . . . 30
    - 2.3.4.1. Reference Element . . . 30
    - 2.3.4.2. Solution Approximation . . . 31
    - 2.3.4.3. Solution Computation at Flux Points . . . 31
    - 2.3.4.4. Definition of the Flux Polynomial from the Set of Flux Values at Flux Points . . . 32

2.3.4.5. Determination of the Scalar Flux Values . . . 33
2.3.4.6. Differentiation of the Flux Polynomial at Solution Points . . . 33
2.3.4.7. Semi-Discretized Form . . . 34
2.3.5. Comparison Between Spectral Difference and Flux Reconstruction Methods for Simplex Elements . . . 34
2.4. Prismatic Elements Formulation . . . 36
2.4.1. Reference Element . . . 36
2.4.2. Solution Approximation . . . 36
2.4.3. Definition of the Flux Polynomial from the Set of Flux Values at Flux Points . . . 37
2.4.4. Differentiation of the Flux Polynomial at Solution Points . . . 38
2.4.5. Semi-Discretized Form . . . 38
2.5. Pyramidal Elements . . . 39

3. Linear Stability Analysis for the Spatial Discretization 41
3.1. Stability Analysis Methods . . . 42
3.1.1. Eigenvalue Analysis . . . 42
3.1.1.1. Matrix Form of the Spatial Operator . . . 42
3.1.1.2. Impact of the Different Parameters on the Stability . . . 43
3.1.2. Fourier Stability Analysis . . . 45
3.1.3. Stability Analysis on a Fixed Number of Cells . . . 46
3.2. Standard One-Dimensional Spectral Difference Scheme . . . 47
3.2.1. Matrix form . . . 47
3.2.2. Stability Analysis on a Fixed Number of Cells . . . 48
3.3. Spectral Difference Using Raviart-Thomas Elements Scheme on Triangular Elements . . . 49
3.3.1. Flux Points Numbering . . . 49
3.3.2. Interior Flux Points Locations Based on Quadrature Rules . . . 49
3.3.3. Fourier Stability Analysis . . . 50
3.3.4. Stability Analysis on a Fixed Number of Cells . . . 53
3.4. Spectral Difference Scheme Using Raviart-Thomas Elements on Tetrahedral Elements 59
3.4.1. Reference Element Convention and Points Numbering . . . 59
3.4.2. Choice of Solution Points and Flux Points Location . . . 61
3.4.3. Matrix Form . . . 61
3.4.4. Fourier Stability Analysis . . . 63

4. Linear Stability Analysis for the Coupled Time-Space Discretization 65
4.1. Temporal Integration . . . 66
4.1.1. Runge-Kutta Integration Methods . . . 66
4.1.2. Matrix Form of the Coupled Time-Space Discretization . . . 67
4.1.3. Fourier Analysis of the Coupled Time-Space Discretization . . . 67
4.2. Triangular Elements . . . 69
4.2.1. Spatially Stable Schemes: $SDRT_2$ and $SDRT_3$ . . . 69
4.2.1.1. Suitable Temporal Schemes . . . 69
4.2.1.2. Highlight of a Severe Stability Condition for Second-Order Temporal Schemes . . . 70
4.2.2. Spatially Unstable Schemes: $SDRT_4$ and $SDRT_5$ . . . 73
4.2.2.1. Analysis for $\theta = \pi/8$ . . . 73
4.2.2.2. Analysis in the General Case . . . 76
4.2.2.3. Analysis for $SDRT_5$, $\theta = 0$ . . . 77
4.2.2.4. Analysis of Other FP Locations . . . 79
4.3. Tetrahedral Elements . . . 80

5. Optimization Process Based on the Fourier Analysis **83**
5.1. Triangular Elements . . . 84
5.1.1. Optimization Algorithm . . . 84
5.1.2. Spatially Stable $SDRT_4$ and $SDRT_5$ Formulations . . . 87
5.1.2.1. Sets of Interior Flux Points . . . 87
5.1.2.2. Fourier Analysis of the Spatial Discretization . . . 88
5.1.2.3. Fourier Analysis of the Coupled Time-Space Discretization . . . 90
5.1.3. Study for the $SDRT_6$ Scheme . . . 92
5.2. Tetrahedral Elements . . . 93
5.2.1. Optimization Algorithm . . . 93
5.2.2. Fourier Analysis of the Spatial Discretization . . . 95
5.2.3. Influence of Flux Points Located on Faces . . . 95

6. Validation for First-Order Partial Differential Equations **97**
6.1. 2D Linear Advection Equation . . . 98
6.1.1. Advection of a Sine . . . 98
6.1.2. Assessment of $SDRT^{WSJ}$ - RK4J Accuracy . . . 99
6.1.3. Assessment of $SDRT^{OPT}$ Accuracy . . . 100
6.2. Convection of an 2D Isentropic Vortex . . . 102
6.2.1. Triangular Grids . . . 103
6.2.1.1. Comparison at 50 Periods . . . 103
6.2.1.2. Convergence Study . . . 104
6.2.2. Hybrid Grids . . . 105
6.2.2.1. Comparison at 5 Periods . . . 105
6.2.2.2. Convergence Study . . . 107
6.3. 3D Euler Equations . . . 108

7. Extension and validation for Second-Order Partial Differential Equations **111**
7.1. Extension of the Spectral Difference Approach for Navier-Stokes Equations on Triangular and Tetrahedral Elements . . . 112
7.2. 2D Viscous Flow Over a NACA0012 Airfoil . . . 114
7.2.1. Case A: Symmetric Subsonic Flow . . . 115
7.2.2. Case B: Asymmetric Subsonic Flow . . . 118
7.2.3. Case C: Transonic Flow . . . 120
7.3. 2D Viscous Flow Around a Circular Cylinder . . . 122
7.4. 3D Taylor-Green Vortex . . . 125

Conclusion et Perspectives **129**
Discussion . . . 129
Perspectives . . . 131

Appendices **133**
A. Jacobian Matrices . . . 135
B. Demonstration of L^2 Orthogonality . . . 147
C. Stability Analysis on a Fixed Number of Cells - Standard One-Dimensional Spectral Difference Scheme . . . 151
D. Matrices Formulation for the Fourier Analysis . . . 153
E. Interior Flux Points Parametrization for the Optimization Algorithm . . . 159
F. Coefficients for the Runge-Kutta Time Discretization . . . 163

List of Figures **165**

List of Tables **170**

List of Algorithms **171**

Bibliography **173**

Nomenclature

Aerodynamic Coefficients

(C_D)	Total drag coefficient
$(C_D)_f$	Skin-friction drag coefficient
$(C_D)_p$	Pressure drag coefficient
C_f	Surface skin-friction coefficient
C_p	Surface pressure coefficient

Basis

Ψ	A complete polynomial basis for the polynomial approximation
Φ	Proriol-Koornwinder-Dubiner basis functions
$\boldsymbol{\psi}$	Raviart-Thomas basis functions
$\boldsymbol{\phi}$	Monomial basis in the Raviart-Thomas space

Coordinates

$\mathbf{x} = (x, y, z)$	Cartesian coordinates in the physical domain
$\boldsymbol{\xi} = (\xi, \eta, \zeta)$	Cartesian coordinates in the reference element
$\boldsymbol{\xi}_j$	Coordinates of the j-th solution point in the reference element
$\boldsymbol{\xi}_k$	Coordinates of the k-th flux point in the reference element

Flux variables

$\mathbf{f}$	Flux vector in the physical domain, $\mathbf{f} = (f, g, h)$ in 3D
$\hat{\mathbf{f}}$	Flux vector in the reference domain, $\hat{\mathbf{f}} = (\hat{f}, \hat{g}, \hat{h})$ in 3D
$\hat{\mathbf{f}}_h$	Polynomial approximation of the flux vector in the reference domain, $\hat{\mathbf{f}}_h = (\hat{f}_h, \hat{g}_h, \hat{h}_h)$ in 3D
f^*	Standard numerical flux in the physical domain in 1D
$\hat{f}^*$	Standard numerical flux in the reference domain in 1D
$(\mathbf{f} \cdot \mathbf{n})^*$	Standard numerical flux in the physical domain in 2D and 3D
$(\hat{\mathbf{f}} \cdot \hat{\mathbf{n}})^*$	Standard numerical flux in the reference domain in 2D and 3D
$\mathbf{f}^i$	Inviscid flux vector in the physical domain

$\mathbf{f}^v$	Viscous flux vector in the physical domain

Matrices

$\mathbf{C}$	Velocity matrix
$\mathbf{D}$	Differentiation matrix
$\hat{\mathbf{F}}$	Flux vector in the reference element at FP
$\mathbf{G}$	Amplification factor of the coupled time/space discretization
$\mathbf{I}$	Identity matrix
$\mathbf{M}$	Spatial operator matrix
$\mathbf{M}_{\mathrm{n}}$	Spatial operator matrix associated with the FNC analysis
$\mathbf{M}_{\mathrm{z}}$	Spatial operator matrix associated with the Fourier analysis
$\mathbf{T}$	Transfer matrix
$\hat{\mathbf{U}}$	Solution vector in the reference element at SP
$\tilde{\mathbf{U}}$	Complex vector composed of the solution harmonic amplitudes
$\hat{\mathbf{V}}$	Solution vector in the reference element at FP
$\mathcal{V}$	Vandermonde matrix

Number of points

N_{SP}	Solution points
N_{SP}^{1d}	Solution points in the 1D element
N_{SP}^{d}	Solution points in the tensor product element in d dimension
N_{SP}^{tri}	Solution points in the triangular element
N_{SP}^{tet}	Solution points in the tetrahedron element
N_{SP}^{pri}	Solution points in the prismatic element
N_{FP}	Flux points
N_{FP}^{1d}	Flux points in the 1D element
N_{FP}^{d}	Flux points in the tensor product element in d dimension
N_{FP}^{s}	Flux points in a simplex cell using the standard SD approach
N_{FP}^{RT}	Flux points in a simplex cell using the SDRT approach
N_{FP}^{tri}	Flux points in the triangular element using the SDRT approach
N_{i}^{tri}	Interior flux points in the triangular element using the SDRT approach
N_{pi}^{tri}	Physical interior flux points in the triangular element using the SDRT approach
N_{e}^{tri}	Flux points located on an edge in the triangular element using the SDRT approach
N_{FP}^{tet}	Flux points in the tetrahedron element using the SDRT approach
N_q	Number of quadrature points

N_i^{tet}	Interior flux points in the tetrahedron element using the SDRT approach
N_{pi}^{tet}	Physical interior flux points in the tetrahedron element using the SDRT approach
N_f^{tet}	Flux points located on a face in the tetrahedron element using the SDRT approach

Reference elements

$\mathcal{H}$	Reference hexahedron
$\mathcal{H}^{PKD}$	Reference hexahedron for the PKD basis definition
$\mathcal{P}$	Reference prism
$\mathcal{Q}$	Reference quadrangle
$\mathcal{Q}^{PKD}$	Reference quadrangle for the PKD basis definition
$\mathcal{S}$	Reference 1D element
$\mathcal{T}$	Reference triangle
$\mathcal{T}^{PKD}$	Reference triangle for the PKD basis definition
$\mathcal{T}_e$	Reference tetrahedron
$\mathcal{T}_e^{PKD}$	Reference tetrahedron for the PKD basis definition

Spaces

H(div)	$\{u \in \left(L^2(K)\right)^d,\ \nabla \cdot\ u \in L^2(K)\}$
$L^2(K)$	Hilbert space of square integrable function defined on K
$\bar{\mathbb{P}}_p$	Space of piecewise polynomials of degree exactly equal to p
$\mathbb{P}_p$	Space of piecewise polynomials of degree $\leq p$
RT_p	Raviart-Thomas space for a degree p

Solution variables

u	Solution in the physical domain
$\hat{u}$	Solution in the reference domain
$\hat{u}_h$	Polynomial approximation of the solution
$\bar{u}$	Modal basis coefficients in the solution approximation
$\widetilde{u}$	Amplitude of the j-th harmonic of the solution

Other Latin Symbols

$\mathbf{c}$	The velocity vector in the advection equation, $\mathbf{c} = (c_x, c_y, c_z)$ in 3D
C	Airfoil chord
d	The problem dimension

E	Total energy
E_c	Turbulent kinetic energy
I	Imaginary unit
J	Jacobian
$\mathbf{k}$	Wavenumber vector
K	Bounded open subset of $\mathbb{R}^d$ with a Lipshitz continuous boundary
k	Wavenumber of the harmonic wave
L	Lenght of the computational domain
L_2	L_2 error norm
l_i	The one-dimensional Lagrange polynomial associated with point i
M	Mach number
M_i	Shape functions associated with shape point i
$\hat{\mathbf{n}}_k$	Outward normal vector associated with point k in the reference domain
$\mathbf{n}_k$	Outward normal vector associated with point k in the physical domain
N	The number of elements in the mesh
n	Number of vertices of an element
P	Pressure
p	The polynomial order of the approximated solution
$P_n^{\alpha,\beta}$	The corresponding n-th order Jacobi polynomials on the interval $[-1,1]$ which, under the Jacobi weight $(1-x)^\alpha(1+x)^\beta$ are orthogonal polynomials
Re	Reynolds number
R_{gas}	Specific gas constant
T	Temperature
t	Time
t_f	Final time of the computation
(U,V,W)	Velocity components in the x,y,z direction

Other Greek Symbols

α	Angle of attack
Δt	Time step
δ	Kronecker delta function
$\partial\Omega_i$	The i-th element boundary
Γ	Gamma function
γ	Constant ratio of specific heats
κ	Grid frequency, $\kappa = k\Delta x$

$\lambda_{\mathbf{M}}$	Eigenvalues of the spatial operator matrix $\mathbf{M}$
μ	Skew angle
μ_d	Dynamic viscosity
$\hat{\nabla}$	Differential operator in the reference space, $\hat{\nabla} = \left(\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta}, \frac{\partial}{\partial \zeta}\right)$
∇	Differential operator in physical space, $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$
ν	CFL number
$\bar{\omega}$	Dimensionless negative vorticity
Ω	The computational domain
Ω_i	The i-th element in computational domain
ω_j	Quadrature weights
ρ	Density
$\rho_{\mathbf{M}}$	Spectrum of the spatial operator matrix $\mathbf{M}$
τ_ω	Skin (or wall) shear stress
θ	The advection angle in 2D
θ_1, θ_2	The advection angle parameters in 3D
ϑ	Orientation angle of the harmonic wave in 2D
ϑ_1, ϑ_2	Orientation angle of the harmonic wave in 3D

Abbreviations

CFD	Computational Fluid Dynamics
CFL	Courant–Friedrichs–Lewy condition
CGNS	CFD General Notation System
CPR	Correction Procedure for Reconstruction
CPU	Central Processing Unit
DFR	Direct Flux Reconstruction
DGSEM	Discontinuous Galerkin Spectral Element Method
DG	Discontinuous Galerkin
DNS	Direct Numerical Simulation
DoF	Degrees of Freedom
ENO	Essentially Non-Oscillatory
FD	Finite Difference
FE	Finite Element
FNC	Fixed Number of Cells
FP	Flux Points

FR	Flux Reconstruction
FV	Finite Volume
HPC	High Performance Computing
JAGUAR	proJect of an Aerodynamic solver using General Unstructured grids And high-ordeR schemes
LES	Large Eddy Simulation
NCO	Newton-Cotes Open
PDEs	Partial Differential Equations
PKD	Proriol-Koornwinder-Dubiner
RANS	Reynolds-averaged Navier–Stokes
RK	Runge-Kutta
RT	Raviart-Thomas
$SDRT^{OPT}$	SDRT implementation based on interior flux points located at optimization points
$SDRT^{WSJ}$	SDRT implementation based on interior flux points located at Williams-Shunn-Jameson quadrature points
SDRT	Spectral Difference method using Raviart-Thomas elements
SD	Spectral Difference
SP	Solution Points
SSP	Strong Stability Preserving
SV	Spectral Volume
TDIBC	Time Domain Impedance Boundary Condition
TVD	Total Variation Diminushing
VKI	Von Karman Institute
WENO	Weighted Essentially Non-Oscillatory
WSJ	Williams-Shunn-Jameson

Résumé Étendu en Français

Contexte et Objectifs

La méthode des Différences Spectrales (SD) fait partie des méthodes spectrales discontinues d'ordre élevé basée sur la forme forte des équations. Cette méthode a été initialement introduite sous le nom de méthode multi-domaine échelonnée de Tchebychev et appliquée à des maillages composés de quadrilatères en utilisant les propriétés de produit tensoriel. Plus tard, la méthode a été adaptée aux cellules de type simplex et renommée méthode des Différences Spectrales. La méthode a ensuite été étendue aux équations d'Euler et de Navier-Stokes pour des maillages composés de quadrilatères puis de triangles. Le principe de la méthode SD consiste à définir deux polynômes : un polynôme de degré p pour la solution et un polynôme de degré $(p+1)$ pour chaque composante du flux, permettant d'obtenir un ordre de précision $(p+1)$. Ce choix de degrés polynomiaux assure la consistance de la formulation. Deux jeux de points distincts, les points solution (SP) et les points flux (FP) permettent de définir les polynômes d'interpolation de Lagrange de degré différents. La stabilité linéaire de la méthode SD en 1D a été démontrée pour tous les ordres de précision à condition que les FP intérieurs soient placés aux zéros des polynômes de Legendre correspondants. Il a également été montré que la position des SP n'influence pas la stabilité. Ces conclusions s'étendent directement aux quadrilatères et aux hexaèdres puisque la méthode SD sur ces cellules est formulée comme le produit tensoriel de formulations 1D.

L'importance d'utiliser une discrétisation d'ordre élevé pour les calculs LES est soulignée dans le chapitre 1. Une revue bibliographique montre également les avantages et les inconvénients des méthodes discontinues d'ordre élevé existantes. Parmi ces méthodes, l'approche standard SD a été choisie pour être implémentée dans le solveur d'ordre élevé JAGUAR pour sa précision et son efficacité pour la Simulation aux Grandes Echelles. Le solveur JAGUAR est un code en copropriété entre l'ONERA et le CERFACS et vise à traiter des applications impliquant des écoulements instationnaires comme la combustion, la transition vers la turbulence, la capture de chocs et la rentrée atmosphérique. Le traitement des quadrilatères et des hexaèdres a été mis en œuvre avec succès en suivant les règles de produit tensoriel.

Lorsque la méthode SD standard est appliquée aux autres éléments fréquemment utilisés dans les maillages (triangles en 2D ; tétraèdres, prismes ou pyramides en 3D), les études concernant la stabilité n'ont pas été concluantes. En 2D, la stabilité de la méthode SD standard n'est pas assurée pour les cellules triangulaires pour un ordre de précision strictement supérieur à deux. Cette limitation est dépassée en utilisant une formulation alternative des SD : la méthode des Différences Spectrales basée sur les éléments de Raviart-Thomas (SDRT). La différence principale entre le schéma SD standard et le schéma SDRT concerne l'approximation polynomiale du flux. La formulation SD standard utilise une base polynomial composée de scalaires et les coefficients à déterminer sont des vecteurs. A l'inverse, la formulation SDRT utilise la base polynomiale de Raviart-Thomas, composée de vecteurs, et les coefficients à déterminer sont des scalaires. La formulation SDRT sur triangles a été démontrée comme étant linéairement stable jusqu'au quatrième ordre dans le cadre d'une analyse de stabilité de Fourier et validée pour les équations d'Euler et de Navier-Stokes pour des maillages triangulaires et hybrides. Enfin, pour les maillages 3D, l'implémentation SDRT sur les tétraèdres n'a pas encore été formulée. Les formulations des approches SD standard sur cellules de type produit-tensoriel et

SDRT sur triangles sont présentées en détail dans le chapitre 2. L'extension des SDRT aux tétraèdres ainsi que la formulation sur éléments prismatiques (obtenue en utilisant un produit tensoriel entre les schémas standard 1D SD et 2D SDRT) y sont également développées. La complexité induite par le traitement des pyramides est enfin abordée et il est choisi de se concentrer sur les triangles et les tétraèdres. Ce choix est motivé par le fait que les cellules de type simplex (triangles et tétraèdres) offrent la possibilité de traiter des géométries complexes.

Dans ce contexte, cette thèse porte sur l'extension de la capacité du solveur JAGUAR à traiter les maillages composés de cellules de type simplex et les maillages hybrides tout en conservant des propriétés stables et robustes. Pour cela, il est d'abord important de proposer une formulation stable de la méthode SD sur des cellules simplex. La première étape consiste à étendre l'approche SDRT pour des ordres de précision supérieurs à quatre pour les cas 2D. Ensuite, la formulation SDRT sur tétraèdres doit être établie. La stabilité linéaire de la discrétisation SDRT spatiale couplée au schéma temporel doit être soigneusement analysée pour déterminer des limites de stabilité fiables. Enfin, l'implémentation de la formulation SDRT dans le JAGUAR pour les maillages triangulaires, hybrides 2D et tétraédriques permettra d'étudier les propriétés du schéma proposé pour la simulation d'écoulement.

Schéma SDRT pour triangles basé sur des règles de quadrature existantes

Pour déterminer des formulations SDRT stables pour des ordres de précision supérieurs à quatre en 2D, la stabilité linéaire du schéma SDRT sur triangle a été d'abord été étudiée dans le chapitre 3. D'après les premiers résultats publiés concernant la méthode SDRT sur triangles, le paramètre qui semble avoir le plus d'impact sur la stabilité est la position des FP intérieurs. Jusqu'au quatrième ordre, on peut noter que les formulations stables ont été obtenues en choisissant de placer les FP intérieurs suivant certaines règles de quadratures déjà publiées. Les FP de bord sont eux placés aux points de Gauss-Lobatto, de façon à correspondre exactement aux FP de bord des quadrilatères et ainsi de faciliter l'implémentation hybride.

Dans un premier temps, il a donc été logique d'étudier toutes les règles de quadrature appropriées pour des ordres supérieurs à 4, l'objectif étant d'étudier plusieurs implémentations de SDRT en fonction du choix de l'emplacement des FP intérieur. L'analyse de Fourier a été utilisée pour retrouver les résultats de stabilité des SDRT pour $p \in [\![1, 3]\!]$ puis pour déterminer la stabilité de chaque implémentation (i.e. de chaque placement de FP intérieurs) pour $p > 3$. Toutes les implémentations ont conduit à de petites valeurs positives de parties réelles dans les spectres de l'opérateur spatial pour $p > 3$. Il a cependant été noté que la règle de quadrature valable pour tous les degrés d'approximation polynomiale qui permet d'obtenir les valeurs les plus faibles et celle de Williams-Shunn-Jameson (WSJ). Cette règle de quadrature sera donc privilégiée dans la suite de l'étude.

Le chapitre 4 a étudié la possibilité de stabiliser la méthode SDRT basée sur les FP intérieurs placés aux points de la règle de quadrature de Williams-Shunn-Jameson (SDRT^{WSJ}) sur des triangles pour $p > 3$ en utilisant la discrétisation temporelle pour dissiper les petites valeurs positives de la partie réelle dans le spectre de l'opérateur spatial. Deux options menant à des discrétisations spatio-temporelles couplées stables ont été déterminées. La première utilise le schéma temporel Runge-Kutta quatre étapes de Jameson (RK4J) pour dissiper l'instabilité spatiale et a été validée pour tous les angles d'advection. Les coefficients de ce schéma ont été choisis de façon à amortir les modes à haute fréquence et permet donc de stabiliser le schéma spatial. Cependant, une erreur de dissipation numérique élevée a été constatée. La seconde utilise des schémas temporels SSP de troisième ordre, qui

sont associés à de meilleures propriétés de dissipation et de dispersion. Cependant, la stabilité de la discrétisation spatio-temporelle couplée est soumise à certaines conditions : pour SDRT_4, le nombre CFL doit se situer dans un intervalle donné, tandis que pour SDRT_5, la stabilité n'est pas assurée pour tous les angles d'advection. La forte dépendance à l'angle d'advection ou à une double condition sur le nombre CFL rend cette seconde possibilité difficilement utilisable en pratique. Enfin, l'utilisation d'autres quadratures comme FP intérieurs a été étudié mais les résultats montrent que la quadrature WSJ reste la plus adaptée. Dans ce chapitre, l'importance de conduire une analyse de stabilité sur la discrétisation spatio-temporelle couplée a été soulignée puisque des schémas SDRT spatialement stables ($p = 2$ et $p = 3$) ont été montrés comme menant à une discrétisation spatio-temporelle instable lorsqu'ils sont associés à des schémas SSP temporels d'ordre deux pour un CFL supérieur à $0,02$. Pour les schéma temporels compatibles, les limites de stabilité CFL ont été indiquées précisément.

Dans le chapitre 6, la précision du schéma SDRT^{WSJ} stabilité par le schéma temporel RK4J pour $p \in [\![4, 5]\!]$ a été étudiée pour les équations différentielles partielles du premier ordre. La simulation de l'advection d'un sinus dans un domaine périodique a permis d'observer des erreurs de dissipation élevées, entièrement pilotées par le schéma temporel RK4J. Cela confirme que cette solution ne permet pas d'obtenir une précision d'ordre élevé pour les problèmes instationnaires sans imposer un pas de temps très faible et donc augmenter drastiquement le coût de calcul. Pour $p < 4$, le schéma SDRT^{WSJ} associé à un schéma temporel avec de bonnes propriétés de dissipation et de dispersion permet bien sûr de retrouver les ordres de précision $p + 1$ attendus. L'association du schéma spatial SDRT^{WSJ} avec le schéma temporal RK4J reste cependant possible pour les problèmes stationnaires. Dans le chapitre 7, cette solution a été utilisée pour simuler l'écoulement visqueux et stationnaire autour d'un cylindre à nombre de Reynolds $\text{Re} = 20$ sur maillage hybride 2D. Les résultats ont conduit à un bon accord avec les données de référence.

Schéma SDRT pour triangles basé sur un processus d'optimisation

Pour dépasser les conditions restrictives associées à la solution proposée dans le chapitre 4, il a été envisagé de faire varier le placement des FP intérieurs à travers un processus d'optimisation afin de déterminer des formulations SDRT spatialement stables. Ce processus d'optimisation a été utilisé avec deux méthodes d'analyse de stabilité : l'analyse de Fourier standard et une nouvelle méthode, introduite dans le chapitre 3 et visant à réduire le temps d'analyse et ainsi augmenter le nombre d'implémentations étudiées. Cette nouvelle méthode de stabilité a été proposée comme moyen efficace d'évaluer la stabilité d'un schéma à partir d'un nombre fixe de cellules, sans injecter de solution harmonique comme solution initiale. Il a cependant été démontré que les résultats obtenus sur un domaine composé d'un nombre fixe de cellules ne pouvaient pas être étendus dans un cadre général. Cette approche a donc été mise de côté au profit de l'analyse de stabilité de Fourier.

Le processus d'optimisation a été utilisé de nouveau pour déterminer des FP intérieurs permettant d'obtenir des formulations stables dans le chapitre 5, associé cette fois ci à l'analyse de stabilité de Fourier. De nouveaux ensembles de FP intérieurs conduisant à des formulations SDRT spatialement stables (schémas dénotés SDRT^{OPT}) ont pu être déterminés pour $p = 4$ et $p = 5$. La preuve de la stabilité spatiale de cette formulation a été donnée par une étude du spectre de l'opérateur spatial et la discrétisation spatio-temporelle couplée a ensuite été étudiée pour déterminer les limites de stabilité CFL. Le processus d'optimisation a également été mené pour $p = 6$ sur les triangles. Une formulation stable a été déterminée pour un angle d'advection $\theta = 0$ mais n'a pour l'instant pas pu être généralisée pour tous les angles d'advection.

La précision de l'implémentation SDRT^{OPT} a été vérifiée dans le chapitre 6 pour les équations différentielles partielles du premier ordre. Pour chaque degré d'approximation polynomiale $p \in [\![4, 5]\!]$,

la précision d'ordre élevé $p+1$ est retrouvée sur maillage triangulaire et hybride. Le schéma SDRT^{OPT} est ensuite utilisé dans le chapitre 7 pour la simulation de fluides visqueux régis par les équations de Navier-Stokes. Pour $p \in [\![4, 5]\!]$, l'implémentation SDRT^{OPT} est utilisée, tandis que pour $p \in [\![2, 3]\!]$, l'implémentation SDRT^{WSJ} est conservée puisque démontrée comme étant spatialement stable. Un écoulement autour d'une aile NACA0012 sur un maillage triangulaire quadratique est simulée pour trois conditions d'écoulement différentes et tous les schémas conduisent à un bon accord avec les données de référence répertoriée par la NASA.

Schéma SDRT pour tétraèdres

Après avoir établi une formulation SDRT stable sur triangles jusqu'au sixième ordre de précision, la stabilité du schéma SDRT sur tétraèdres a été étudiée à l'aide de l'analyse de Fourier (chapitre 3). Pour le placement des FP, les FP de bord ont d'abord été choisi comme correspondant aux SP du triangle (ici choisi comme correspondant à la quadrature de WSJ), de façon à ce que les FP de bord d'un tétraèdre et d'un prisme soient localisés exactement au même endroit pour simplifier l'implémentation hybride. Plusieurs localisations de FP intérieurs ont été étudiées, basées sur des règles de quadrature existantes pour les tétraèdres. La stabilité linéaire de la méthode SDRT a été établie sur tétraèdres jusqu'au troisième ordre de précision ($p = 2$) en choisissant la position des FP intérieurs comme correspondant aux points de quadrature de Shunn-Ham. Pour $p = 3$, aucune des quadratures existantes n'a mené à un schéma SDRT stable.

Le processus d'optimisation a ensuite été mené pour $p = 3$ sur les tétraèdres dans le chapitre 5 mais n'a pas conduit à des résultats concluants. Cependant, le problème d'optimisation a considéré en tant que paramètre d'optimisation uniquement la position des FP intérieurs, alors que les FP situés sur les faces étaient fixés aux points de quadrature de WSJ. Il a ensuite été démontré que la position des FP situés sur les faces a un impact important sur la stabilité du schéma. Un problème d'optimisation considérant à la fois la position des FP intérieurs et la position FP situés sur les faces comme paramètres d'optimisation pourrait mener à la détermination de formulations SDRT stables pour $p > 2$.

La stabilité de la discrétisation spatio-temporelle sur tétraèdre a été vérifiée dans le chapitre 4 pour $p \in [\![1, 2]\!]$. Tous les schémas temporels étudiés ont conduit à des résultats stables lorsqu'ils étaient associés au schéma SDRT. Les limite de stabilité CFL ont été données pour chaque schéma temporel.

L'ordre de précision du schéma SDRT sur tétraèdres a été vérifié dans le chapitre 6 à l'aide d'un cas test Euler 3D. Pour $p = 1$ et $p = 2$, l'ordre de précision $p + 1$ attendu est retrouvé. Enfin, les résultats de la simulation du cas du Taylor-Green Vortex (TGV) sur des maillages composé de tétraèdres sont présentés dans le chapitre 7. Un bon accord avec les données de référence et un coût de calcul cohérent permettent de valider l'implémentation du SDRT 3D.

Chapter 1

Introduction

This chapter aims to present the thesis context. First, an overview of Computational Fluid Dynamics is given, introducing the notions of mesh, numerical method, and computational efficiency. The concept of turbulence and approaches to simulate turbulent complex flows are then presented. The need for accuracy, mesh flexibility and computational efficiency for turbulence simulations leads to consider high-order discontinuous methods. A review of high-order discontinuous methods is provided. From this state of the art, the thesis motivations can be stated.

Overview

1.1. Computational Fluid Dynamics . . . **2**
1.1.1. Mesh Requirements . . . 2
1.1.2. Numerical Methods . . . 3
1.1.3. Computational Efficiency . . . 4
1.2. Turbulence Modeling and Computation . . . **5**
1.2.1. Direct Numerical Simulation . . . 6
1.2.2. Reynolds-Averaged Navier–Stokes Approach . . . 6
1.2.3. Large Eddy Simulation . . . 7
1.3. High-Order Discontinuous Methods for Unstructured Grid Based on the Integral Form . . . **9**
1.3.1. Discontinuous Galerkin Method . . . 9
1.3.2. Spectral Volume Method . . . 10
1.4. High-Order Discontinuous Methods for Unstructured Grid Based on the Strong Form . . . **11**
1.4.1. Flux Reconstruction Method . . . 11
1.4.2. Spectral Difference Method . . . 12
1.5. Motivations for the Thesis . . . **14**
1.6. Outline . . . **14**
1.7. Publications . . . **15**
1.7.1. Papers . . . 15
1.7.2. Presentations . . . 15

1.1. Computational Fluid Dynamics

Fluid Dynamics consists in describing the flow of fluids (liquids and gases). The discipline is based on conservation laws (mass, momentum and energy). Computational Fluid Dynamics (CFD) numerically solves approximations of conservation laws over a given geometry. Compared to carrying experimental tests, CFD offers the possibility to be aware of the flow state at all times on the whole domain. Several geometries can be tested which would be experimentally costly. Using CFD during the design process is faster and less expensive than conducting physical tests on prototypes, therefore it has become an indispensable tool in the aerodynamics industry. A CFD solver is based on a numerical method used to discretize a mathematical model of the physics (e.g. Navier-Stokes equations) on a mesh. In the next sections, the notions of mesh, numerical method and computational efficiency are presented.

1.1.1. Mesh Requirements

To discretize equations, the numerical method relies on a mesh. A mesh (or grid) is a subdivision of the domain into small volumes called cells. Cells are defined by their corners (also called nodes), their edges and their faces (in 3D). Two types of mesh are possible: a structured mesh or an unstructured mesh.

A structured mesh is based on a discretization of the domain in quadrilateral elements (in 2D) or hexahedra (in 3D) called blocks. Any block gives 2 or 3 privileged space directions and the number of discretization points on parallel edges is forced to be kept constant. As a consequence, any node is uniquely identified by two indices (i, j) in 2D or three indices (i, j, k) in 3D. Using a structured mesh allows one to easily spot each node by its indices. Additionally, when considering a certain cell, information about neighboring cells is easily available. However, structured mesh generation is difficult and can take weeks or months for a complex geometry.

On the other hand, an unstructured mesh is based on elements that can be triangles or quadrilaterals in 2D and tetrahedra, prisms, pyramids or hexahedra in 3D. From the list of mesh points, any cell is defined through a subset of the mesh points (8 points for a hexahedron, 4 for a tetrahedron...). The connectivity to build 2D and 3D cells is explicitly required and cells are not ordered as for a structured mesh. The minimum required information for an unstructured grid is:

- the number of nodes,
- the coordinates (x_i, y_i, z_i) of each node i,
- a connectivity table where each cell is defined by its vertices and
- a list of elements located at the domain boundary on which boundary conditions will be applied.

To get one node coordinates for a given cell, one first needs to find this node in the vertex list of the cell and then locate it in the coordinate table. Spotting one node is thus more complicated than with a structured mesh: it needs indirect addressing. However, unstructured meshes are more flexible than structured grids and their generation can be automated easily. Generation of an unstructured mesh is much easier and takes a short time even for a complex geometry (some days at the most) but the local mesh refinement is generally not manufactured as easily as using structured mesh generation tools.

1.1.2. Numerical Methods

Today, most of industrial CFD solvers use a second-order accurate Finite Volume (FV) method to discretize the Reynolds-averaged Navier–Stokes (RANS) equations. The second-order FV scheme is part of what is called low-order numerical methods.

Mathematically, a numerical method is said to be k-th-order accurate (or of order k) if the error between the exact solution and its numerical approximation behaves as h^k, where h is the mesh size. Low-order methods are defined as numerical methods of order below or equal to 2 whereas high-order methods are at least third-order [1]. Low-order methods have been subject to intense research over the past decades and are now robust and efficient. However, these schemes generally induce numerical dissipation and dispersion. The notion of dissipation and dispersion needs to be presented. Let us consider the model problem of the linear advection equation at velocity c in a periodic domain. For a given computational domain of length L ensuring a periodic initial solution, the transportation of a planar wave leads to a periodic solution of period $T = L/c$. In practice, the solution at time kT, $k \in \mathbb{N}$, and the initial solution are not located exactly at the same position and don't have the same amplitude. The phase shift is called dispersion, whereas the loss of energy during the computation is called dissipation (or diffusion). Both are due to time-space discretization errors. Dispersion and dissipation are summarized in Fig. 1.1.

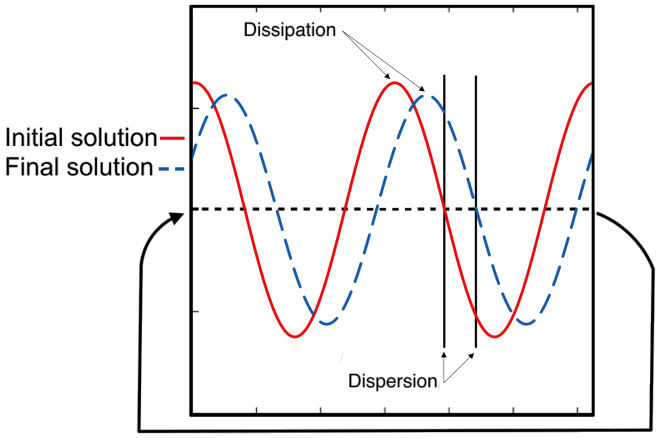


Figure 1.1. – Representation of dissipation and dispersion effects for the linear advection of a planar wave in a periodic domain

A way to circumvent this behavior using a low-order method is to consider a very refined mesh but the number of grid points can become unaffordable. To accurately compute vortical flows at a reasonable cost, one can consider high-order methods. Because of their higher accuracy, high-order methods have better wave propagation properties than low-order methods. As a result, they can solve turbulent vortices accurately with fewer degrees of freedom (DoF) and besides, the total Central Processing Unit (CPU) time for the computation can become even smaller than with low-order schemes (for a given accuracy).

The first type of high-order methods is based on the FV formalism. As mentioned before, the second-order accurate FV scheme is the most commonly used in industrial solvers, but the FV formalism can be extended to higher orders of accuracy. Such methods, like ENO [2–4], WENO [5–7] or k-exact schemes [8–11] consist in a solution representation using a polynomial built from cell-averaged values of the unknowns over the neighboring cells. An alternative approach considers a polynomial approximation defined locally, inside the cell. This is the principle of the family of schemes called high-order spectral discontinuous methods.

1.1.3. Computational Efficiency

A numerical method will be efficient if its formulation stands on a compact stencil. The stencil is the number of neighboring cells used to update the solution in one cell. Standard numerical methods - Finite Element (FE), Finite Difference (FD), Finite Volume (FV) - can easily reach high-order accuracy by extending the stencil. For FD and FV methods based on a structured mesh, extending the stencil is particularly simple since information about neighboring cells is immediately available. However, the same situation on an unstructured grid can become extremely difficult to handle.

The large stencil must be built by hand, using the direct neighboring cells. The number of unknowns to define the scheme is known *a priori* and must be recovered on the stencil. In an unstructured mesh, the number of neighboring cells is not constant and the definition of the stencil is not straightforward. Additionally, a large stencil can be defined on several blocks, which leads to many exchanges between CPU to perform at any iteration to deal with parallel computation. To sum up, being based on a compact stencil is crucial for a numerical method to reach good computational efficiency in a massively parallel environment.

In this section, main notions associated with CFD simulations were introduced: mesh, numerical method and computational efficiency. Benefits of using an unstructured mesh were underlined: an unstructured grid is more flexible to treat complex geometries and its generation is way faster. Dissipation and dispersion properties of a numerical method were defined. High-order methods lead to a better accuracy than low-order methods but can be difficult to parallelize if they are not based on a compact stencil. It was pointed out that for a CFD solver to be computationally efficient, the chosen numerical method should be based on a compact stencil. The class of high-order discontinuous methods fulfills this requirement since they are based on local polynomial approximations. They were briefly mentioned and will be detailed later, but first, the concept of turbulence and the different approaches to deal with it are presented.

1.2. Turbulence Modeling and Computation

Industrial applications are generally associated with high Reynolds number flows, for which the relatively low importance of diffusion effects compared with nonlinear transport can lead to the transition to turbulence, one of the most complex physical phenomena to account for when studying fluid dynamics. Turbulence is a natural, chaotic and three-dimensional phenomenon driven by energy transfers occurring in the fluid. Turbulence results from the superposition of vortices (also called structures or eddies) of different energy distributed over a large range of temporal and spatial scales [12]. In a turbulent viscous flow, large vortices are found at small wavenumbers and are carrying energy. Those large structures break into smaller and smaller structures, without energy production or dissipation. At high wavenumbers, turbulence structures are small enough to be dissipated by viscosity effects. This process is known as the Kolmogorov energy cascade [13] and is described in Fig. 1.2. Three main zones can be distinguished. The production zone is located at small wavenumbers κ and composed of large turbulent structures containing the main part of the turbulent kinetic energy E_c. The inertial area contains medium turbulent scales and the kinetic turbulent energy follows Kolmogorov theory: as the wavenumber increases, the slope is in $\kappa^{-5/3}$. Finally, the dissipation zone is composed of small eddies which kinetic turbulent energy is dissipated in heat transfer. In the turbulent spectrum, the energy transfer is from large structures (small wavenumbers) to small structures (high wavenumbers), i.e. from the production zone to the dissipation zone.

In CFD, turbulence structures can be either computed (simulated, solved) or modeled. Today, there are essentially three classes of CFD methods to deal with turbulence, represented in Fig. 1.2:

- Direct Numerical Simulation (DNS): all turbulent scales are *computed*,
- Reynolds-averaged Navier–Stokes (RANS): all turbulent scales are *modeled*,
- Large Eddy Simulation (LES): large turbulent scales are *computed* whereas small scales are *modeled*.

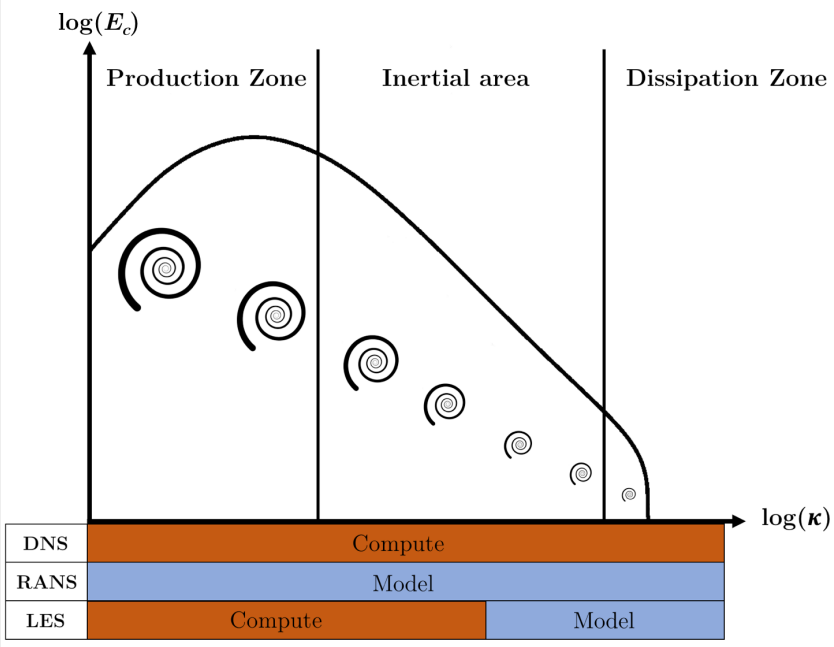


Figure 1.2. – Simplified view of the Kolmogorov spectrum and illustration of CFD methods for turbulence

1.2.1. Direct Numerical Simulation

The first method, called **Direct Numerical Simulation** (DNS), consists in solving the Navier-Stokes equations without modeling (except for the temporal and spatial discretizations), implying that every temporal and spatial turbulence scales have to be computed. The instantaneous fluid information can thus be obtained at any time and any location and leads to a reliable and accurate description of turbulence. To properly capture all turbulent scales:

- The mesh should be extremely refined (same size order than small turbulent structures located in the dissipation area);
- Numerical errors should be minimized by using a small time step and a high-order scheme with optimized dissipation and dispersion properties (capability to transport a wave without damping it nor having a phase shift).

These requirements lead to a huge computational cost that increases with the Reynolds number Re: the number of grid points for the computation varies as $\mathrm{Re}^{9/4}$ [14]. Today, DNS seems out of reach for most industrial applications, which deal with flows associated with high Reynolds numbers (typically higher than 10^7). However, instantaneous fluid information obtained from DNS computations at low Reynolds numbers help to better understand turbulence mechanisms and can be used to validate RANS and LES models.

1.2.2. Reynolds-Averaged Navier–Stokes Approach

The second technique is based on taking into account the mean effect of turbulence on the flow and deals with the **Reynolds-averaged Navier–Stokes** (RANS) equations. Starting from the Navier-Stokes equations, an ensemble-averaging procedure is applied using Reynolds decomposition: a variable is split into a mean and a fluctuating component, leading to the RANS equations. Using RANS equations, the mean effect of turbulence on the flow is considered and turbulent fluctuations are modeled. RANS equations have the same form as the Navier-Stokes equations, except for additional terms in the heat flux and the Reynolds stress tensor. The Reynolds stress tensor comes from the Navier-Stokes nonlinearity and represents turbulence effects on averaged variables. The Reynolds stress tensor introduces 6 new variables, leading to 11 unknowns for 5 equations (in 3D). It is then mandatory to close the system. To do so, many closures were proposed, all of them aiming at modeling the Reynolds stress tensor. They can be classified into first-order models and second-order models.

The latter consists in introducing one transport equation for each Reynolds stress tensor component (e.g. Reynolds Stress Modeling models) and induces a high computational cost. On the other hand, first-order models rely on the Boussinesq hypothesis [15] and the turbulent viscosity concept, which links the Reynolds stress tensor to the average flow. For example, the class of $k-\epsilon$ turbulence models [16–18], that is probably the most commonly used in industry, determines the turbulent viscosity from the turbulent kinetic energy and the dissipation rate, both of them determined from their transport equations. Many other turbulence models have been proposed, based on one [19], two [20] or more transport equations. However, none of them is considered as appropriate for all kinds of flow.

Since the RANS approach does not compute all turbulent scales but model them, requirements about the mesh and the spatial-temporal numerical schemes are lighter, leading to a reasonable computational cost. Contrary to approaches solving turbulent fluctuations, the RANS approach admits steady solutions, thus it has been massively used for steady problems in the industry. The RANS approach is nowadays the most common technique for industrial applications and RANS equations are generally solved using a second-order numerical scheme. However, since turbulent fluctuations are not computed but modeled, the model capability to reproduced complex phenomena can be questioned.

As mentioned before, there is no universal turbulence model and none of them can accurately represent the mean effects of turbulence on the flow. For aeroacoustics, where turbulent structures interact between themselves, or combustion simulations, where interactions between chemical and turbulent scales take place, none of the RANS models can give accurate results.

1.2.3. Large Eddy Simulation

The last alternative is called **Large Eddy Simulation** (LES) [21]. The approach consists in computing the largest turbulent scales and modeling the smallest ones, which only contain a small fraction of turbulent energy. To do so, the Navier-Stokes equations are spatially filtered. The filter size corresponds to the mesh cells size, defined according to the grid resolution that can be afforded. Turbulent scales larger than the filter size are captured by the mesh whereas smaller turbulent scales are removed and modeled using a subgrid-scale model [22]. The use of a model for small turbulent scales is justified by the universal nature of the smallest scales (Kolmogorov hypothesis [13]). In the filtered Navier-Stokes equations, the Reynolds stress tensor, which describes the interaction between resolved and unresolved structures, is modeled using subgrid-scale eddy-viscosity. A first model of the subgrid-scale eddy-viscosity was presented by Smagorinsky [23], followed by the development of different other models (static, transport, dynamic) [24–27].

Compare to DNS, LES allows the use of a coarser grid since small scales are modeled: in 3D, the number of grid points is approximately $\mathrm{Re}^{3/2}$, leading to a lower computational cost. However, LES computations remain too expensive to be routinely used in the industrial design process. Compared to RANS, LES offers the possibility to capture unsteady phenomena (vortex shedding, acoustic waves) and thus to better represent turbulent flows. For combustion, the initial form of the Navier-Stokes equations is extended to account for several reactive species with chemical reactions. Among the possible techniques, LES has shown its superior accuracy to model chemistry-turbulence interactions compared to the RANS approach [28–32].

LES requires accurate schemes with optimized properties to correctly capture the largest turbulence scales. The chosen numerical scheme should have low-dissipation and low-dispersion properties to capture the vortex-dominated flow while being able to deal with an unstructured mesh to cope with any complex geometry. Additionally, it is also required to maintain a good parallel efficiency. The common class of high-order methods based on the FV formalism deals with polynomial reconstruction using cell-centered quantities. Since the polynomial is constructed based on information from neighboring cells, the number of cells needed increases with the degree of accuracy, making the method difficult to parallelize.

To handle polynomials more efficiently, high-order discontinuous methods were introduced. Rather than using a cell and its neighbors to construct the polynomial, these methods are based on a polynomial reconstruction of the solution from data directly available in the considered cell. Consequently, the stencil is reduced to the considered cell and its immediate neighbors, leading to a compact stencil and making the method much easier to parallelize. Direct neighbors are required for the flux computation since they allowed information to transit from one cell to another. The reconstruction of a high-order polynomial in each cell is possible if there are enough information available, thus the local amount of DoF increases with the order of accuracy.

LES appears to be the most promising approach to obtain an accurate representation of a turbulent complex flow with a practicable computational cost. To compute large turbulent scales while keeping a reasonable mesh size, high-order numerical methods are considered. The compact stencil properties of high-order discontinuous methods make them particularly suitable for high-order accuracy and computational efficiency. Among high-order discontinuous methods, two classes can be identified: methods based on an integral formalism, as Discontinuous Galerkin or Spectral Volume, and methods using a strong formalism, as Spectral Difference or Flux Reconstruction. They are presented in detail in the next sections.

1.3. High-Order Discontinuous Methods for Unstructured Grid Based on the Integral Form

1.3.1. Discontinuous Galerkin Method

The most popular approach, the Discontinuous Galerkin (DG) method, has been successfully implemented in many solvers and lead to a very rich research. Without being exhaustive, a partial literature review focused on CFD is available in several books [33–40] and many contributions in Europe also come from projects [41–43] involving research centers and industry [44]. The first DG method was introduced in 1973 by Reed and Hill [45] for a linear hyperbolic equation: the neutron transport equation. It was then adapted for nonlinear hyperbolic conservation laws in a series of papers by Cockburn *et al.* [46–49]. The DG method links the standard FE method and the FV method: unknowns defined on a polynomial basis are solutions of a weak problem, as in FE, but discontinuities at mesh interfaces are solved using an approximated Riemann solver, as in FV.

The principle is to find a polynomial representation of the solution that satisfies a variational form of the governing equations within each element. For a $(p+1)$-th-order accurate DG scheme, variables are approximated with a polynomial of degree p in each cell. The DG unknowns are the polynomial coefficients. At cell interfaces, the proper flux polynomial degree needs to be ensured for the approach to be conservative and consistent. Flux values are thus uniquely defined at each cell interface quadrature points as the solution of a Riemann problem. In the literature, there are many possible DG approaches - modal [50–52], nodal [34, 53, 54], DG Spectral Element Method (DGSEM) [55–58] - and all depend on the chosen set of basis functions. Basis functions choice will have an impact on the stability, accuracy and efficiency of the DG method.

Like FV based methods, the DG method is fully conservative due to the Riemann solver used to compute numerical fluxes across element boundaries: since the flux polynomial is uniquely defined and continuous on the whole computational domain, conservation is ensured. In terms of stability, the DG method was proven stable for any order of accuracy through a cell entropy inequality [59] which holds for all scalar nonlinear conservation laws. A major asset compare to FV-based methods is the DG accuracy property: the method can be easily designed for any order of accuracy without increasing the stencil. Since the stencil is compact (the considered cell and its direct neighbors), the DG approach is suitable for High-Performance Computing (HPC). Even if more than one piece of information needs to be stored and evolved for each cell (the polynomial coefficients), meaning that the memory requirement is higher than FV methods, there is no reconstruction procedure involved since needed values to evaluate the flux are immediately available. Moreover, the DG method can be applied to unstructured meshes composed of any standard element (hexahedron, tetrahedron, prism, pyramid), which allows one to treat complex geometries. The DG method is also compatible with h and p adaptation: the solution accuracy is enhanced by playing with the local mesh refinement (h-adaptation) and solution accuracy (p-adaptation). A lot of work was performed on this subject and was presented during a dedicated session at the Von Karman Institute (VKI) [42].

However, the DG method has at least two disadvantages. First, the use of the integral formulation leads to an expensive computational cost to obtain a high-order of accuracy since high-order surface and volume integral evaluations using quadrature rules are required. Secondly, the modal and nodal DG methods have a severe Courant–Friedrichs–Lewy (CFL) stability constraint: compared to an FV method, the time step is divided by $(2p+1)$ due to the mass matrix. Thus the higher the order of accuracy is, the more stringent the CFL condition becomes. That can be a curb for unsteady computations. The situation is overcome with the DGSEM for which, compared to an FV method, the time step is divided by $(p+1)$ only, but the multi-element shape capability is not trivial [60, 61].

1.3.2. Spectral Volume Method

The Spectral Volume (SV) method was first developed in 2002 by Z.J. Wang in a series of papers [62–68]. Research has been conducted about the SV application to Euler equations [69, 70], implementation efficiency [71], p-multigrid refinement [72] and discretization of spatial derivatives [73, 74].

As discussed in section1.1, high-order methods based on an FV formalism consists in defining a local polynomial approximation of the state variables on a considered cell by using a stencil that contains the cell and a sufficient number of neighboring cells. The SV method also uses averages over control volumes as solution variables but unlike FV methods, control volumes are not whole cells. Control volumes are obtained by subdividing each cell (also called spectral volume) into smaller volumes called sub-volumes. The number of sub-volumes required is as large as the number of unknowns in the polynomial approximation. Thus, the reconstruction process does not require information from neighboring cells, which leads to a compact stencil, suitable for HPC. The method's order of accuracy is then defined by the number of sub-volumes in each cell. Cell-averaged data from these sub-volumes are used to reconstruct a high-order approximation in the spectral volume. Inside a spectral volume, state variables are continuous across sub-volumes boundaries thus analytic fluxes can be used rather than a Riemann solver, which is not necessary. Riemann solvers are only used to compute fluxes at the spectral volume interfaces. As for the DG method, the SV method maintains a high-order of accuracy using a high-order quadrature for flux computation (at spectral volumes boundaries and interior sub-volume boundaries).

However, when considering 3D problems, the SV method becomes highly expensive. This is due to the number of quadrature points needed to compute surface integrals, which can reach hundreds of thousand points. The difficulty does not lie in the definition of the locations (everything can be simplified using the reference element) but in the evaluation of the data from the polynomial reconstruction. This default is partially circumvented using a quadrature-free implementation of the SV method, as proposed by Harris *et al.* [71, 75]. Additionally to the high quadrature computation price, for orders of accuracy higher than two, the partitioning of a spectral volume into sub-volumes is not unique and has a considerable influence on the stability of the method. This issue was investigated by Van den Abeele *et al.* in [76] where they identified a weak instability in several SV partitions and proposed stable ones. It can be noticed that even if a partition leads to a stable scheme, it does not necessarily preserve the scheme order of accuracy. To maintain the order of accuracy, it was demonstrated that two criteria have to be respected. First, the number of sub-volumes at a spectral volume interfaces must be large enough to be exact for polynomials using dedicated quadrature rules. A minimum number of control volumes per spectral volume then needs to be ensured. However, these rules are insufficient to uniquely define spectral volume partitions for an order of accuracy higher than two. Work was also performed on the analysis of dissipation and dispersion properties of the SV scheme depending on the definition of the sub-cells [77].

Today, the SV method does not seem to be of strong interest, essentially due to the fact that the sub-cell definition did not lead to best practices. Last studies concerned a time implicit formulation [72], the treatment of the diffusion terms for the Navier-Stokes equations [73, 78], its extension to third-order spatial derivative terms [74], an efficient time integration procedure (implicit formulation) [69, 72] and the choice of a suitable diffusion scheme [73, 78].

The Discontinuous Galerkin and the Spectral Volume methods are both based on the integral form of equations and are thus expensive to use, especially due to the need for quadrature rules. Kopriva and Kolias [79] demonstrated that employing the differential form is also possible by introducing the staggered-grid spectral method, later called the Spectral Difference method. This approach and the Flux Reconstruction technique are presented in the next sections.

1.4. High-Order Discontinuous Methods for Unstructured Grid Based on the Strong Form

In section 1.3 were presented high-order discontinuous spectral methods based on an integral formulation and it was highlighted that quadrature rules for surface and volume integrals were required. Additionally, the standard DG method is associated with a severe CFL condition depending on the order of accuracy of the polynomial reconstruction. To avoid the CFL condition coming from the mass matrix, it is interesting to solve the strong form of equations, playing directly with the differential equations, but taking into account standard additional constraints such as conservation, efficient theoretical background and strong parallel capability. In this section, attention is first paid to the Flux Reconstruction (FR) approach and then to the Spectral Difference (SD) method.

1.4.1. Flux Reconstruction Method

The first class of high-order discontinuous methods based on the strong form is called the Flux Reconstruction (FR) or the Correction Procedure for Reconstruction (CPR) approach. First established in 1D by Huynh in 2007 [80], the method consists in approximating both the solution and the flux as polynomials of degree p. In a second step, a lifting operator defined as a polynomial of degree $(p+1)$ is introduced to correct the flux polynomial of degree p to make the approach conservative and consistent. The lifting operator plays a central role in the properties of the scheme and enables to link the FR method, the DG formulation and other methods [81, 82].

Mathematical properties of the FR method are subject to many papers. The stability of the FR method was studied by Jameson *et al.* in the linear and nonlinear cases [83, 84] and then extended to 2D flows [85]. Spectral properties (dissipation and dispersion) of the FR were analyzed in [86, 87]. In 2011, Vincent, Castonguay and Jameson [88] proposed a new class of lifting operator that makes the formulation energy stable. For this scheme, the naming VCJH scheme (Vincent, Castonguay, Jameson and Huynh) is introduced in [84]. They extended the method for triangles in 2012 [89]. Huynh, Z.J. Wang and Vincent published in 2014 a reference paper on the latest developments for the FR/CPR schemes [90]. The treatment of the diffusion term in the Navier-Stokes was also the subject of research [91, 92]. On a coarse grid, the stability seems to be a problem and leads to intense research [85, 93]. The FR method is today compatible with h and p-adaptation [94] and with the adjoint formulation [95].

Recently, Jameson's team proposed the Direct Flux Reconstruction (DFR) method [96, 97] for 1D flows and tensor product cells. The principle of the DFR lies in the fact that the flux polynomial is required to be of degree at least $(p+1)$ to cope with consistency. In the DFR, the authors propose to build a flux polynomial of degree $(p+2)$ using the flux at solution points and flux points at the same time. A simple Lagrange polynomial is then built and differentiated at solution points to time-match the hyperbolic equations. The procedure was extended by Romero and Jameson [96] to triangles using a specific geometric transform that links a triangle in the physical domain to a quadrilateral in the reference space.

The FR method was applied to the simulation of 3D non linear systems including a turbulent flow over a cylinder [98], an SD7003 infinite wing at 4 degrees angle of attack [99, 100], supersonic jet simulations [101] and flows over turbomachinery [102] as well as the Large Eddy Simulation of a turbulent channel flow [103]. Reviews of recent FR/CPR developments were provided by Z.J. Wang and Huynh [104] and by Witherden, Vincent and Jameson [105].

The FR method has gained a lot of attention during last years due to its capability to handle hybrid grids in 2D and 3D. Many authors also argue that the FR method can recover other standard

discontinuous spectral methods but it needs to be underlined that this was demonstrated in the linear case only [80]. Finally, the FR scheme strongly depends on the lifting operator that is built artificially: the number of constraints is not large enough to control all polynomial coefficients.

1.4.2. Spectral Difference Method

The second class of high-order discontinuous spectral method dealing with the strong form of equations is called the Spectral Difference (SD) method. Initially named the staggered-grid Chebyshev multi-domain method, the approach was initiated by Kopriva and Kolias [79] in 1996 and applied to structured quadrilateral grids using a tensor-product framework by Kopriva in [106]. In 2006, Liu *et al.* [107] proposed an extension of Kopriva and Kolias' work to simplex cells and called the approach the Spectral Difference method. Z.J. Wang *et al.* [108] adapted the procedure to Euler equations on triangular grids. The method was then extended to Navier-Stokes equations by May and Jameson [109] for triangular meshes and Sun *et al.* [110] for hexahedral grids. It is important to notice that for grids based on tensor product cells, the Spectral Difference method formulation is identical to the multi-domain spectral method introduced by Kopriva and Kolias [79].

For tensor product cells, the SD method principle consists in defining two polynomials: a polynomial of degree p for the solution and a polynomial of degree $(p+1)$ for each flux component, leading to an order of accuracy of $(p+1)$. This choice of polynomial degrees ensures the consistency of the formulation. However, contrary to the FR approach, no lifting operator is introduced in the formulation: two sets of points, the Solution Points (SP) and the Flux Points (FP) enable the definition of the Lagrange interpolation polynomials. An alternative approach was derived very recently by Chen *et al.* [111] for tensor-product cells. This technique and the standard one differ in the definition of the flux derivative. In the new formulation, the flux derivative is built from the set of SP plus the interface FP. Such a formulation avoids the need to interpolate from SP to internal FP.

Developments of the SD method also involve multi-grid refinement and implicit time integrators to enhance convergence to steady solutions [112, 113] and sliding-mesh interface approach [114]. The SD method was also used for high-fidelity simulations of 3D vortex-induced vibrations on quadrilateral mesh [115]. In addition, several works concerned the validation of the SD method capability to compute unsteady solutions of compressible flows by means of LES [116–121]. Extensions of the SD method were proposed to handle shocks [122]. The SD method was recently made compatible with the non-reflecting boundary conditions [123], written specifically to cope with the SD algorithm and then coupled with a Time Domain Impedance Boundary Condition (TDIBC) formulation [124, 125]. The SD method was applied to three-dimensional turbulence simulations by Chapelier *et al.* [126] and to astrophysical fluid dynamics by Wang *et al.* [127].

The SD, FR and standard DG methods were compared by Liang *et al.* in [128]. It was proven that the most efficient method is the FR discretization technique and the slowest one is the DG method. The DG method leads to the more accurate results and the FR to the less accurate ones. For both performance and accuracy, the SD method lies in between. Recently, Cox *et al.* [129] compared the accuracy, stability and performance of the standard SD method compare to the FR approach. A nonlinear stability analysis and numerical experiments show that the SD scheme leads to better accuracy and stability. Finally, the quadrature-free DG scheme and the SD method were proven as equivalent under given conditions (use of a nodal Lagrange basis, the quadrature-free paradigm and the numerical flux) for nonlinear hyperbolic conservation laws by May [130].

The stability of the SD method for tensor product cells was studied by Van den Abeele *et al.* [131, 132]. They showed that the SP position influences neither the stability nor the accuracy of the scheme. Jameson confirmed this statement [83] and also showed that for the one-dimensional linear advection case, the SD method is stable for all orders of accuracy in a norm of Sobolev-type provided that the interior flux collocation points are placed at the zeros of the corresponding Legendre polynomials. Vanharen *et al.* revisited the spectral analysis for high-order spectral discontinuous methods and applied it to the 1D SD method in [133].

When considering the standard SD method on simplex cells, stability analysis leads to different conclusions. Van den Abeele *et al.* [132] showed that for an order of accuracy strictly greater than two, the scheme stability is not ensured for triangular cells. For high-order SD schemes on triangular cells, several FP positions are tested but none of them lead to a stable scheme. This explains why after several papers using the SD approach on triangles (see [107, 108, 134–136] among the possible literature), most researchers focused on unstructured grids composed of hexahedra only. To overcome this limitation, Liang *et al.* [137] proposed to decompose any triangular cell into quadrilaterals using cell center and mid-edges, leading to quadrilateral cells of half the size of the original element. Using this option, a 2D hybrid mesh is transformed into an unstructured grid composed of quadrilaterals only but the number of mesh elements is strongly increased. Finally, one must also mention the work of Meister *et al.* on the SD method on triangles based on Proriol-Koornwinder-Dubiner basis on triangles for both solution and flux polynomials [138, 139]. Using this approach, the set of FP is chosen as the set of two-dimensional Lobatto points on a triangle proposed by Blyth and Pozrikidis [140], leading to FP located at triangle vertices. Since a triangle vertex is generally shared by more than two triangles, this choice is questionable to properly define the inputs of the Riemann problem.

Balan *et al.* proposed another alternative in [141, 142]. Instead of splitting any mesh cell into subcells to define the computational grid, they build an alternative SD formulation using Raviart-Thomas elements on triangles, leading to the naming SDRT. The SDRT scheme is proven to be linearly stable up to the fourth-order under a Fourier stability analysis originally initiated by May and Schöberl [143] and validated on Euler test cases. The SDRT method was then extended to simulate 2D viscous flows on unstructured hybrid grids up to the fourth-order by Li *et al.* [144] and used for the simulation of vortex-induced vibrations using a sliding-mesh method on hybrid grids by Qiu *et al.* [145].

1.5. Motivations for the Thesis

The thesis addresses the definition of a high-order convection-diffusion scheme for unstructured hybrid grids based on a compact formulation. The standard staggered SD approach was chosen to be implemented in the high-order solver JAGUAR (proJect of an Aerodynamic solver using General Unstructured grids And high-ordeR schemes) [146] because of its accuracy [133] and its efficiency [147] for LES. The JAGUAR solver is owned by ONERA and CERFACS and aims to treat applications including combustion, transition, shock capturing and re-entry unsteady flows using HPC. The treatment of quadrilaterals and hexahedra has been successfully implemented following tensor product rules.

In this context, the present thesis focuses on the extension of the JAGUAR solver to deal with hybrid unstructured grids composed of standard element shapes (triangles and quadrilaterals in 2D; hexahedra, tetrahedra and prismatic elements in 3D) while maintaining stable and robust properties. Before performing advanced LES, it is of strong importance to propose a stable formulation of the SD method on simplex cells. The first step is to extend the SDRT approach introduced by Balan *et al.* [142] for orders of accuracy higher than four for 2D cases. The SDRT formulation on tetrahedral elements has then to be established. The linear stability of the spatial SDRT discretization coupled with the temporal scheme has to be carefully analyzed to determine reliable stability limits. Finally, the implementation of the SDRT formulation in JAGUAR for triangular, 2D hybrid and tetrahedral grids allows one to investigate the scheme properties for fluid flow problems.

1.6. Outline

The outline of this thesis is organized as follows.

In Chapter 2, the Spectral Difference method is presented in detail. The standard SD formulation is first recalled in the 1D case and for tensor product elements. The extension to simplex cells using Raviart-Thomas elements is then introduced for triangles and developed for tetrahedra. The SD formulation on prismatic elements is obtained using a tensor product of the 1D standard SD approach and the 2D SDRT method. Finally, the complexity induced by the treatment of pyramids is addressed.

Chapter 3 examines the linear stability properties of the standard 1D SD method and the SDRT spatial discretization on triangular and tetrahedral elements. The common Fourier stability analysis is presented for triangles and extended to tetrahedral elements. A different way to assess stability is proposed and results are compared with the Fourier stability analysis for the 1D standard SD scheme and the SDRT scheme on triangles.

In Chapter 4, the linear stability properties of the SDRT spatial discretization based on interior FP located at quadrature points coupled with different temporal schemes are studied for simplex cells in 2D (triangles) and 3D (tetrahedra) using Fourier analysis. For spatially stable SDRT schemes, the analysis of the coupled time-space discretization is conducted to determine stability limits for each temporal scheme. For unstable SDRT schemes, the ability of time integration schemes to stabilize the full discretization through their numerical dissipation properties is examined.

Chapter 5 proposes an optimization technique to determine stable SDRT formulations. Sets of FP leading to stable SDRT formulations on triangles are given up to the sixth-order of accuracy ($p = 5$). Results of the Fourier analysis are presented for both spatial and coupled time-space discretizations. The optimization process is then extended to $p = 6$ on triangles and $p = 3$ on tetrahedral elements. For tetrahedral elements, the influence of the position of FP located on faces is examined.

Validation test cases are presented for first and second-order Partial Differential Equations (PDEs) in Chapters 6 and 7, respectively. The advection of a sine and the convection of an isentropic vortex test cases are used to assess the accuracy of the SDRT discretization on triangular, tetrahedral and 2D hybrid grids. The 2D Navier-Stokes equations are then numerically solved to simulate a flow over a NACA 0012 airfoil using a quadratic triangular mesh and a flow around a circular cylinder using a 2D hybrid grid. The SDRT method is eventually validated for the 3D simulation of the Taylor-Green Vortex using tetrahedral elements.

Finally, conclusions are given and future perspectives are considered.

1.7. Publications

1.7.1. Papers

- A. Veilleux, G. Puigt, H. Deniau and G. Daviller. A stable Spectral Difference approach for computations with triangular and hybrid grids up to the sixth-order of accuracy. *Submitted for Journal of Computational Physics in August 2020.*
- A. Veilleux, G. Puigt, H. Deniau and G. Daviller. Stable Spectral Difference approach using Raviart-Thomas elements for 3D computations on tetrahedral grids. *Under Preparation, to be submitted.*

1.7.2. Presentations

- A. Veilleux, G. Puigt, H. Deniau and G. Daviller. Towards a multi-element-shape extension for the Spectral Difference method. *NAHOM Con 19*, San Diego, CA, Jun 2-5, 2019.
- A. Veilleux, G. Puigt, H. Deniau and G. Daviller. Extension of the Spectral Difference method using Raviart-Thomas elements to the sixth-order of accuracy on triangles and formulation on tetrahedra. *16th U.S. National Congress on Computational Mechanics*, July 25-29, 2021.

Chapter 2

The Spectral Difference Method

In this chapter, the standard Spectral Difference method is first presented for the 1D case and for tensor-product cells in 3D (hexahedral elements). The formulation extension on simplex cells, named the SDRT method, is presented for triangles in 2D and developed for tetrahedron in 3D. A brief theoretical comparison between the FR and the SD method based on the number of SP is established to highlight differences between the two methods. The procedure to deal with prismatic elements using a tensor product of the 2D SDRT scheme and the 1D standard SD formulation is detailed. Finally, the complexity induced by the treatment of pyramids is addressed and possible solutions used in the literature are given.

Overview

2.1. One-Dimensional Formulation . . . 18
2.2. Tensor Product Elements Formulation . . . 21
2.3. Triangular and Tetrahedral Elements Formulation . . . 25
2.3.1. Algebraic Framework . . . 25
2.3.2. General Principle . . . 27
2.3.3. Raviart-Thomas Basis . . . 28
2.3.4. The Spectral Difference Scheme Using Raviart-Thomas Elements . . . 30
2.3.5. Comparison Between Spectral Difference and Flux Reconstruction Methods for Simplex Elements . . . 34
2.4. Prismatic Elements Formulation . . . 36
2.4.1. Reference Element . . . 36
2.4.2. Solution Approximation . . . 36
2.4.3. Definition of the Flux Polynomial from the Set of Flux Values at Flux Points . . . 37
2.4.4. Differentiation of the Flux Polynomial at Solution Points . . . 38
2.4.5. Semi-Discretized Form . . . 38
2.5. Pyramidal Elements . . . 39

2.1. One-Dimensional Formulation

To get a $(p+1)$-th-order accurate scheme, the Spectral Difference principle is to represent the solution as a polynomial of degree p. To ensure consistency, the flux derivative has to be represented by a polynomial of degree p, leading to a representation of the flux density as a polynomial of degree $p+1$. In 1D, the solution is defined within each cell on a first set of $N_{SP}^{1d} = p+1$ points called Solution Points (SP) and denoted $\xi_j, j \in [\![1, N_{SP}^{1d}]\!]$ whereas fluxes are computed in another set of $N_{FP}^{1d} = p+2$ points called Flux Points (FP) and denoted $\xi_k, k \in [\![1, N_{FP}^{1d}]\!]$. In the standard 1D formulation, the SD method is said staggered: one FP is located between two adjacent SP and additional FP are located at the segment endpoints. The detailed implementation is described by the following steps.

Let us consider an hyperbolic equation under its differential form in 1D:

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0, \quad \text{in } \Omega \times [0, t_f], \tag{2.1}$$

on a domain Ω, where x is a spatial coordinate, t is time, u is a conservative variable, $f = f(u)$ is the flux in the x direction and t_f is the final time of the computation. The computational domain Ω is divided into N non-overlapping elements Ω_i:

$$\Omega = \bigcup_{i=1}^{N} \Omega_i. \tag{2.2}$$

For implementation simplicity, Eq. (2.1) is solved in the reference domain. Each element Ω_i of the domain Ω is transformed into a standard element $\mathcal{S} := \{\xi, 0 \leq \xi \leq 1\}$. This transformation can be written:

$$x = \sum_{i=1}^{n} M_i(\xi) x_i, \tag{2.3}$$

where x_i are the Cartesian coordinates in the physical domain of the n vertices of the cell and $M_i(\xi)$ are the shape functions. The non-singular Jacobian matrix of this transformation from the physical domain x to the reference domain ξ is:

$$J = \frac{\partial x}{\partial \xi}, \tag{2.4}$$

and the inverse transformation is related to the Jacobian matrix according to:

$$\frac{\partial \xi}{\partial x} = J^{-1}. \tag{2.5}$$

The shape functions and Jacobian matrix expressions are given in Appendix A for linear (A.2.1) and quadratic (A.2.2) 1D element. Under this mapping, Eq. (2.1) becomes:

$$\frac{\partial \hat{u}}{\partial t} + \frac{\partial \hat{f}}{\partial \xi} = 0, \quad \text{in } \mathcal{S} \times [0, t_f], \tag{2.6}$$

where the transformed variables in the reference domain are defined as:

$$\hat{u} = |J| \cdot u, \tag{2.7}$$

$$\hat{f} = |J| J^{-1} \cdot f. \tag{2.8}$$

To solve Eq. (2.6) with a $(p+1)$-th-order of accuracy, the Spectral Difference approach approximates the solution $\hat{u}$ with a polynomial $\hat{u}_h$ of degree p on each cell Ω_i. Given the solution values $\hat{u}_j$ defined

on a set of SP $\xi_j, j \in [\![1, N_{SP}^{1d}]\!]$, the polynomial approximation $\hat{u}_h$ is defined in the reference element $\mathcal{S}$ using a Lagrange interpolation of degree p:

$$\hat{u}_h(\xi) = \sum_{j=1}^{N_{SP}^{1d}} \hat{u}_j \, l_j(\xi), \tag{2.9}$$

where

$$l_j(\xi) = \prod_{s=1, s\neq j}^{N_{SP}^{1d}} \frac{\xi - \xi_s}{\xi_j - \xi_s}, \tag{2.10}$$

and

$$N_{SP}^{1d} = p + 1. \tag{2.11}$$

Usually, SP are taken as Gauss points:

$$\xi_j = \frac{1}{2}\left[1 - \cos\left(\frac{2j-1}{2(p+1)} \cdot \pi\right)\right], \; j \in [\![1, N_{SP}^{1d}]\!]. \tag{2.12}$$

The flux is approximated using a Lagrange interpolation polynomial of degree $p+1$:

$$\hat{f}_h(\xi) = \sum_{k=1}^{N_{FP}^{1d}} \hat{f}_k \, l_k(\xi), \tag{2.13}$$

where

$$l_k(\xi) = \prod_{s=1, s\neq i}^{N_{FP}^{1d}} \frac{\xi - \xi_s}{\xi_k - \xi_s}, \tag{2.14}$$

and

$$N_{FP}^{1d} = p + 2. \tag{2.15}$$

The FP are taken as Gauss-Lobatto points or as the roots of the Legendre polynomial of degree p plus the segment endpoints. The initial choice, the Gauss-Lobatto points (see [113, 137, 148, 149] for instance), was demonstrated to lead to unstable simulations for some values of p whereas the choice of the Legendre roots was demonstrated as stable for any order of accuracy by Jameson [83]. The latter choice gives today the standard position for the flux points. The flux polynomial approximation given by Eq. (2.13) is built from N_{FP}^{1d} flux values. To determine them, the solution polynomial approximation given by Eq. (2.9) is first used to compute solution values at FP, denoted $\hat{u}_k$. The flux values $\hat{f}_k$ at interior FP can then be computed directly from the approximate solution values at FP:

$$\hat{f}_k = \hat{f}(\xi_k) = \hat{f}(\hat{u}_h(\xi_k)), \quad \xi_k \in \mathcal{S} \setminus \partial\mathcal{S}. \tag{2.16}$$

The flux is likely to be discontinuous at the interface between elements since it is computed from the approximated solution which is continuous on each cell Ω_i but has no reason to be continuous on the whole domain Ω. At element interfaces, two different values are available at a single flux point. At the left interface, one comes from the right state, i.e., the considered cell i, and one comes from the left state, i.e., cell $i-1$. At the right interface, one comes from the left state, i.e., the considered cell i, and one comes from the right state, i.e., cell $i+1$. To have a continuous flux function on the whole domain Ω and thus to ensure conservation, a Riemann solver is used to compute a single flux value. Note that the Riemann solver gives the numerical flux f^*, computed from the physical values of the approximated solution at FP located on edges. In 1D, the numerical flux f^* is equal to the reference numerical flux $\hat{f}^*$. However, it won't be the case in 2D and 3D, thus the relation between

the numerical flux in the reference and in the physical domain will have to be expressed. For FP on edges, the numerical flux is used as the flux value:

$$\hat{f}_k = \hat{f}_k^*, \quad \xi_k \in \partial\mathcal{S}. \tag{2.17}$$

Once flux values are known at flux points and uniquely defined, the flux divergence at SP can be computed:

$$\left(\frac{\partial \hat{f}_h}{\partial \xi}\right)_j = \sum_{k=1}^{N_{FP}^{1d}} \hat{f}_k \cdot \frac{\partial l_k}{\partial \xi}(\xi_j). \tag{2.18}$$

These values are finally used to update the solution in time on each cell i using a temporal discretization of the following semi-discrete equation:

$$\frac{\partial \hat{u}_j^{(i)}}{\partial t} = -\sum_{k=1}^{N_{FP}^{1d}} \hat{f}_k^{(i)} \cdot \frac{\partial l_k}{\partial \xi}(\xi_j). \tag{2.19}$$

2.2. Tensor Product Elements Formulation

The SD method can easily be extended from the 1D case to tensor-product elements. The procedure is described in this section for 3D tensor-product cells, i.e., hexahedral elements. Let us consider an hyperbolic equation under its differential form in 3D:

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f} = 0, \quad \text{in } \Omega \times [0, t_f], \tag{2.20}$$

where u is a conservative variable, $\mathbf{f} = (f, g, h)$ are fluxes in the x, y and z directions respectively and t_f is the final time of the computation. The computational domain Ω is divided into N non-overlapping conforming hexahedral elements Ω_i. As for the 1D case, Eq. (2.20) is solved in the reference domain. Each element Ω_i of the domain Ω is mapped to a standard hexahedron $\mathcal{H} = \{(\xi, \eta, \zeta), 0 \leq \xi, \eta, \zeta \leq 1\}$. This transformation can be written:

$$\mathbf{x} = \sum_{i=1}^{n} M_i(\boldsymbol{\xi}) \mathbf{x}_i, \tag{2.21}$$

where $\mathbf{x}_i = (x_i, y_i, z_i)$ are the Cartesian coordinates in the physical domain of the n vertices of the cell and $M_i(\xi)$ are the shape functions. The non-singular Jacobian matrix of this transformation from the physic (x, y, z) to the reference domain (ξ, η, ζ) is:

$$J = \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} = \begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix}, \tag{2.22}$$

and the inverse transformation is related to the Jacobian matrix according to:

$$\frac{\partial(\xi, \eta, \zeta)}{\partial(x, y, z)} = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} = J^{-1}. \tag{2.23}$$

The shape functions and Jacobian matrix expressions are given in Appendix A for the linear (A.3.3) and quadratic (A.3.4) quadrilateral and the linear hexahedron (A.4.3). Under this mapping, Eq. (2.20) becomes:

$$\frac{\partial \hat{u}}{\partial t} + \frac{\partial \hat{f}}{\partial \xi} + \frac{\partial \hat{g}}{\partial \eta} + \frac{\partial \hat{h}}{\partial \zeta} = 0, \quad \text{in } \mathcal{H} \times [0, t_f], \tag{2.24}$$

where the transformed variables in the reference domain are defined as:

$$\hat{u} = |J| \cdot u, \tag{2.25}$$

$$\begin{pmatrix} \hat{f} \\ \hat{g} \\ \hat{h} \end{pmatrix} = |J| J^{-1} \cdot \begin{pmatrix} f \\ g \\ h \end{pmatrix}. \tag{2.26}$$

The Spectral Difference approach to solve Eq. (2.24) on tensor product cells consists in a 1D treatment in each direction. On tensor product cell, the number of points needed to represent the solution as a polynomial of degree p is given by:

$$N_{SP}^d = (p+1)^d, \tag{2.27}$$

and the number of points needed to represent the flux as a polynomial of degree $p+1$ is:

$$N_{FP}^d = d(p+2)(p+1)^{d-1}, \tag{2.28}$$

where d is the dimension. To highlight the tensor product properties, the solution points $\boldsymbol{\xi}_j = (\xi_j, \eta_j, \zeta_j), j \in [\![1, N_{SP}^d]\!]$ are denoted:

$$\boldsymbol{\xi}_{(j_1,j_2,j_3)},\ (j_1, j_2, j_3) \in [\![1, N_{SP}^{1d}]\!], \tag{2.29}$$

and the set of flux points $\boldsymbol{\xi}_k, k \in [\![1, N_{FP}^d]\!]$ are denoted:

$$\boldsymbol{\xi}_{(k_1,j_2,j_3)},\ k_1 \in [\![1, N_{FP}^{1d}]\!] \quad \text{in the } \xi\text{-direction}, \tag{2.30}$$
$$\boldsymbol{\xi}_{(j_1,k_2,j_3)},\ k_2 \in [\![1, N_{FP}^{1d}]\!] \quad \text{in the } \eta\text{-direction}, \tag{2.31}$$
$$\boldsymbol{\xi}_{(j_1,j_2,k_3)},\ k_3 \in [\![1, N_{FP}^{1d}]\!] \quad \text{in the } \zeta\text{-direction}. \tag{2.32}$$
$$\tag{2.33}$$

An example of solution and flux points distribution on the reference hexahedron is given in Fig. 2.1 for the case $p = 1$.

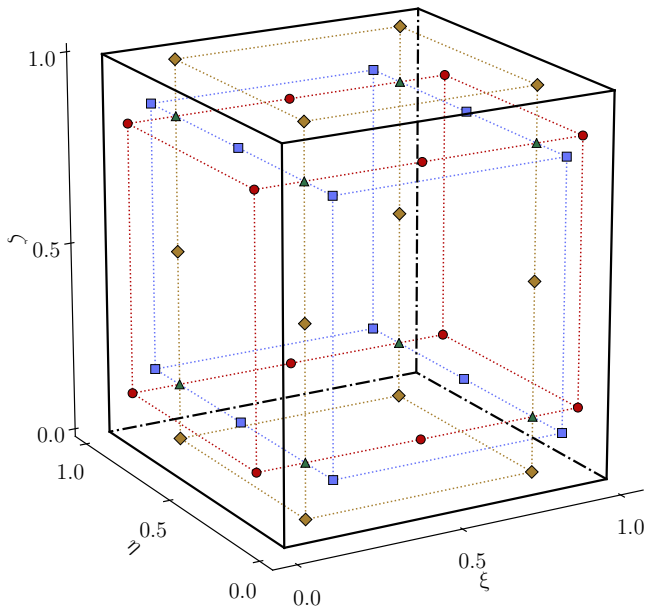


Figure 2.1. – Points distribution in the hexahedral reference element for $p = 1$: SP $\boldsymbol{\xi}_{(j_1,j_2,j_3)}$ (▲), FP in the ξ-direction $\boldsymbol{\xi}_{(k_1,j_2,j_3)}$ (●), FP in the η-direction $\boldsymbol{\xi}_{(j_1,k_2,j_3)}$ (■), FP in the ζ-direction $\boldsymbol{\xi}_{(j_1,j_2,k_3)}$ (◆)

Given the solution $\hat{u}_{(j_1,j_2,j_3)}$ at solution points, the solution is approximated as the tensor products of three one-dimensional Lagrange interpolation polynomials of degree p:

$$\hat{u}_h(\xi, \eta, \zeta) = \sum_{j_3=1}^{p+1} \sum_{j_2=1}^{p+1} \sum_{j_1=1}^{p+1} \hat{u}_{(j_1,j_2,j_3)}\, l_{j_1}(\xi) \cdot l_{j_2}(\eta) \cdot l_{j_3}(\zeta), \tag{2.34}$$

where $l_{j_1}(\xi)$, $l_{j_2}(\eta)$ and $l_{j_3}(\zeta)$ are the 1D Lagrange polynomials associated with the 1D SP located at ξ_{j_1}, η_{j_2} and ζ_{j_3} (respectively) and $\hat{u}_{(j_1,j_2,j_3)}$ is the solution value located at $(\xi_{j_1}, \eta_{j_2}, \zeta_{j_3})$ The approximated solution values at FP ($\hat{u}_{(k_1,j_2,j_3)}$, $\hat{u}_{(j_1,k_2,j_3)}$, $\hat{u}_{(j_1,j_2,k_3)}$) can be computed from Eq. (2.34). At

internal FP, fluxes $\hat{f}_{(k_1,j_2,j_3)}$, $\hat{g}_{(j_1,k_2,j_3)}$ and $\hat{h}_{(j_1,j_2,k_3)}$ are computed from the solution values extrapolated at FP:

$$\hat{\mathbf{f}}_k = \begin{pmatrix} \hat{f}_{(k_1,j_2,j_3)} \\ \hat{g}_{(j_1,k_2,j_3)} \\ \hat{h}_{(j_1,j_2,k_3)} \end{pmatrix} = \begin{pmatrix} \hat{f}\left(\hat{u}_h(\xi_{k_1},\eta_{j_2},\zeta_{j_3})\right) \\ \hat{g}\left(\hat{u}_h(\xi_{j_1},\eta_{k_2},\zeta_{j_3})\right) \\ \hat{h}\left(\hat{u}_h(\xi_{j_1},\eta_{j_2},\zeta_{k_3})\right) \end{pmatrix}, \quad \boldsymbol{\xi}_k \in \mathcal{H} \setminus \partial\mathcal{H}. \tag{2.35}$$

At element interfaces, two different values are available at a single FP and, in order to have a continuous flux function and thus to ensure conservation, a Riemann solver is used to compute a single flux value. The Riemann solver gives the physical normal numerical flux, denoted $(\mathbf{f}\cdot\mathbf{n})^*$ and computed from the physical values of the approximated solution at FP located on edges. The physical normal numerical flux is converted to the transformed space using:

$$\left(\hat{\mathbf{f}}\cdot\hat{\mathbf{n}}\right)^* = (\mathbf{f}\cdot\mathbf{n})^*\,|J^{face}|, \tag{2.36}$$

where J^{face} is the face Jacobian. The relation between the physical and the reference outward normal vectors on a face is:

$$\hat{\mathbf{n}} = \frac{1}{|J|}|J^{face}|J^\top\mathbf{n}, \tag{2.37}$$

or

$$\mathbf{n} = |J|\frac{1}{|J^{face}|}(J^\top)^{-1}\hat{\mathbf{n}}. \tag{2.38}$$

By injecting Eq. (2.38) into Eq. (2.36), one gets:

$$\left(\hat{\mathbf{f}}\cdot\hat{\mathbf{n}}\right)^* = \left(\mathbf{f}\cdot|J|(J^\top)^{-1}\hat{\mathbf{n}}\right)^*. \tag{2.39}$$

The flux values at FP located on edges read:

$$\hat{\mathbf{f}}_k = \left(\hat{\mathbf{f}}_k\cdot\hat{\mathbf{n}}_k\right)^* = \left(\mathbf{f}_k\cdot(|J|J^{-1})^\top\hat{\mathbf{n}}_k\right)^*, \quad \boldsymbol{\xi}_k \in \partial\mathcal{H}. \tag{2.40}$$

Once flux values at flux points are known, the flux components in each direction are approximated by polynomials constructed using Lagrange interpolation polynomials of degree $p+1$ in the corresponding direction:

$$\hat{f}_h(\xi,\eta,\zeta) = \sum_{j_3=1}^{p+1}\sum_{j_2=1}^{p+1}\sum_{k_1=1}^{p+2} \hat{f}_{(k_1,j_2,j_3)}\, l_{k_1}(\xi)\cdot l_{j_2}(\eta)\cdot l_{j_3}(\zeta), \tag{2.41}$$

$$\hat{g}_h(\xi,\eta,\zeta) = \sum_{j_3=1}^{p+1}\sum_{k_2=1}^{p+2}\sum_{j_1=1}^{p+1} \hat{g}_{(j_1,k_2,j_3)}\, l_{j_1}(\xi)\cdot l_{k_2}(\eta)\cdot l_{j_3}(\zeta), \tag{2.42}$$

$$\hat{h}_h(\xi,\eta,\zeta) = \sum_{k_3=1}^{p+2}\sum_{j_2=1}^{p+1}\sum_{j_1=1}^{p+1} \hat{h}_{(j_1,j_2,k_3)}\, l_{j_1}(\xi)\cdot l_{j_2}(\eta)\cdot l_{k_3}(\zeta), \tag{2.43}$$

where $l_{k_1}(\xi)$, $l_{k_2}(\eta)$ and $l_{k_3}(\zeta)$ are the 1D Lagrange polynomials associated with the 1D FP located at ξ_{k_1}, η_{k_2} and ζ_{k_3} (respectively). The divergence of the approximate flux at SP is:

$$\hat{\nabla}\cdot\hat{\mathbf{f}}_h(\xi_{j_1},\eta_{j_2},\zeta_{j_3}) = \left(\frac{\partial\hat{f}_h}{\partial\xi}\right)_{(j_1,j_2,j_3)} + \left(\frac{\partial\hat{g}_h}{\partial\eta}\right)_{(j_1,j_2,j_3)} + \left(\frac{\partial\hat{h}_h}{\partial\zeta}\right)_{(j_1,j_2,j_3)}, \tag{2.44}$$

where

$$\left(\frac{\partial\hat{f}_h}{\partial\xi}\right)_{(j_1,j_2,j_3)} = \sum_{k_1=1}^{p+2}\hat{f}_{(k_1,j_2,j_3)}\cdot\frac{\partial l_{k_1}}{\partial\xi}(\xi_{j_1}), \tag{2.45}$$

$$\left(\frac{\partial \hat{g}_h}{\partial \eta}\right)_{(j_1,j_2,j_3)} = \sum_{k_2=1}^{p+2} \hat{g}_{(j_1,k_2,j_3)} \cdot \frac{\partial l_{k_2}}{\partial \eta}(\eta_{j_2}), \tag{2.46}$$

$$\left(\frac{\partial \hat{h}_h}{\partial \zeta}\right)_{(j_1,j_2,j_3)} = \sum_{k_3=1}^{p+2} \hat{h}_{(j_1,j_2,k_3)} \cdot \frac{\partial l_{k_3}}{\partial \zeta}(\zeta_{j_3}). \tag{2.47}$$

The solution is updated in time on each cell i using a time discretization scheme of the following semi-discrete equation:

$$\frac{\partial \hat{u}_j^{(i)}}{\partial t} = -\sum_{k_1=1}^{p+2} \hat{f}^{(i)}_{(k_1,j_2,j_3)} \cdot \frac{\partial l_{k_1}}{\partial \xi}(\xi_{j_1}) - \sum_{k_2=1}^{p+2} \hat{g}^{(i)}_{(j_1,k_2,j_3)} \cdot \frac{\partial l_{k_2}}{\partial \eta}(\eta_{j_2}) - \sum_{k_3=1}^{p+2} \hat{h}^{(i)}_{(j_1,j_2,k_3)} \cdot \frac{\partial l_{k_3}}{\partial \zeta}(\zeta_{j_3}). \tag{2.48}$$

For tensor product cells, the SD method is easy to implement since only one-dimensional operations are involved. High-order accuracy is achieved through high-order local representation of the state variables. The use of Riemann solvers guarantees that the method is fully conservative. Since operations are mostly one-dimensional, the method is quite efficient. However, using tensor product cells only does not offer much flexibility when it comes to complex geometries. In particular, the tensor product method cannot be applied to simplex cells. The Spectral Difference approach using Raviart-Thomas elements on simplex cells is presented in the next section. Before that, an algebraic framework about polynomial expansion on simplices is given.

2.3. Triangular and Tetrahedral Elements Formulation

2.3.1. Algebraic Framework

2.3.1.1. Polynomial Expansion on Simplices

Let us consider the interpolation of a polynomial of degree p, $u_h(\boldsymbol{\xi}) \in \mathbb{P}_p$, through a set of distinct points $\boldsymbol{\xi}_m, m \in [\![1, N_{SP}]\!]$ defined as:

$$\boldsymbol{\xi}_m = \begin{cases} (\xi_m, \eta_m) & \text{in 2D,} \\ (\xi_m, \eta_m, \zeta_m) & \text{in 3D,} \end{cases} \tag{2.49}$$

where

$$N_{SP} = \begin{cases} (p+1)(p+2)/2 & \text{in 2D,} \\ (p+1)(p+2)(p+3)/6 & \text{in 3D,} \end{cases} \tag{2.50}$$

and

$$\mathbb{P}_p = \begin{cases} span\{\xi^i \eta^j, 0 \leq i, 0 \leq j \text{ and } i+j \leq p & \text{in 2D,} \\ span\{\xi^i \eta^j \zeta^k, 0 \leq i, 0 \leq j, 0 \leq k \text{ and } i+j+k \leq p & \text{in 3D.} \end{cases} \tag{2.51}$$

The polynomial $u_h(\boldsymbol{\xi})$ can be expanded using a nodal or a modal representation. When using the nodal representation, the polynomial is represented in term of point values by way of a Lagrangian interpolant, which is defined as the polynomial of lowest degree that assumes at each value $\boldsymbol{\xi}_j$ the corresponding value u_j so that the function coincide at each point:

$$u_h(\boldsymbol{\xi}) = \sum_{j=1}^{N_{SP}} u_j \, l_j(\boldsymbol{\xi}), \tag{2.52}$$

where l_j is a Lagrange polynomial and u_j are the known solution values at point $\boldsymbol{\xi}_j$. Since there is not a closed-form expression of the Lagrange polynomials through an arbitrary set of points on the triangular element [150], a solution is to expand the polynomial u_h using a modal representation:

$$u_h(\boldsymbol{\xi}) = \sum_{m=1}^{N_{SP}} \bar{u}_m \, \Psi_m(\boldsymbol{\xi}), \tag{2.53}$$

where $\Psi_m(\boldsymbol{\xi}) \in \mathbb{P}_p$ is a complete polynomial basis and $\bar{u}_m$ are the modal basis coefficients, which do not represent the value of a function at a point. Since $u_h(\boldsymbol{\xi})$ and $\Psi_m(\boldsymbol{\xi})$ span the same polynomial space, any projection form will recover the exact expansion coefficient $\bar{u}_m$. By performing a collocation projection at the points $\boldsymbol{\xi}_j$ such that:

$$u_h(\boldsymbol{\xi}_j) = u_j = \sum_{m=1}^{N_{SP}} \bar{u}_m \, \Psi_m(\boldsymbol{\xi}_j), \tag{2.54}$$

the coefficients $\bar{u}_m$ can then be determined as:

$$\bar{u}_m = \sum_{m=1}^{N_{SP}} u_j \, (\Psi_m(\boldsymbol{\xi}_j))^{-1} = \sum_{m=1}^{N_{SP}} u_j \, (\mathcal{V}_{j,m})^{-1}. \tag{2.55}$$

The term $\Psi_m(\boldsymbol{\xi}_j)$ corresponds to the matrix of basis change, also known as the generalized Vandermonde matrix $\mathcal{V}_{j,m} = \Psi_m(\boldsymbol{\xi}_j)$. The choice of the basis $\Psi_m(\boldsymbol{\xi})$ is of primary importance since a matrix inversion is involved in the polynomial expansion process. The chosen basis will dictate the conditioning of the matrix $\mathcal{V}$ and thus the computational stability. The most straightforward choice would be

the monomial basis ($\{1, x, y, x^2, xy, y^2, ..., y^p\}$ in 2D). However, this choice leads to a dense Vandermonde matrix whose condition number rapidly increases with the degree p. A solution is to choose a hierarchical orthogonal basis, whose Vandermonde matrix is diagonal and thus better conditioned. An appropriated basis choice is to define $\Psi_m(\boldsymbol{\xi})$ as the Proriol-Koornwinder-Dubiner (PKD) basis, which has been defined on triangles by Proriol [151], Koornwinder [152] and Dubiner [153] and on tetrahedra by Sherwin and Karniadakis in [154].

From the literature [150, 155], three main assets of the PKD basis can be noted. First, it is based on Jacobi polynomials, which can be evaluated to a high degree using simple recurrence relations. The PKD L^2 orthogonality will then tend to a well-conditioned Vandermonde matrix. Finally, the PKD basis hierarchical nature (the expansion set of degree p contains the expansion set of degree $p-1$) simplifies the construction of certain finite element spaces, such as the Raviart-Thomas space, which will be used to approximate the flux function in the SDRT formulation. The PKD basis is detailed in the following section.

2.3.1.2. Proriol-Koornwinder-Dubiner Basis

This section follows the generalized tensor product modal expansion notations defined by Karniadakis and Sherwin [150].

Collapsed Coordinated System

2D The PKD basis is obtained by forming a tensor product of one-dimensional Jacobi polynomials based on a Cartesian coordinate system. Since the Jacobi polynomials are defined on the reference interval $[-1, 1]$, the 2D PKD basis obtained using tensor product is defined on the reference quadrilateral $\mathcal{Q}^{PKD}$ bounded by the same constant limits, i.e., $\mathcal{Q}^{PKD} := \{(x, y) : -1 \leq x, y \leq 1\}$. To express the PKD basis on the reference triangle $\mathcal{T}^{PKD} := \{(x, y) : -1 \leq x, y \leq 0, x + y \leq 0\}$, the first step is to map the reference triangle onto the reference quadrilateral. This transformation is defined by:

$$\begin{cases} \xi = 2\dfrac{(1+x)}{(1-y)} - 1, \\ \eta = y, \end{cases} \quad \text{or} \quad \begin{cases} x = \dfrac{(1+\xi)(1-\eta)}{2} - 1, \\ y = \eta. \end{cases} \tag{2.56}$$

The triangular element can now be described with the new coordinates as $\mathcal{T}^{PKD} := \{(\xi, \eta) : -1 \leq \xi, \eta \leq 1\}$, bounded by the very same constant limits. The coordinates system (ξ, η) is referred to as the collapsed coordinate system or Duffy coordinates.

3D Following the same reasoning, the tetrahedron $\mathcal{T}_e^{PKD} := \{(x, y, z) : -1 \leq x, y, z, x + y + z \leq 1\}$ is mapped onto the reference hexahedron $\mathcal{H}^{PKD} := \{(x, y, z) : -1 \leq x, y, z \leq 1\}$. The mapping is done by applying the 2D rectangle-to-triangle transformation given by Eq. (2.56) several times : the hexahedron is mapped into a prism, the prism into a pyramid and the pyramid into a tetrahedron. The transformation from the hexahedron coordinates (x, y, z) to the collapsed 3D coordinates (ξ, η, ζ) is given by:

$$\begin{cases} \xi = \dfrac{2(1+x)}{-y-z} - 1, \\ \eta = \dfrac{2(1+y)}{1-z} - 1, \\ \zeta = z, \end{cases} \quad \text{or} \quad \begin{cases} x = \dfrac{(1+\xi)(1-\eta)(1-\zeta)}{4} - 1, \\ y = \dfrac{(1+\eta)(1-\zeta)}{2} - 1, \\ z = \zeta. \end{cases} \tag{2.57}$$

The tetrahedron can now be described with the new coordinates as $\mathcal{T}_e^{PKD} := \{(\xi, \eta, \zeta) : -1 \leq \xi, \eta, \zeta \leq 1\}$, bounded by the very same constant limits as the hexahedron.

Proriol-Koornwinder-Dubiner Basis Using the collapsed coordinates, the PKD basis is constructed as a family of L^2-orthogonal polynomials on $\mathcal{T}^{PKD}$ derived from a warped product of one-dimensional Jacobi polynomials:

$$\Phi_{i,j} = P_i^{0,0}(\xi)\left(\frac{1-\eta}{2}\right)^i P_j^{2i+1,0}(\eta), \quad i+j \leq p. \tag{2.58}$$

The PKD basis was extended in 3D by Sherwin and Karniadakis[154] and is defined on $\mathcal{T}_e^{PKD}$ as:

$$\Phi_{i,j,k} = P_i^{0,0}(\xi)\left(\frac{1-\eta}{2}\right)^i P_j^{2i+1,0}(\eta)\left(\frac{1-\zeta}{2}\right)^{i+j} P_k^{2(i+j+1),0}(\zeta), \quad i+j+k \leq p. \tag{2.59}$$

In Eq. (2.58) and Eq. (2.59), $P_i^{\alpha,\beta}$ denotes the corresponding n-th-order Jacobi polynomials on the interval $[-1,1]$ which are orthogonal polynomials under the Jacobi weight $(1-x)^\alpha(1+x)^\beta$, i.e.:

$$\int_{-1}^{1}(1-x)^\alpha(1+x)^\beta P_i^{\alpha,\beta}(x)P_j^{\alpha,\beta}(x)dx = \frac{2^{\alpha+\beta+1}}{2i+\alpha+\beta+1}\frac{\Gamma(i+\alpha+1)\Gamma(i+\beta+1)}{i!\Gamma(i+\alpha+\beta+1)}\delta_{ij}, \tag{2.60}$$

where Γ is the Gamma function

$$\Gamma(n) = (n-1)!. \tag{2.61}$$

In the case of $\beta = 0$, Eq. (2.60) can be simplified as:

$$\int_{-1}^{1}(1-x)^\alpha P_i^{\alpha,0}(x)P_j^{\alpha,0}(x)dx = \frac{2^{\alpha+1}}{2i+\alpha+1}\delta_{ij}. \tag{2.62}$$

Remark: The 2D (respectively 3D) PKD basis functions are polynomials in both (x,y) and (ξ,η) (respectively (x,y,z) and (ξ,η,ζ)) spaces.

Remark: The "warped product" property refers to the fact that the 2D (respectively 3D) PKD basis functions can be expressed as the product of two polynomials, one in ξ and one in η (respectively three polynomials).

Normalized Proriol-Koornwinder-Dubiner Basis In the following sections, the L^2 normalized PKD basis is used. The proof of L^2 orthogonality and the normalization are given in Appendix B. The normalized 2D and 3D PKD basis are:

$$\Phi_{i,j} = \sqrt{(i+1/2)(i+j+1)}\, P_i^{0,0}(\xi)\left(\frac{1-\eta}{2}\right)^i P_j^{2i+1,0}(\eta), \quad i+j \leq p, \tag{2.63}$$

$$\Phi_{i,j,k} = \sqrt{(i+1/2)(i+j+1)(i+j+k+3/2)}\, P_i^{0,0}(\xi)\left(\frac{1-\eta}{2}\right)^i P_j^{2i+1,0}(\eta)\left(\frac{1-\zeta}{2}\right)^{i+j} P_k^{2(i+j+1),0}(\zeta), \quad i+j+k \leq p. \tag{2.64}$$

2.3.2. General Principle

As mentioned in Chapter 1, the stability of the standard SD method on triangular cells is not ensured for an order of accuracy strictly greater than 2. In this framework, Balan *et al.* proposed another alternative in [141, 142]. They build a SD formulation using Raviart-Thomas elements on triangles, leading to the naming SDRT. The SDRT scheme was proven to be linearly stable up to the fourth-order under a Fourier stability analysis originally initiated by May [143]. To obtain a $(p+1)$-th-order accurate scheme, the solution in the reference domain $\hat{u}$ is approximated by a polynomial of degree p denoted $\hat{u}_h$. The polynomial $\hat{u}_h$ is expanded in the orthonormal PKD basis introduced in section2.3.1. From the polynomial expansion of $\hat{u}_h$, solution values at FP can then be computed, as

for the standard SD scheme. The main difference with the standard SD scheme comes from the flux approximation. With the standard SD scheme, the flux is approximated by $\hat{\mathbf{f}}_h$ in the reference domain by projecting it component-wise into a finite dimensional polynomial space of degree $p+1$ as:

$$\hat{\mathbf{f}}_h(\boldsymbol{\xi}) = \sum_{k=1}^{N_{FP}^s} \hat{\mathbf{f}}_k l_k(\boldsymbol{\xi}), \tag{2.65}$$

where

$$N_{FP}^s = \begin{cases} \dfrac{(p+2)(p+3)}{2} & \text{in 2D,} \\ \dfrac{(p+2)(p+3)(p+4)}{6} & \text{in 3D,} \end{cases} \tag{2.66}$$

and $l_k(\boldsymbol{\xi}), k \in [\![1, N_{FP}^s]\!]$ is the one-dimensional Lagrange polynomial associated with the k-th FP. In Eq. (2.65), the basis is scalar and the coefficients are vectors. With the SDRT scheme, the flux function is approximated in the Raviart-Thomas space as:

$$\hat{\mathbf{f}}_h(\boldsymbol{\xi}) = \sum_{k=1}^{N_{FP}^{RT}} \hat{f}_k \boldsymbol{\psi}_k(\boldsymbol{\xi}), \tag{2.67}$$

where

$$N_{FP}^{RT} = \begin{cases} N_{FP}^{tri} = (p+1)(p+3) & \text{in 2D,} \\ N_{FP}^{tet} = \dfrac{(p+1)(p+2)(p+4)}{2} & \text{in 3D,} \end{cases} \tag{2.68}$$

and $\boldsymbol{\psi}_k$ are interpolation functions which form a basis in the Raviart-Thomas space. In Eq. (2.67), the basis functions are vectors and the coefficients are scalars. By nature, the Raviart-Thomas space is the smallest polynomial space such that the divergence maps it onto the space of polynomial of degree p. This ensures that the solution and the flux divergence will both be polynomials of degree p. In the next sections, the Raviart-Thomas basis and its interpolation functions are first detailed, followed by the description of the SDRT scheme methodology for triangles and its extension to tetrahedral elements.

2.3.3. Raviart-Thomas Basis

The Raviart-Thomas (RT) finite element spaces were originally introduced by Raviart and Thomas [156] to approximate the Sobolev space H(div) defined by:

$$\mathrm{H(div)} = \{u \in \left(L^2(K)\right)^d,\ \nabla \cdot\ u \in L^2(K)\}, \tag{2.69}$$

where d is the dimension, K is a bounded open subset of $\mathbb{R}^d$ with a Lipshitz continuous boundary, $L^2(K)$ is the Hilbert space of square integrable function defined on K. The extension to the three-dimensional case considering K as a tetrahedron or a cube was proposed by Nedelec [157]. The space RT_p spanned by the Raviart-Thomas basis functions of degree p is the smallest polynomial space such that the divergence maps RT_p onto $\mathbb{P}_p$, the space of piecewise polynomials of degree $\leq p$. The definition of the space RT_p is given in Table 2.1.

	Triangle (2D)	Tetrahedron (3D)
RT_p	$(\mathbb{P}_p)^2 + \begin{pmatrix} x \\ y \end{pmatrix} \bar{\mathbb{P}}_p$	$(\mathbb{P}_p)^3 + \begin{pmatrix} x \\ y \\ z \end{pmatrix} \bar{\mathbb{P}}_p$
$\mathbb{P}_p$ (space of polynomials of degree at most p)	$span\{x^i y^j,\ i,j \geq 0, i+j \leq p\}$	$span\{x^i y^j z^k,\ i,j,k \geq 0, i+j+k \leq p\}$
$\bar{\mathbb{P}}_p$ (space of polynomials of degree p)	$span\{x^i y^j,\ i,j \geq 0, i+j = p\}$	$span\{x^i y^j z^k,\ i,j,k \geq 0, i+j+k = p\}$
dim $\mathbb{P}_p$	$\frac{(p+1)(p+2)}{2}$	$\frac{(p+1)(p+2)(p+3)}{6}$
dim $(\mathbb{P}_p)^d$	$(p+1)(p+2)$	$\frac{(p+1)(p+2)(p+3)}{2}$
dim $\bar{\mathbb{P}}_p$	$p+1$	$\frac{(p+1)(p+2)}{2}$
dim RT_p	$(p+1)(p+3)$	$\frac{(p+1)(p+2)(p+4)}{2}$

Table 2.1. – Definition of the RT_p space

Example: Determination of the Monomials for the Two-Dimensional Raviart-Thomas Space In this example, the 8 monomials which form the 2D RT_1 basis are determined.

$$\mathbb{P}_1(x,y) = span\{1,x,y\}, \tag{2.70}$$

$$\mathbb{P}_1^2(x,y) = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} x \\ 0 \end{pmatrix}, \begin{pmatrix} y \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ x \end{pmatrix}, \begin{pmatrix} 0 \\ y \end{pmatrix} \right\}, \tag{2.71}$$

$$\bar{\mathbb{P}}_1(x,y) = span\{x,y\}, \tag{2.72}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} \bar{\mathbb{P}}_1 = \left\{ \left(\begin{pmatrix} x \\ y \end{pmatrix} x \right), \left(\begin{pmatrix} x \\ y \end{pmatrix} y \right) \right\} = \left\{ \begin{pmatrix} x^2 \\ xy \end{pmatrix}, \begin{pmatrix} xy \\ y^2 \end{pmatrix} \right\}, \tag{2.73}$$

$$RT_1 = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} x \\ 0 \end{pmatrix}, \begin{pmatrix} y \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ x \end{pmatrix}, \begin{pmatrix} 0 \\ y \end{pmatrix}, \begin{pmatrix} x^2 \\ xy \end{pmatrix}, \begin{pmatrix} xy \\ y^2 \end{pmatrix} \right\}. \tag{2.74}$$

2.3.4. The Spectral Difference Scheme Using Raviart-Thomas Elements

Let us consider the following scalar conservation law under its differential form:

$$\frac{\partial u(\mathbf{x},t)}{\partial t} + \nabla \cdot \mathbf{f}(u) = 0, \quad \text{in } \Omega \times [0, t_f], \tag{2.75}$$

where u is a state variable, ∇ is the differential operator in the physical domain, t_f is the final time of the computation and $\mathbf{f}$ is the flux vector. In the case of a 2D triangular mesh, $\mathbf{f} = (f, g)$ where f and g are flux densities in the x and y directions respectively and $\mathbf{x} = (x, y)$. In the case of a 3D tetrahedral mesh, $\mathbf{f} = (f, g, h)$ where f, g and h are flux densities in the x, y and z directions respectively and $\mathbf{x} = (x, y, z)$. The computational domain Ω is discretized into N non-overlapping cells (triangles or tetrahedra) and the i-th element is denoted Ω_i:

$$\Omega = \bigcup_{i=1}^{N} \Omega_i. \tag{2.76}$$

2.3.4.1. Reference Element

For implementation simplicity, Eq. (2.75) is solved in the reference domain. Each cell Ω_i of the domain Ω is mapped into a reference triangle:

$$\mathcal{T} := \{(\xi, \eta) : 0 \leq \xi, \eta \leq 1, \xi + \eta \leq 1\}, \tag{2.77}$$

or a reference tetrahedron:

$$\mathcal{T}_e := \{(\xi, \eta, \zeta) : 0 \leq \xi, \eta, \zeta \leq 1, \xi + \eta + \zeta \leq 1\}. \tag{2.78}$$

The transformation can be written as:

$$\mathbf{x} = \sum_{i=1}^{n} M_i(\boldsymbol{\xi}) \mathbf{x}_i, \tag{2.79}$$

where $\boldsymbol{x_i}$ are the Cartesian coordinates in the physical domain of the n vertices of the cell and $M_i(\boldsymbol{\xi})$ are the shape functions. The Jacobian matrix of the transformation given by Eq. (2.79) from the physical to the reference element is determined from the shape functions derivatives and the physical nodes coordinates. Jacobian matrix's expressions are given in Appendix A for the linear (A.3.1) and quadratic (A.3.2) triangle and the linear (A.4.1) tetrahedron. In the reference domain, Eq. (2.75) becomes:

$$\frac{\partial \hat{u}(\boldsymbol{\xi},t)}{\partial t} + \hat{\nabla} \cdot \hat{\mathbf{f}} = 0, \tag{2.80}$$

where $\hat{\nabla}$ is the differential operator in the reference domain and $\hat{u}$, $\hat{\mathbf{f}}$ are the solution and the flux in the reference domain defined by:

$$\hat{u} = |J| u, \tag{2.81}$$

and

$$\hat{\mathbf{f}} = |J| J^{-1} \mathbf{f}. \tag{2.82}$$

The SDRT method solves the hyperbolic equations in the reference element following the procedure described below.

2.3.4.2. Solution Approximation

The solution $\hat{u}$ is approximated as a polynomial of degree p using a modal expansion:

$$\hat{u}_h(\boldsymbol{\xi}) = \sum_{m=1}^{N_{SP}} \bar{u}_m \, \Psi_m(\boldsymbol{\xi}). \tag{2.83}$$

As mentioned in section 2.3.1, the complete polynomial basis Ψ_m is chosen as the PKD orthonormal basis, denoted Φ_m and defined in 2D as:

$$\Phi_m = \sqrt{(i+1/2)(i+j+1)} \; P_i^{0,0}(\xi) \left(\frac{1-\eta}{2}\right)^i P_j^{2i+1,0}(\eta), \quad i+j \leq p, \tag{2.84}$$

and in 3D as:

$$\Phi_m = \sqrt{(i+1/2)(i+j+1)(i+j+k+3/2)} \; P_i^{0,0}(\xi) \left(\frac{1-\eta}{2}\right)^i P_j^{2i+1,0}(\eta) \left(\frac{1-\zeta}{2}\right)^{i+j} P_k^{2(i+j+1),0}(\zeta), \quad i+j+k \leq p, \tag{2.85}$$

where the subscript $m \in [\![1, N_{SP}]\!]$ has replaced the subscript (i,j) in 2D (respectively (i,j,k) in 3D) with any arbitrary bijection $m \equiv m(i,j)$ (respectively $m \equiv m(i,j,k)$). In Eq. (2.83), the coefficients $\bar{u}_m$ are given by:

$$\bar{u}_m = \sum_{j=1}^{N_{SP}} \hat{u}_j \; \left(\Phi_m(\boldsymbol{\xi}_j)\right)^{-1}, \tag{2.86}$$

where $\hat{u}_j$ is the solution value at the solution point $\boldsymbol{\xi}_j$. The polynomial approximation $\hat{u}_h$ of the solution u is thus defined by:

$$\hat{u}_h(\boldsymbol{\xi}) = \sum_{m=1}^{N_{SP}} \sum_{j=1}^{N_{SP}} \hat{u}_j \; \left(\Phi_m(\boldsymbol{\xi}_j)\right)^{-1} \; \Phi_m(\boldsymbol{\xi}). \tag{2.87}$$

It is recalled that for a triangle, the number of SP needed to represent the solution as a polynomial of degree p is:

$$N_{SP}^{tri} = \frac{(p+1)(p+2)}{2}, \tag{2.88}$$

whereas for a tetrahedral, N_{SP} is given as:

$$N_{SP}^{tet} = \frac{(p+1)(p+2)(p+3)}{6}. \tag{2.89}$$

2.3.4.3. Solution Computation at Flux Points

To compute the flux values at FP (denoted $\boldsymbol{\xi}_k, k \in [\![1, N_{FP}^{RT}]\!]$), the solution values at those points need to be determined. With the polynomial distribution given by Eq. (2.87), the solution at FP can be computed as:

$$\hat{u}_h(\boldsymbol{\xi}_k) = \sum_{m=1}^{N_{SP}} \sum_{j=1}^{N_{SP}} \hat{u}_j \; \left(\Phi_m(\boldsymbol{\xi}_j)\right)^{-1} \; \Phi_m(\boldsymbol{\xi}_k) = \sum_{m=1}^{N_{SP}} \sum_{j=1}^{N_{SP}} \hat{u}_j \; \left(\mathcal{V}_{j,m}\right)^{-1} \Phi_m(\boldsymbol{\xi}_k). \tag{2.90}$$

Numerically, the extrapolation step is represented by the transfer matrix $\mathbf{T}_{kj} = \left[\left(\mathcal{V}_{j,m}\right)^{-1} \Phi_m(\boldsymbol{\xi}_k)\right]$.

2.3.4.4. Definition of the Flux Polynomial from the Set of Flux Values at Flux Points

Now that solution values at FP are known, the flux values $\hat{f}_k$ at the k-th flux point are assumed to be computed. The details of the flux values computation will be given below. First, the flux vector is approximated in the reference domain by $\hat{\mathbf{f}}_h$ in the RT space as:

$$\hat{\mathbf{f}}_h(\boldsymbol{\xi}) = \sum_{k=1}^{N_{FP}^{RT}} \hat{f}_k \boldsymbol{\psi}_k(\boldsymbol{\xi}), \tag{2.91}$$

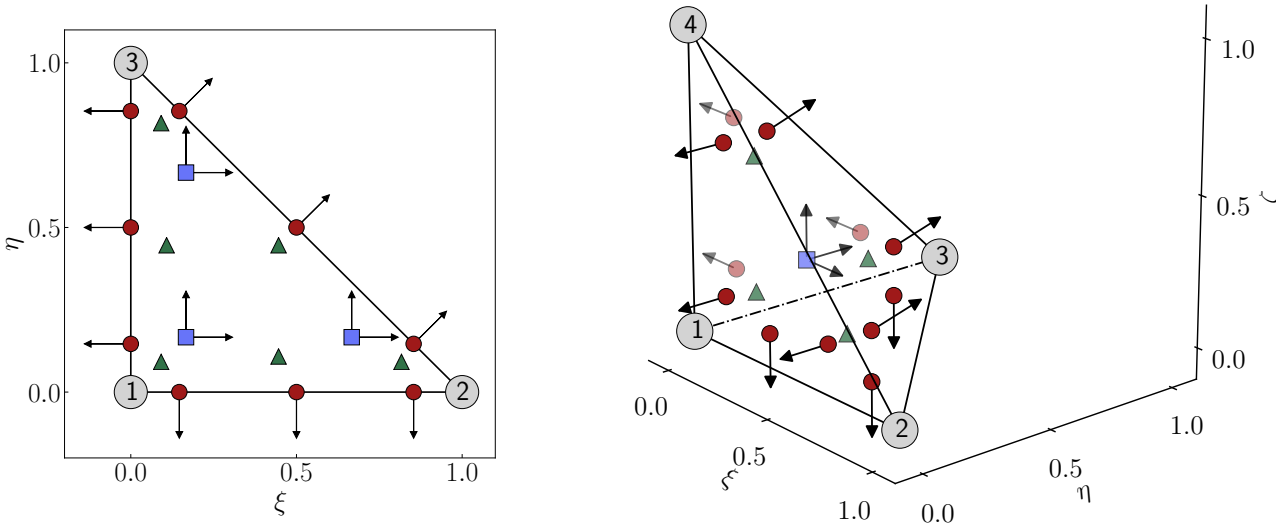
where N_{FP}^{RT} is the number of DoF needed to represent a vector-valued function in the RT_p space, defined as:

$$N_{FP}^{RT} = N_{FP}^{tri} = (p+1)(p+3) \quad \text{(2D)}, \qquad N_{FP}^{RT} = N_{FP}^{tet} = \frac{1}{2}(p+1)(p+2)(p+4) \quad \text{(3D)}, \tag{2.92}$$

and $\boldsymbol{\psi}_k$ are interpolation functions which form a basis in the RT space with the property:

$$\boldsymbol{\psi}_j(\boldsymbol{\xi}_k) \cdot \hat{\mathbf{n}}_k = \delta_{jk}, \tag{2.93}$$

where δ is the Kronecker symbol and $\hat{\mathbf{n}}_k$ is the outward unit normal vectors defined at the flux point $\boldsymbol{\xi}_k$. At this level, it must be highlighted that some flux points will be located inside the element and the definition of the normal vector needs to be described accurately. For interior FP, one physical point is associated with several DoF through the definition of unit vectors in different directions. In 2D (respectively 3D), one physical interior flux point is associated with two (respectively three) degrees of freedom whose normal vectors are $\hat{\mathbf{n}} = (1,0)^\top$ and $\hat{\mathbf{n}} = (0,1)^\top$ (respectively $\hat{\mathbf{n}} = (1,0,0)^\top$, $\hat{\mathbf{n}} = (0,1,0)^\top$ and $\hat{\mathbf{n}} = (0,0,1)^\top$) in the reference element. An example of SP and FP distributions and their associated degrees of freedom on the reference triangle and on the reference tetrahedron is given in Fig. 2.2.



(a) Reference triangle, $p = 2$

(b) Reference tetrahedron, $p = 1$

Figure 2.2. – Points distribution and associated DoF in the triangular and tetrahedal reference elements: SP (▲), interior FP (■), edge FP (●)

In Eq. (2.91), the vector-valued interpolation basis functions $\boldsymbol{\psi}_k$ can be expressed from the known monomial basis defined in section 2.3.3. To do so, the first step is to express the known monomial basis $\boldsymbol{\phi}_n$, $n \in [\![1, N_{FP}^{RT}]\!]$ in the RT space defined as a linear combination of the basis functions $\boldsymbol{\psi}_k$:

$$\boldsymbol{\phi}_n(\boldsymbol{\xi}) = \sum_{k=1}^{N_{FP}^{RT}} a_{n,k} \boldsymbol{\psi}_k(\boldsymbol{\xi}). \tag{2.94}$$

To determine the unknown coefficients $a_{n,k}$, Eq. (2.94) is multiplied by $\hat{\mathbf{n}}_k$ and then by enforcing the condition given by Eq. (2.93):

$$\boldsymbol{\phi}_n(\boldsymbol{\xi}) \cdot \hat{\mathbf{n}}_k = a_{n,k} \boldsymbol{\psi}_k(\boldsymbol{\xi}) \cdot \hat{\mathbf{n}}_k, \tag{2.95}$$

and

$$\boldsymbol{\phi}_n(\boldsymbol{\xi}_k) \cdot \hat{\mathbf{n}}_k = \sum_{l=1}^{N_{FP}^{RT}} a_{n,k} \boldsymbol{\psi}_l(\boldsymbol{\xi}_k) \cdot \hat{\mathbf{n}}_k, \tag{2.96}$$

so

$$a_{n,k} = \boldsymbol{\phi}_n(\boldsymbol{\xi}_k) \cdot \hat{\mathbf{n}}_k. \tag{2.97}$$

The interpolation functions $\boldsymbol{\psi}_k$ can thus be determined as:

$$\boldsymbol{\psi}_k(\boldsymbol{\xi}) = \sum_{n=1}^{N_{FP}^{RT}} \left(\boldsymbol{\phi}_n(\boldsymbol{\xi}_k) \cdot \hat{\mathbf{n}}_k\right)^{-1} \boldsymbol{\phi}_n(\boldsymbol{\xi}). \tag{2.98}$$

2.3.4.5. Determination of the Scalar Flux Values

The last step is to determine the scalar flux values $\hat{f}_k$ at FP on which the polynomial approximation given by Eq. (2.91) relies on. In the case of a first-order partial differential equation as given by Eq. (2.80), the flux are only functions of the solution. For interior FP, the flux values in the reference domain, denoted $\hat{f}_k$, are computed directly from the approximated solution value and projected on the unit normal vector previously defined. For FP located on edges, $\hat{f}_k$ is computed using a standard numerical flux function given as a solution of a Riemann problem using two extrapolated quantities, one on each side of the interface.

$$\hat{f}_k = \begin{cases} \hat{\mathbf{f}} \cdot \hat{\mathbf{n}}_k = |J| J^{-1} \mathbf{f}_k(u_h(\boldsymbol{\xi}_k)) \cdot \hat{\mathbf{n}}_k, & \boldsymbol{\xi}_k \in \mathcal{T} \setminus \partial\mathcal{T} \quad (\text{or} \in \mathcal{T}_e \setminus \partial\mathcal{T}_e), \\ \left(\hat{\mathbf{f}}_k \cdot \hat{\mathbf{n}}_k\right)^* = \left(\mathbf{f}_k \cdot |J| (J^{-1})^\top \hat{\mathbf{n}}_k\right)^*, & \boldsymbol{\xi}_k \in \partial\mathcal{T} \quad (\text{or} \in \partial\mathcal{T}_e), \end{cases} \tag{2.99}$$

where $\left(\hat{\mathbf{f}}_k \cdot \hat{\mathbf{n}}_k\right)^*$ is the standard numerical flux in the reference element and $u_h(\boldsymbol{\xi}_k) = \dfrac{1}{|J|} \hat{u}_h(\boldsymbol{\xi}_k)$ is the approximated solution in the physical domain. The flux vector approximation $\hat{\mathbf{f}}_h$ is finally defined by:

$$\hat{\mathbf{f}}_h(\boldsymbol{\xi}) = \sum_{k=1}^{N_{FP}^{RT}} \hat{f}_k \left(\boldsymbol{\phi}_n(\boldsymbol{\xi}_k) \cdot \hat{\mathbf{n}}_k\right)^{-1} \boldsymbol{\phi}_n(\boldsymbol{\xi}). \tag{2.100}$$

2.3.4.6. Differentiation of the Flux Polynomial at Solution Points

Once the flux vector is approximated on the reference element by Eq. (2.100), it can be differentiated at SP:

$$\begin{aligned} \hat{\nabla} \cdot \hat{\mathbf{f}}(u) &= \left(\hat{\nabla} \cdot \hat{\mathbf{f}}_h\right)(\boldsymbol{\xi}_j) \\ &= \hat{f}_k \left(\hat{\nabla} \cdot \boldsymbol{\psi}_k\right)(\boldsymbol{\xi}_j) \\ &= \hat{f}_k \, \hat{\nabla} \cdot \left(\sum_{n=1}^{N_{FP}^{RT}} \left(\boldsymbol{\phi}_n(\boldsymbol{\xi}_k) \cdot \hat{\mathbf{n}}_k\right)^{-1} \boldsymbol{\phi}_n(\boldsymbol{\xi}_j) \right) \\ &= \hat{f}_k \sum_{n=1}^{N_{FP}^{RT}} \left(\boldsymbol{\phi}_n(\boldsymbol{\xi}_k) \cdot \hat{\mathbf{n}}_k\right)^{-1} \; \hat{\nabla} \cdot \boldsymbol{\phi}_n(\boldsymbol{\xi}_j). \end{aligned} \tag{2.101}$$

The term $\left(\hat{\nabla}\cdot\boldsymbol{\psi}_k\right)(\boldsymbol{\xi}_j)$ in Eq. (2.101) can be written as a matrix of size $[N_{SP}\times N_{FP}^{RT}]$ called differentiation matrix and computed as:

$$\mathbf{D}_{jk}=\left[\left(\hat{\nabla}\cdot\boldsymbol{\psi}_k\right)(\boldsymbol{\xi}_j)\right]=\left[(\boldsymbol{\phi}_n(\boldsymbol{\xi}_k)\cdot\hat{\mathbf{n}}_k)^{-1}\,\hat{\nabla}\cdot\boldsymbol{\phi}_n(\boldsymbol{\xi}_j)\right]. \tag{2.102}$$

2.3.4.7. Semi-Discretized Form

The final form of the Spectral Difference scheme using Raviart-Thomas element discretization can be written for each degree of freedom of the solution function in each cell i as:

$$\frac{d\hat{u}_j^{(i)}}{dt}+\sum_{k=1}^{N_{FP}^{RT}}\hat{f}_k^{(i)}\left(\hat{\nabla}\cdot\boldsymbol{\psi}_k\right)(\boldsymbol{\xi}_j)=0,\quad j\in[\![1,N_{SP}]\!],\quad i\in[\![1,N]\!], \tag{2.103}$$

and the solution can be time-integrated using any standard time integration scheme (Runge-Kutta scheme for instance).

2.3.5. Comparison Between Spectral Difference and Flux Reconstruction Methods for Simplex Elements

The FR/CPR technique was introduced as a way to recover SD, DG and other schemes for any linear hyperbolic equation. But an open question concerns possible differences between the SDRT technique and the FR/CPR scheme. Let us consider the FR/CPR method described by Castonguay and Williams in their respective Ph.D. thesis [99, 158].

- Triangular element
  - FR/CPR method: The flux polynomial definition involves $\frac{(p+1)(p+2)}{2}$ SP (located inside the element) and $(p+1)$ FP located on each edge.
  - SDRT method: The flux polynomial relies on $(p+1)(p+3)$ FP, including $(p+1)$ FP on each edge. The number of FP located inside the element is thus $p(p+1)$.
  - Methods will differ if:

$$\frac{(p+1)(p+2)}{2}\neq p(p+1)\Longrightarrow p\neq 2 \text{ and } p\neq -1. \tag{2.104}$$

- Tetrahedral element
  - FR/CPR method: The flux polynomial definition requires $\frac{(p+1)(p+2)(p+3)}{6}$ SP (located inside the element) and $\frac{(p+1)(p+2)}{2}$ FP located on each edge.
  - SDRT method: The flux polynomial is build from $\frac{(p+1)(p+2)(p+4)}{2}$ FP, including $\frac{(p+1)(p+2)}{2}$ FP on each edge. The number of FP located inside the element is thus $\frac{p(p+1)(p+2)}{2}$.
  - Methods will differ if:

$$\frac{(p+1)(p+2)(p+3)}{6}\neq\frac{p(p+1)(p+2)}{2}\Longrightarrow p\neq -2, p\neq -1 \text{ and } p\neq\frac{3}{2}. \tag{2.105}$$

The present analysis to build a link between SDRT and FR flux polynomial computation is valid for any hyperbolic equation. For the linear advection equation, a connection could be established due to

the linear relation between the solution and the flux, as performed by Jameson [83]. The definition of this link is out of the scope of the present thesis.

In this section, mathematical foundations of the SDRT approach introduced by Balan *et al.* [142] were detailed. The SDRT method was described on triangles and extended to the treatment of tetrahedral elements. Recalling the standard SD approach and the SDRT scheme formulations now allows us to establish the SD/SDRT formulation on prismatic elements.

2.4. Prismatic Elements Formulation

The Spectral Difference method can be easily formulated on prismatic elements by considering the tensor product between the SDRT scheme on a triangle and the standard SD scheme in 1D. Let us consider the 3D scalar conservation law under its differential form:

$$\frac{\partial u(\mathbf{x},t)}{\partial t} + \nabla \cdot \mathbf{f}(u) = 0, \quad \text{in } \Omega \times [0, t_f], \tag{2.106}$$

where u is the state variable, ∇ is the differential operator in the physical domain and $\mathbf{f} = (f, g, h)$ where f, g and h are flux densities in the x, y and z directions respectively and $\mathbf{x} = (x, y, z)$. The computational domain Ω is discretized into N non-overlapping prismatic cells and the i-th element is denoted Ω_i:

$$\Omega = \bigcup_{i=1}^{N} \Omega_i. \tag{2.107}$$

2.4.1. Reference Element

For implementation simplicity, Eq. (2.106) is solved in the reference domain. Each cell Ω_i of the domain Ω is mapped into a reference prismatic element:

$$\mathcal{P} := \{(\xi, \eta) : 0 \leq \xi, \eta \leq 1, \xi + \eta \leq 1\} \times \{\zeta : 0 \leq \zeta \leq 1\}. \tag{2.108}$$

The transformation is written as:

$$\mathbf{x} = \sum_{i=1}^{n} M_i(\boldsymbol{\xi}) \mathbf{x}_i, \tag{2.109}$$

where $\boldsymbol{x_i}$ are the Cartesian coordinates in the physical domain of the $n = 6$ vertices of the prism and $M_i(\boldsymbol{\xi})$ are the shape functions. The shape functions and the Jacobian matrix expressions are given in Appendix A for the linear prismatic element (A.4.2). By introducing the transformed variables in the reference domain $\hat{u} = |J| \cdot u$ and $\hat{\mathbf{f}} = |J| J^{-1} \cdot \mathbf{f}$, Eq. (2.106) becomes:

$$\frac{\partial \hat{u}(\boldsymbol{\xi},t)}{\partial t} + \hat{\nabla} \cdot \hat{\mathbf{f}} = 0, \tag{2.110}$$

where $\hat{\nabla}$ is the differential operator in the reference domain.

2.4.2. Solution Approximation

The solution $\hat{u}$ is approximated as a polynomial of degree p using the tensor product between the 2D expansion using PKD basis and the standard 1D Lagrangian interpolation from its values at a set of N_{SP}^{pri} SP defined as:

$$N_{SP}^{pri} = N_{SP}^{tri}\, N_{SP}^{1d} = \frac{1}{2}(p+1)^2 (p+2). \tag{2.111}$$

To highlight the tensor product properties, the SP $\boldsymbol{\xi}_j, j \in [\![1, N_{SP}^{pri}]\!]$ are denoted:

$$\boldsymbol{\xi}_{(j_1,j_2)},\ j_1 \in [\![1, N_{SP}^{tri}]\!],\ j_2 \in [\![1, N_{SP}^{1d}]\!], \tag{2.112}$$

where j_1 indicates the position of the solution point on the triangle in the (ξ, η) plane and j_2 indicates the position in the ζ-direction. The polynomial approximation $\hat{u}_h$ of the solution $\hat{u}$ is defined by:

$$\hat{u}_h(\boldsymbol{\xi}) = \sum_{m=1}^{N_{SP}^{tri}} \sum_{j_2=1}^{N_{SP}^{1d}} \hat{u}_{j_1,j_2}\ (\Phi_m(\boldsymbol{\xi}_{j_1,j_2}))^{-1}\ \Phi_m(\boldsymbol{\xi})\, l_{j_2}(\boldsymbol{\xi}), \tag{2.113}$$

where $\hat{u}_{j_1,j_2}$ is the value of the solution $\hat{u}$ at the solution point $\boldsymbol{\xi}_{j_1,j_2}$ located at the position $(\xi_{j_1}, \eta_{j_1}, \zeta_{j_2})$.

2.4.3. Definition of the Flux Polynomial from the Set of Flux Values at Flux Points

The flux vector is approximated as the tensor product of the 2D approximation of $(\hat{f}, \hat{g})$ (the flux components in the (ξ, η) plane) in the RT space and the 1D Lagrangian approximation of the ζ-component $\hat{h}$. The two polynomial approximations are done independently on separate sets of FP. The 2D approximation is performed on a first set of FP denoted:

$$\boldsymbol{\xi}_{(k_1,j_2)}, \quad k_1 \in [\![1, N_{FP}^{tri}]\!], \quad j_2 \in [\![1, N_{SP}^{1d}]\!]. \tag{2.114}$$

The solution values at this first set of points (k_1, j_2) are computed using:

$$\hat{u}_h(\boldsymbol{\xi}_{(k_1,j_2)}) = \sum_{m=1}^{N_{SP}^{tri}} \hat{u}_{j_1,j_2} \ (\Phi_m(\boldsymbol{\xi}_{j_1,j_2}))^{-1} \ \Phi_m(\boldsymbol{\xi}_{(k_1,j_2)}), \tag{2.115}$$

and the 2D flux approximation reads:

$$\hat{\mathbf{f}}_h^{2D}(\boldsymbol{\xi}) = \sum_{j_2=1}^{N_{SP}^{1D}} \sum_{k_1=1}^{N_{FP}^{tri}} \hat{f}_{(k_1,j_2)} \boldsymbol{\psi}_{(k_1,j_2)}(\boldsymbol{\xi}), \tag{2.116}$$

with the flux values:

$$\hat{f}_{(k_1,j_2)} = \begin{cases} \hat{\mathbf{f}}\left(\hat{u}_h\left(\boldsymbol{\xi}_{(k_1,j_2)}\right)\right) \cdot \hat{\mathbf{n}}_{(k_1,j_2)}, & \boldsymbol{\xi}_{(k_1,j_2)} \in \mathcal{P} \setminus \partial\mathcal{P}, \\ \left(\mathbf{f}_{(k_1,j_2)} \cdot |J|(J^{-1})^\top \hat{\mathbf{n}}_{(k_1,j_2)}\right)^*, & \boldsymbol{\xi}_{(k_1,j_2)} \in \partial\mathcal{P}. \end{cases} \tag{2.117}$$

The 1D Lagrangian approximation is performed on a second set of FP denoted :

$$\boldsymbol{\xi}_{(j_1,k_2)} \quad j_1 \in [\![N_{SP}^{tri}]\!], \quad k_2 \in [\![1, N_{FP}^{1d}]\!]. \tag{2.118}$$

The solution values at this second set of points (j_1, k_2) are computed using:

$$\hat{u}_h(\boldsymbol{\xi}_{(j_1,k_2)}) = \sum_{j_2=1}^{N_{SP}^{1d}} \hat{u}_{j_1,j_2} \, l_{j_2}(\boldsymbol{\xi}_{(j_1,k_2)}), \tag{2.119}$$

and the 1D flux approximation reads:

$$\hat{f}_h^{1D}(\xi) = \sum_{j_1=1}^{N_{SP}^{tri}} \sum_{k_2=1}^{N_{FP}^{1d}} \hat{f}_{(j_1,k_2)} \, l_{k_2}(\xi), \tag{2.120}$$

with the flux values:

$$\hat{f}_{(j_1,k_2)} = \begin{cases} \hat{f}(\hat{u}_h(\xi_{(j_1,k_2)})), & \boldsymbol{\xi}_{(j_1,k_2)} \in \mathcal{P} \setminus \partial\mathcal{P}, \\ \hat{f}^*_{(j_1,k_2)}, & \boldsymbol{\xi}_{(j_1,k_2)} \in \partial\mathcal{P}. \end{cases} \tag{2.121}$$

The SP, the two sets of FP points and the associated DoF are represented in Fig. 2.3

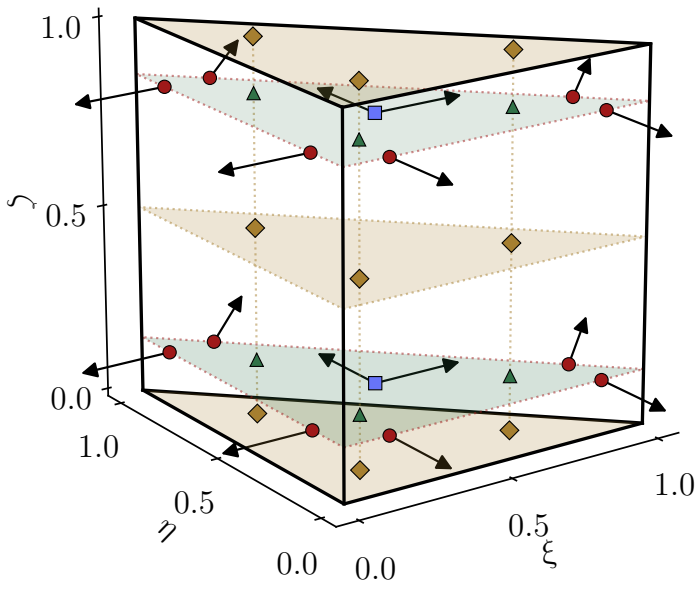


Figure 2.3. – Points distribution in the prismatic reference element for $p = 1$: SP $\boldsymbol{\xi}_{(j_1,j_2)}$ (▲), FP in the (ξ, η) plane $\boldsymbol{\xi}_{(k_1,j_2)}$ (edge: ●, interior: ■), FP in the ζ-direction $\boldsymbol{\xi}_{(j_1,k_2)}$ (◆)

2.4.4. Differentiation of the Flux Polynomial at Solution Points

Once the flux vector is approximated on each set of FP in the reference element by Eq. (2.116) and Eq. (2.120), it can be differentiated at SP:

$$\begin{aligned}\hat{\nabla} \cdot \left(\hat{\mathbf{f}}(u)\right) &= \left(\hat{\nabla} \cdot \hat{\mathbf{f}}_h^{2D}\right)(\boldsymbol{\xi}_{j_1,j_2}) + \left(\hat{\nabla} \cdot \hat{f}_h^{1D}\right)(\boldsymbol{\xi}_{j_1,j_2}) \\ &= \hat{f}_{(k_1,j_2)} \left(\hat{\nabla} \cdot \boldsymbol{\psi}_{(k_1,j_2)}\right)(\boldsymbol{\xi}_{j_1,j_2}) + \hat{f}_{(j_1,k_2)} \cdot \frac{\partial l_{k_2}}{\partial \zeta}(\xi_{(j_1,j_2)}).\end{aligned} \tag{2.122}$$

2.4.5. Semi-Discretized Form

The final form of the Spectral Difference scheme on prismatic elements can be written for each degree of freedom of the solution function in each cell i as:

$$\begin{aligned}\frac{du_j^{(i)}}{dt} + \left(\hat{f}^{(i)}_{(k_1,j_2)} \left(\hat{\nabla} \cdot \boldsymbol{\psi}_{(k_1,j_2)}\right)(\boldsymbol{\xi}_{j_1,j_2}) + \hat{f}^{(i)}_{(j_1,k_2)} \cdot \frac{\partial l_{k_2}}{\partial \zeta}(\xi_{(j_1,j_2)})\right) = 0, \\ j_1 \in [\![1, N_{SP}^{tri}]\!], \quad j_2 \in [\![1, N_{SP}^{1d}]\!], \quad k_1 \in [\![1, N_{FP}^{tri}]\!], \quad k_2 \in [\![1, N_{FP}^{1d}]\!], \quad i \in [\![1, N]\!],\end{aligned} \tag{2.123}$$

and the solution can be time-integrated using any standard time integration scheme (Runge-Kutta scheme for instance).

2.5. Pyramidal Elements

Like prismatic elements, pyramids are mainly used to link tetrahedral and hexahedral elements. However, contrary to other 3D simplices, it was demonstrated that in addition to polynomial functions, the pyramid approximation space necessary contains rational functions [159, 160].

In 2010, Bergot *et al.* [161] proposed an L_2 orthogonal basis on the pyramid for second-order and higher pyramid elements leading to stable and efficient solutions. Their application to Discontinuous Galerkin was studied in [162] and validated through numerical experiments. This basis has then been used by Chan and Warburton [163] to optimize the nodes distribution in the pyramid: on a pyramid triangular face, points are located as the same position as on a tetrahedron face, whereas on a pyramid quadrilateral face, points are located as the same position as on a hexahedron face. Validation for a GPU-accelerated time-domain DG method using hybrid grids composed of hexahedral, tetrahedral, prismatic and pyramidal elements was done in [164]. Additionally, Chan and Warburton proposed to split a pyramidal element into two tetrahedra and study different points distribution in [165].

The Flux Reconstruction approach was implemented for the treatment of pyramids in the solver PyFR [166]. However, in 2013, the FR stability was not proven for prisms and pyramids [99]. In 2020, a spectral analysis was conducted by Pereira and Vermeire in [167] for hexahedral, tetrahedral and prismatic elements. To the author's knowledge, such an analysis was conducted on pyramids neither for the FR nor for the SD method. Pyramidal elements have not been implemented during this thesis but stability properties should be investigated in the future. If stability results were not conclusive, the approach of splitting a pyramidal element into two tetrahedra should be preferred.

Although prismatic and pyramidal elements remain important to link tetrahedral and hexahedral cells, it is chosen in this thesis to focus on the SDRT stability properties and implementation on triangles and tetrahedra. This choice is motivated by three reasons:

- Since the Spectral Difference formulation on prismatic elements simply follows the tensor product between a 1D standard SD and a 2D SDRT scheme, the investigation of the 2D SDRT scheme stability properties should first be considered. Once a stable 2D SDRT formulation is proposed, the extension to prismatic elements does not seem to raise any specific difficulties.
- On the other hand, the treatment of pyramidal elements is more complex and not as well studied in the literature as the treatment of tetrahedra.
- Finally, using a grid fully composed of tetrahedral elements allows one to deal with complex geometries. A full prismatic or pyramidal grid does not offer as much flexibility. The possibility to run 3D computations on tetrahedral grids is thus preferred.

Chapter 3

Linear Stability Analysis for the Spatial Discretization

In this chapter, the stability of the SD and SDRT spatial discretizations is investigated using the advection equation. First, the general eigenvalues analysis is presented. This analysis consists in studying the eigenvalues spectrum of the spatial operator, which is written under a matrix form. The eigenvalues analysis is applied to two methods to determine the stability: the common Fourier analysis and a new procedure, which considers a domain composed of a fixed number of cells (FNC method) without assuming the solution as a harmonic place wave.
The FNC approach is introduced to reduce the computational time needed to compute the spatial operator matrix, leading to the possibility to study multiple SDRT implementations depending on the choice of the interior FP location. It is first applied to the standard 1D SD scheme to verify its consistency with Fourier analysis results. It is shown that the FNC approach can recover the common 1D SD stability results regarding the interior FP location.
The SDRT linear stability is then studied on triangular grids using both approaches. The SDRT stability is established up to the fourth-order using the Fourier analysis. Employing the FNC method, sets of interior FP leading to a stable SDRT scheme for $p > 3$ on two triangles are determined. The method limitations are then shown by increasing the number of triangles in the domain.
Finally, the Fourier analysis of the SDRT is eventually conducted on tetrahedral elements.

Overview

3.1. Stability Analysis Methods **42**
3.1.1. Eigenvalue Analysis 42
3.1.2. Fourier Stability Analysis 45
3.1.3. Stability Analysis on a Fixed Number of Cells 46
3.2. Standard One-Dimensional Spectral Difference Scheme **47**
3.2.1. Matrix form 47
3.2.2. Stability Analysis on a Fixed Number of Cells 48
3.3. Spectral Difference Using Raviart-Thomas Elements Scheme on Triangular Elements **49**
3.3.1. Flux Points Numbering 49
3.3.2. Interior Flux Points Locations Based on Quadrature Rules 49
3.3.3. Fourier Stability Analysis 50
3.3.4. Stability Analysis on a Fixed Number of Cells 53
3.4. Spectral Difference Scheme Using Raviart-Thomas Elements on Tetrahedral Elements **59**
3.4.1. Reference Element Convention and Points Numbering 59
3.4.2. Choice of Solution Points and Flux Points Location 61
3.4.3. Matrix Form 61
3.4.4. Fourier Stability Analysis 63

3.1. Stability Analysis Methods

3.1.1. Eigenvalue Analysis

Let us consider the linear advection equation

$$\frac{\partial u(\mathbf{x},t)}{\partial t} + \nabla \cdot \mathbf{f} = 0, \quad \text{in } \Omega \times [0, t_f], \tag{3.1}$$

within a domain Ω, where u is a conserved scalar quantity and $\mathbf{f} = \mathbf{c} \cdot u$ is the flux. The velocity vector $\mathbf{c}$ is defined by:

$$\mathbf{c} = \begin{cases} c_x, \quad c_x \in [-1, 1] & \text{(1D)}, \\ (c_x, c_y) = (\cos\theta, \sin\theta)^\top, \quad \theta \in [0, 2\pi] & \text{(2D)}, \\ (c_x, c_y, c_z) = (\sin\theta_2 \cos\theta_1, \sin\theta_2 \sin\theta_1, \cos\theta_2)^\top, \quad (\theta_1, \theta_2) \in [0, 2\pi]^2 & \text{(3D)}, \end{cases} \tag{3.2}$$

and the domain Ω is subdivided into N non-overlapping elements $\Omega_i, i \in [\![1, N]\!]$.

3.1.1.1. Matrix Form of the Spatial Operator

To analyze a scheme stability using eigenvalue analysis, the spatial discretization of Eq. (3.1) is formulated under a matrix form. To do so, the column vector $\hat{\mathbf{U}}_j^i$ of size N_{SP} whose components are the solution values in the reference element at the j-th SP in the i-th cell is introduced:

$$\hat{\mathbf{U}}_j^i = [\hat{u}_i(\boldsymbol{\xi}_j)]_{1 \leq j \leq N_{SP}}. \tag{3.3}$$

The solution values at FP are obtained by multiplying $\hat{\mathbf{U}}_j^i$ by the transfer matrix $\mathbf{T}_{kj}$, which represents the extrapolation step, leading to the column vector $\hat{\mathbf{V}}_k^i$ of size N_{FP}:

$$\hat{\mathbf{V}}_k^i = [\hat{u}_i(\boldsymbol{\xi}_k)]_{1 \leq k \leq N_{FP}} = \mathbf{T}_{kj} \hat{\mathbf{U}}_j^i. \tag{3.4}$$

From the solution values at FP, the flux values are then computed. At cell interfaces, two different solution values are available at the same point. An upwind Godunov scheme is used as the numerical flux to ensure the flux continuity. For the standard SD scheme, the flux vector values read:

$$\hat{\mathbf{f}}_k = \begin{cases} \left(|J|J^{-1} \cdot \mathbf{c}\right) \, u_i(\boldsymbol{\xi}_k), \quad \boldsymbol{\xi}_k \in \Omega \setminus \partial\Omega, \\ \left(\mathbf{c} \cdot \mathbf{n}_k^i\right) \left(\dfrac{1 + \text{sign}(\mathbf{c} \cdot \mathbf{n}_k^i)}{2} u_i(\boldsymbol{\xi}_k) + \dfrac{1 - \text{sign}(\mathbf{c} \cdot \mathbf{n}_k^i)}{2} u_{nei}(\boldsymbol{\xi}_k) \right), \quad \boldsymbol{\xi}_k \in \partial\Omega, \end{cases} \tag{3.5}$$

whereas for a SDRT scheme, the flux scalar values read:

$$\hat{f}_k = \begin{cases} |J|J^{-1} \left(\mathbf{c} \cdot \hat{\mathbf{n}}_k^i\right) \, u_i(\boldsymbol{\xi}_k), \quad \boldsymbol{\xi}_k \in \Omega \setminus \partial\Omega, \\ \left(\mathbf{c} \cdot \mathbf{n}_k^i\right) \left(\dfrac{1 + \text{sign}(\mathbf{c} \cdot \mathbf{n}_k^i)}{2} u_i(\boldsymbol{\xi}_k) + \dfrac{1 - \text{sign}(\mathbf{c} \cdot \mathbf{n}_k^i)}{2} u_{nei}(\boldsymbol{\xi}_k) \right), \quad \boldsymbol{\xi}_k \in \partial\Omega, \end{cases} \tag{3.6}$$

where $u_{nei}(\boldsymbol{\xi}_k)$ is the solution value coming from the neighboring cell. Given that the flux is linear, the flux computation can be expressed as a matrix product vector between a velocity matrix denoted $\mathbf{C}$ and the vector $\hat{\mathbf{V}}_k^i$ containing the solution values at FP:

$$\hat{\mathbf{F}}_k^i = \mathbf{C} \cdot \hat{\mathbf{V}}_k^i. \tag{3.7}$$

Since solving a Riemann problem implies two states, the velocity matrix $\mathbf{C}$ has to associate to each FP located at the cell interface of the i-th cell the correct neighboring FP located on the adjacent cell. The matrix $\mathbf{C}$ actually accounts for the mesh connectivity, thus its size is $[N_{FP} \times N, N_{FP} \times N]$. Its analytical expression depends on the considered computational domain and will be given case-by-case. The vector $\hat{\mathbf{F}}_k^i$, containing the flux values at FP, is eventually multiplied by the derivation matrix $\mathbf{D}_{jk}$, leading to the flux derivative values at SP. The semi-discrete scheme under a matrix form on the domain Ω is finally:

$$\begin{bmatrix} \dfrac{\partial \hat{\mathbf{U}}_j^1(t)}{\partial t} \\ \vdots \\ \dfrac{\partial \hat{\mathbf{U}}_j^i(t)}{\partial t} \\ \vdots \\ \dfrac{\partial \hat{\mathbf{U}}_j^N(t)}{\partial t} \end{bmatrix} + \begin{bmatrix} \mathbf{D}_{jk} & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{D}_{jk} & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & \mathbf{D}_{jk} \end{bmatrix} \mathbf{C} \begin{bmatrix} \mathbf{T}_{kj} & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{T}_{kj} & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & \mathbf{T}_{kj} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{U}}_j^1(t) \\ \vdots \\ \hat{\mathbf{U}}_j^i(t) \\ \vdots \\ \hat{\mathbf{U}}_j^N(t) \end{bmatrix} = 0,$$

$$i \in [\![1, N]\!], j \in [\![1, N_{SP}]\!], k \in [\![1, N_{FP}]\!]. \quad (3.8)$$

The overall spatial discretization (extrapolation, flux computation, differentiation) on the computational domain can be written in a compact matrix form:

$$\frac{\partial \hat{\mathbf{U}}^i(t)}{\partial t} = \mathbf{M}\ \hat{\mathbf{U}}^i(t), \quad (3.9)$$

where

$$\mathbf{M} = -\,\mathrm{bdiag}(\mathbf{D})\ \mathbf{C}\ \mathrm{bdiag}(\mathbf{T}). \quad (3.10)$$

In Eq. (3.10), the operator bdiag() applied to an arbitrary square matrix $\mathbf{A}$ gives a block diagonal matrix such that the main-diagonal blocks are the square matrix $\mathbf{A}$ and all off-diagonal blocks are zeros matrices. The stability of the spatial discretization can be investigated through an eigenvalues analysis using Eq. (3.9). The spectrum of the SD (or SDRT) spatial discretization is obtained by computing eigenvalues of the spatial operator matrix $\mathbf{M}$. The eigenvalues of $\mathbf{M}$, denoted $\lambda_{\mathbf{M}}$, form a spectrum $\rho_{\mathbf{M}}$ in the complex plane. For a given advection angle, there are N_{SP} eigenvalues per cell, leading to a spectrum composed of $N \times N_{SP}$ eigenvalues. If the real part of all eigenvalues of $\mathbf{M}$, denoted $\mathrm{Re}(\lambda_{\mathbf{M}})$, is non-positive, then the spatial discretization is stable.

3.1.1.2. Impact of the Different Parameters on the Stability

According to Eq. (3.10), the spatial operator matrix $\mathbf{M}$ depends on the transfer matrix $\mathbf{T}$, the velocity matrix $\mathbf{C}$ and the differentiation matrix $\mathbf{D}$:

- The *transfer matrix* is expressed from the polynomial basis (the Lagrangian basis for tensor product cells, the PKD basis for simplex cells) at SP and FP locations.
- The *velocity matrix* relies on the velocity vector $\mathbf{c}$ and the normal vector defined at each flux point.
- The *differentiation matrix* involves the polynomial basis in which the flux is differentiated (the Lagrangian basis for tensor product cells, the Raviart-Thomas basis for simplex cells) at SP and FP locations as well as on the normal vector defined at each flux point.

The velocity vector is set through the definition of c_x in 1D, or through the advection angle (θ in 2D, (θ_1, θ_2) in 3D) as described by Eq. (3.2). The linear stability will be studied for several values of the advection angle.

As presented in Chapter 2, for tensor product cells, the polynomial basis is the Lagrangian basis for both extrapolation and differentiation whereas for simplex cells, the PKD basis and the Raviart-Thomas basis are used for the extrapolation and the differentiation (respectively). Those polynomial bases rely on the SP and FP sets of points and the normal vector associated with each FP. Since it was shown by Van den Abeele *et al.* [132] that the SD scheme stability is independent of the SP position, our main concern is to find a set of FP leading to a stable SDRT scheme for all advection angles.

The FP location has a direct impact on the SD scheme stability. In 1D, it was shown by Van den Abeele [132] that if FP are chosen as the Chebyshev-Gauss-Lobatto nodes, the standard 1D SD scheme can be unstable for $p > 2$. Following this work, Jameson [83] has proven that the stability of the SD scheme for all orders of accuracy in the case of a 1D linear advection *'provided that the interior fluxes collocation points are placed at the zeros of the corresponding Legendre polynomial'*.

For triangular elements, it was observed by Balan *et al.* [141] that the placement of FP on edges does not affect the linear stability properties for second- to fourth-order accurate SDRT schemes. To simplify the 2D hybrid implementation, the position of FP located on the edge is set to the Gauss-Chebyshev points given by Eq. (2.12). FP on edges for a quadrilateral and a triangle are thus located at the same coordinates. By doing so, there is no need to apply mortar techniques as introduced by Kopriva [106]. This technique is useful when FPs between interfaces are not matching (e.g. p or h-refinement), and consists in a solution projection from both interfaces into an intermediate interface, called a mortar. The flux is uniquely defined on the mortar by solving a Riemann problem and is then projected back onto each face. However, the projection steps bring an additional cost, which can be easily avoided for hybrid grids by setting the position of FP located on the edge to the Gauss-Chebyshev points. In a same way, for the extension to 3D, FP located on a prismatic element square face (respectively triangular face) should be placed at the very same coordinates as FP located on an hexahedral element face (respectively tetrahedral element face). Since the edge (in 2D, or face in 3D) FP position is chosen to be fixed, the remaining unknown is the interior FP location. For a SDRT scheme, the number of interior FP is given by:

$$N_i^{tri} = p\,(p+1) \quad \text{(2D)}, \qquad N_i^{tet} = \frac{p\,(p+1)\,(p+2)}{2} \quad \text{(3D)}. \tag{3.11}$$

As mentioned in section 2.3.4.4, the number of physical interior points is reduced from N_i to N_i/d, where d is the dimension, by considering one physical point as d separate DoF with different 'normal' vectors $\hat{\mathbf{n}}$. Each physical interior flux point is associated with d normal vectors given as:

$$\hat{\mathbf{n}} = (1,0)^\top, \hat{\mathbf{n}} = (0,1)^\top \quad \text{(2D)}, \tag{3.12}$$

$$\hat{\mathbf{n}} = (1,0,0)^\top, \hat{\mathbf{n}} = (0,1,0)^\top, \hat{\mathbf{n}} = (0,0,1)^\top \quad \text{(3D)}. \tag{3.13}$$

The number of physical interior FP, denoted N_{pi}, is thus given as:

$$N_{pi}^{tri} = \frac{p\,(p+1)}{2} \quad \text{(2D)}, \qquad N_{pi}^{tet} = \frac{p\,(p+1)\,(p+2)}{6} \quad \text{(3D)}, \tag{3.14}$$

which correspond to the number of SP for a $\text{SDRT}_{(p-1)}$ scheme.

In the following, two different methods to analyze the scheme stability are introduced: the common Fourier stability analysis and a new approach, which considers a fixed number of cells rather than imposing a planar harmonic wave. For both methods, the stability is studied through eigenvalues analysis. However, the spatial operator matrix $\mathbf{M}$ is not defined in the same way for both methods. To avoid confusion, the matrix $\mathbf{M}$ associated with the Fourier stability analysis will be denoted $\mathbf{M}_\text{z}$, whereas the matrix $\mathbf{M}$ associated with the analysis on a Fixed Number of Cells (FNC) will be denoted $\mathbf{M}_\text{n}$.

3.1.2. Fourier Stability Analysis

The Fourier (or Von Neumann) method for stability analysis in 1D considers a computational domain of length L. The domain is divided into N cells of size $\Delta x = L/N$. By repeating the domain periodically, the solution can be expressed as a finite Fourier series on the domain $2L$:

$$u_i = \tilde{u}_j \exp(Ik_j x_i) = \tilde{u}_j \exp(Ik_j i\Delta x), \tag{3.15}$$

where $k_j = \dfrac{j\pi}{N\Delta x}$ is the wavenumber, $\tilde{u}_j$ is the amplitude of the j^{th} harmonic of u_i and I is the imaginary unit ($I^2 = -1$). The shortest and the largest wavelengths are respectively $l_{min} = 2\Delta x$ and $l_{max} = 2L$, leading to the minimum and maximum wavenumbers $k_{min} = \pi/L$ and $k_{max} = \pi/\Delta x$. Therefore, to represent all harmonics on a finite mesh, $k\Delta x$ should cover the range $[-\pi, \pi]$. The product $k\Delta x$ represents the grid frequency, denoted κ. The maximum value π corresponds to the frequency of the wavelength $2\Delta x$, which is the highest frequency resolvable on the mesh (Nyquist–Shannon sampling theorem).

For 2D and 3D cases, the analysis is performed by introducing the wave number vector $\mathbf{k}$:

$$\mathbf{k} = \begin{cases} k(\cos\vartheta, \sin\vartheta)^\top, & \vartheta \in [0, 2\pi] \quad \text{(2D)}, \\ \\ k\left(\cos\vartheta_1 \sin\vartheta_2, \sin\vartheta_1 \sin\vartheta_2, \cos\vartheta_2\right)^\top, & (\vartheta_1, \vartheta_2) \in [0, 2\pi]^2 \quad \text{(3D)}, \end{cases} \tag{3.16}$$

k being the wavenumber of the harmonic wave and ϑ (in 2D) and $(\vartheta_1, \vartheta_2)$ (in 3D) its orientation angles.

The spatial harmonic solution given by Eq. (3.15) is imposed in the matrix form of the spatial scheme, leading to a relation:

$$\frac{d\tilde{\mathbf{U}}^i(t)}{dt} = \mathbf{M}_\mathrm{z} \tilde{\mathbf{U}}^i(t), \tag{3.17}$$

where $\tilde{\mathbf{U}}^i$ is a complex vector composed of u_i harmonic amplitudes and $\mathbf{M}_\mathrm{z}$ depends on the advection velocity $\mathbf{c}$, the wavenumber k and the orientation angle (ϑ in 2D, $(\vartheta_1, \vartheta_2)$ in 3D). The stability of the spatial operator can be studied through the spectrum of $\mathbf{M}_\mathrm{z}$ using eigenvalues analysis.

As detailed in section 3.1.1.2, the stability of SD schemes seems to be strongly dependent on the interior FP locations. To determine stable SDRT schemes on triangles, the spectrum of $\mathbf{M}_\mathrm{z}$ should be studied for multiple FP locations. However, using Fourier analysis, the computation matrix $\mathbf{M}_\mathrm{z}$ is dependent on a large number of parameters: the spectrum of the spatial operator has to be computed for each advection angle, each grid frequency and, for 2D and 3D cases, each orientation angle. To reduce the number of parameters on which the spatial operator depends and thus the computational time needed to study each set of FP location, another stability analysis method is proposed.

3.1.3. Stability Analysis on a Fixed Number of Cells

As it was presented in the previous section, the Fourier method is the most common way to investigate linear stability. Additionally to the advection angle discretization, the computation of the spatial operator is dependent on the grid frequency. In this section, a different stability analysis - the Fixed Number of Cells (FNC) approach - is proposed to avoid the grid frequency dependency.

Rather than imposing a planar harmonic wave as the initial solution, no initial condition is specified and the periodic boundary conditions are imposed through the numerical flux. This approach requests to write the matrix form of the SDRT discretization through one global matrix $\mathbf{M}_{\mathrm{n}}$ of size $[N \times N_{SP}]$ where the value of N is fixed. By doing so, the spatial operator only depends on the advection angle and the computation of eigenvalues becomes less expensive. Moreover, it must be highlighted that using this formulation, the matrix $\mathbf{M}_{\mathrm{n}}$ does not depend on the initial solution. Consequently, if the real part of all eigenvalues of the spatial operator matrix is negative, i.e. $\mathrm{Re}(\lambda_{\mathbf{M}_{\mathrm{n}}}) \leq 0$, the linear stability of the SDRT scheme is ensured on the considered domain regardless of the initial solution. By considering a small domain, multiple interior FP locations can be tested in a reasonable computational time. Since the spatial stability of SD schemes seems to be strongly related to the interior FP location, whether for the standard SD scheme [83, 132] or the SDRT formulation [141], this approach could be a first step towards the definition of stable SDRT schemes.

However, stability results are only valid for the considered domain, and since the time required to compute the global matrix $\mathbf{M}_{\mathrm{n}}$ quickly increases with the number of cells, the approach is limited to small computational domains. One could assume that obtaining a stable scheme on a small domain is sufficient to ensure the scheme stability on any domain, but this hypothesis has to be assessed. To evaluate the capability of the FNC approach to give general stability results, it is first applied to the standard 1D scheme on computational domains of different sizes. The aim is to verify that results are consistent with the common stability conclusions about interior FP location.

3.2. Standard One-Dimensional Spectral Difference Scheme

3.2.1. Matrix form

Eq. (3.1) is solved on a 1D regular mesh composed of N cells denoted $S_i, i \in [\![1, N]\!]$ with periodic boundary conditions. Figure 3.1 show an example for the case $N = 3$. The velocity $c = c_x$ varies within $[-1, 1]$.

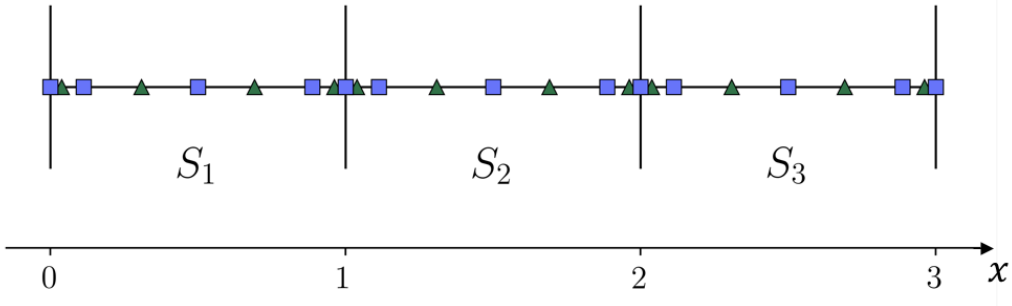


Figure 3.1. – Computational domain for the 1D FNC linear stability analysis - Example of SP (▲) and FP (■) distributions for $p = 3$, $N = 3$

The extrapolation matrix in Eq. (3.8) is given as:

$$\mathbf{T}_{kj} = l_j(\boldsymbol{\xi}_k). \tag{3.18}$$

Using the example of $N = 3$, the velocity matrix of size $[N \cdot (p + 2)]$ is:

$$\mathbf{C} = c \begin{bmatrix} \frac{1-\text{sign}(c)}{2} & O_{1,p} & \cdots & \cdots & \cdots & O_{1,p} & \frac{1+\text{sign}(c)}{2} \\ O_{p,1} & \mathbf{I}_{p,p} & \ddots & & & & O_{p,1} \\ \vdots & \ddots & \mathbf{S} & \ddots & & & \vdots \\ \vdots & & \ddots & \mathbf{I}_{p,p} & \ddots & & \vdots \\ \vdots & & & \ddots & \mathbf{S} & \ddots & \vdots \\ O_{p,1} & & & & \ddots & \mathbf{I}_{p,p} & O_{p,1} \\ \frac{1-\text{sign}(c)}{2} & O_{1,p} & \cdots & \cdots & \cdots & O_{1,p} & \frac{1+\text{sign}(c)}{2} \end{bmatrix}, \tag{3.19}$$

where

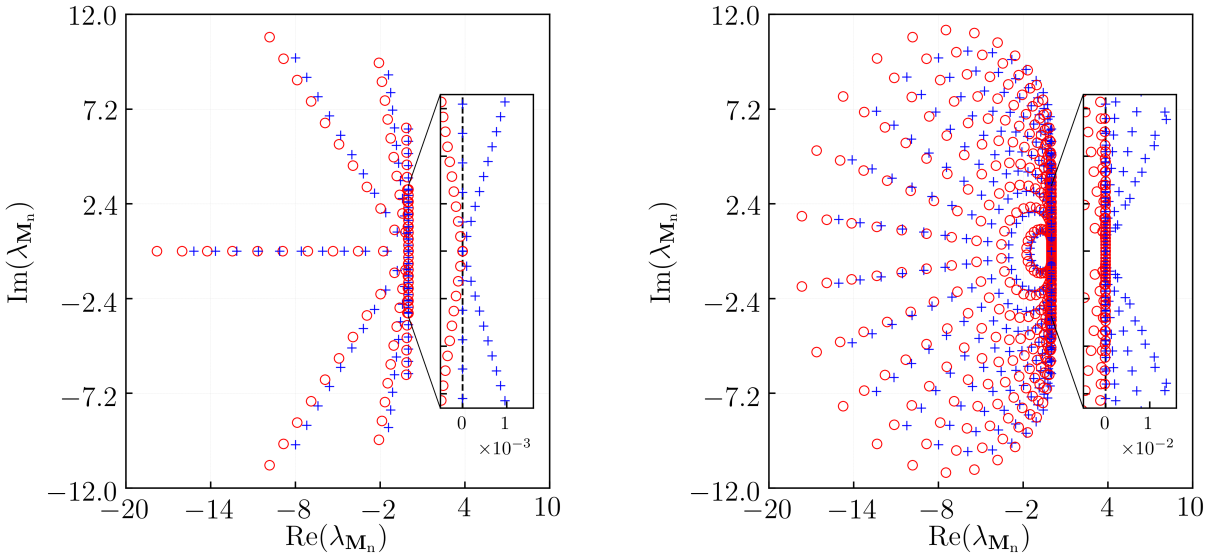
$$\mathbf{S} = \begin{bmatrix} \frac{1+\text{sign}(c)}{2} & \frac{1-\text{sign}(c)}{2} \\ \frac{1+\text{sign}(c)}{2} & \frac{1-\text{sign}(c)}{2} \end{bmatrix}, \tag{3.20}$$

and $\text{sign}(c) = c/|c|$. This pattern can easily be extended to N cells. The differentiation matrix is:

$$\mathbf{D}_{jk} = \frac{\partial l_k}{\partial \xi}(\boldsymbol{\xi}_j). \tag{3.21}$$

3.2.2. Stability Analysis on a Fixed Number of Cells

To verify that the stability results given by the FNC method are consistent for the 1D standard SD scheme, the first step is to recover the usual stability results regarding the influence of the interior FP location. For $p = 4$, the eigenvalues of the matrix $\mathbf{M}_\mathrm{n} = -\operatorname{diag}(\mathbf{D})\ \mathbf{C}\ \operatorname{diag}(\mathbf{T})$ for two possible FP locations (Chebyshev-Gauss-Lobatto and Legendre nodes) are plotted in Fig. 3.2 for a domain composed of 2 elements (left) and for a domain composed of 9 elements (right). Considering symmetry properties, the velocity c_x is reduced from $[-1, 1]$ to $[0, 1]$. A closer view at $\mathrm{Re}(\lambda_{\mathbf{M}_\mathrm{n}}) = 0$ is presented on the same figure. For both domains, as expected, the spectrum of $\mathbf{M}_\mathrm{n}$ obtained when using the Chebyshev-Gauss-Lobatto nodes as FP reveals positive values of $\mathrm{Re}(\lambda_{\mathbf{M}_\mathrm{n}})$, indicating that the SD scheme is unstable. The spectrum obtained with the Legendre nodes only presents negative values of $\mathrm{Re}(\lambda_{\mathbf{M}_\mathrm{n}})$, establishing the scheme stability. The discretization of the velocity clearly appears on each spectrum: eigenvalues form lines composed of 10 points, corresponding to 10 velocity values. The number of eigenvalues for each velocity value is given by $N(p + 1)$.



(a) $N = 2$ (b) $N = 9$

Figure 3.2. – Spectrum of the matrix $\mathbf{M}_\mathrm{n}$ for the standard 1D SD scheme, $p = 4$: Chebyshev-Gauss-Lobatto FP (+), and Legendre FP(○)

The FNC approach seems to be able to recover the standard 1D SD scheme stability results regarding the possible interior FP locations. This first study allows us to consider the FNC approach to study the SDRT scheme stability on triangles.
Before that, a complete Fourier stability analysis is presented for SDRT schemes based on a polynomial approximation of degree $p \in [\![2, 5]\!]$ using different interior FP locations based on quadrature rules. The FNC method is then used on two triangles to study multiple interior FP locations for orders of accuracy higher than four. The stability results of both analyses are finally compared by expanding the number of cells used in the FNC approach.

3.3. Spectral Difference Using Raviart-Thomas Elements Scheme on Triangular Elements

3.3.1. Flux Points Numbering

For clarity purposes, the FP numbering in the reference triangle needs to be settled and their normal vector defined. On each edge, there are $N_e^{tri} = (p+1)$ FP. Since this section is dedicated to triangular element, they will simply be denoted N_e. The FP located on edges are represented with red circles and numbered as follow:

- on face 1 ($\eta = 0$), $k \in [\![1, N_e]\!]$, k increasing with ξ, $\hat{\mathbf{n}} = (0, -1)^\top$,
- on face 2 ($\eta = 1 - \xi$), $k \in [\![N_e + 1, 2N_e]\!]$, k increasing with η, $\hat{\mathbf{n}} = (1, 1)^\top$,
- on face 3 ($\xi = 0$), $k \in [\![2N_e + 1, 3N_e]\!]$, k increasing when η decreases, $\hat{\mathbf{n}} = (-1, 0)^\top$.

The remaining $N_i^{tri} = p\,(p+1)$ FP, simply denoted N_i in this section, are located in the interior and represented with blue squares. Since one physical point is considered as two separate DoF with different normal vectors, there are $N_i/2$ physical FP. FP associated with the unit vector $\hat{\mathbf{n}} = (1, 0)^\top$ in the reference element are numbered with $k \in [\![3N_e + 1, 3N_e + 1 + N_i/2]\!]$ whereas FP whose unit vector is $\hat{\mathbf{n}} = (0, 1)^\top$ are numbered with $k \in [\![3N_e + 1 + N_i/2, 3N_e + 1 + N_i]\!]$. An example of the FP numbering and their associated normal vector is given in Fig. 3.3 for the case $p = 2$.

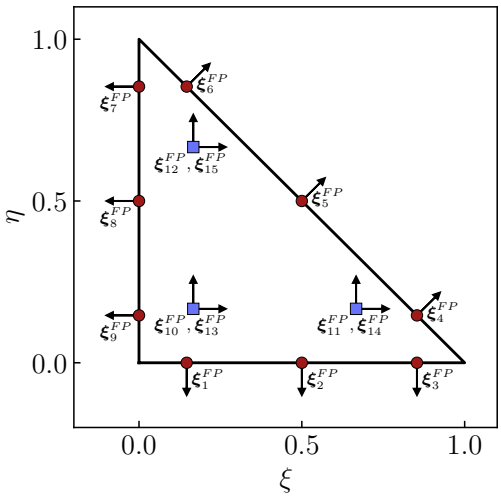


Figure 3.3. – FP numbering in the reference triangular element - Example of FP distribution for $p = 2$ (edge: ●, interior: ■)

3.3.2. Interior Flux Points Locations Based on Quadrature Rules

Now that the FP numbering is settled, the choice of the interior FP location can be addressed. It is recalled here that FP located on edges are placed at the Gauss-Chebyshev points given by Eq. (2.12) and that the number of physical interior FP is given by Eq. (3.14). The placement of interior FP leading to stable SDRT schemes was studied for $p \in [\![1, 3]\!]$ by May and Schöberl [143]. Their conclusions can be summarized as follow:

- For SDRT_1, the interior physical FP is placed at the triangle centroid;
- For SDRT_2, interior physical FP are placed according to the three-points quadrature rule of order 2. This quadrature rule was given by many authors [168–174] as the higher order three-points rules;

– For $SDRT_3$, interior physical FP are located at the six-point quadrature rule of order 4 given by [169–177].

Choosing a quadrature rule associated with the same number of points as the interior physical FP seems to be a promising choice. To be suitable, the quadrature rule should not include integration points located on edges or outside the triangle. Among the available literature, several appropriate quadrature rules are found for $p > 3$:

– For $SDRT_4$, the 10-points quadrature rules of order 5 of Vioreanu-Rokhlin [173] and Williams-Shunn-Jameson [174];

– For $SDRT_5$, the 15-points quadrature rule of order 7 of Williams-Shunn-Jameson [174], Witherden-Vincent [177], Xiao-Gimbutas [172], Vioreanu-Rokhlin [173], Papanicolopulos [178] and Laursen-Gellert [169].

– For $SDRT_6$, the 21-points quadrature rule of order 8 of Williams-Shunn-Jameson [174] and Vioreanu-Rokhlin [173] and the 21-points quadrature rule of order 9 of Laursen-Gellert [169]

3.3.3. Fourier Stability Analysis

In this section, the Fourier analysis presented by Castonguay in his Ph.D. thesis [158] for the FR method is adapted to the SDRT scheme and results are presented for $p \in [\![4, 6]\!]$.

Let us consider the 2D linear advection problem given by Eq. (3.1) on a square domain $\Omega = [0, L]^2$ with periodic boundary conditions. The domain Ω is meshed as a Cartesian mesh composed of $N_x \times N_y$ quadrilateral elements of size $\Delta x \times \Delta y$. The mesh is distorted using the skew angle μ. Each quadrilateral cell is then divided into two triangles, identified as $T_{i_1,i_2,1}$ and $T_{i_1,i_2,2}$, $i_1 \in [\![1, N_x]\!]$, $i_2 \in [\![1, N_y]\!]$ (Fig. 3.4). To properly define the mesh pattern, two vectors are introduced: $\mathbf{B}_1 = (\Delta x, 0)$ and $\mathbf{B}_2 = \Delta x(\cos\mu, \sin\mu)$. The mesh is made dimensionless using a scaling by the Cartesian mesh edge length Δx, leading to the dimensionless vectors $\hat{\mathbf{B}}_1 = (1, 0)$ and $\hat{\mathbf{B}}_2 = (\cos\mu, \sin\mu)$.

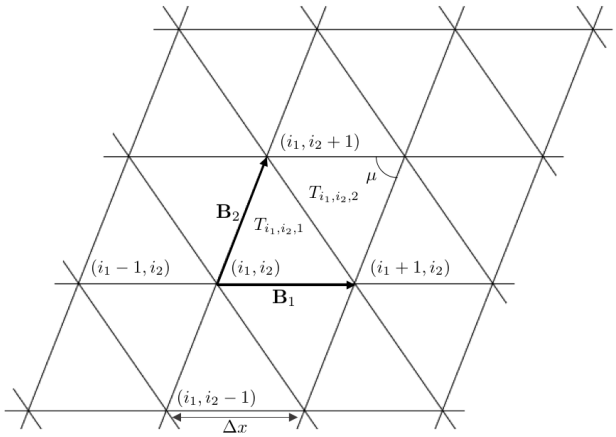


Figure 3.4. – Mesh generating pattern used for the 2D Fourier stability analysis on triangles

Defining $\hat{\mathbf{U}}_j^{i_1,i_2} = [\hat{\mathbf{U}}_j^{i_1,i_2,1}, \hat{\mathbf{U}}_j^{i_1,i_2,2}]^\top$ as the vector collecting the solution in the reference domain on the two triangles $T_{i_1,i_2,1}$ and $T_{i_1,i_2,2}$ for each SP $j \in [\![1, N_{SP}]\!]$, the SDRT spatial discretization using an upwind flux on this mesh takes the form:

$$\frac{d\hat{\mathbf{U}}_j^{i_1,i_2}}{dt} = -\frac{||\mathbf{c}||}{\Delta x}\Big[\mathbf{M}^{0,0}\hat{\mathbf{U}}_j^{i_1,i_2} + \mathbf{M}^{-1,0}\hat{\mathbf{U}}_j^{i_1-1,i_2} + \mathbf{M}^{+1,0}\hat{\mathbf{U}}_j^{i_1+1,i_2} + \mathbf{M}^{0,-1}\hat{\mathbf{U}}_j^{i_1,i_2-1} + \mathbf{M}^{0,+1}\hat{\mathbf{U}}_j^{i_1,i_2+1}\Big]. \quad (3.22)$$

In Eq. (3.22), $\mathbf{M}^{0,0}$, $\mathbf{M}^{-1,0}$, $\mathbf{M}^{+1,0}$, $\mathbf{M}^{0,-1}$ and $\mathbf{M}^{0,+1}$ are matrices of size $[2N_{SP}, 2N_{SP}]$ containing the three steps of the spatial discretization (extrapolation, flux computation and differentiation), which depend on the advection angle θ, the grid angle μ as well as on the SP and FP locations. The exact formulation of those matrices is given in Appendix D.1. The discrete numerical solution is now assumed under the form of a planar harmonic wave:

$$\hat{\mathbf{U}}^{i_1,i_2} = \tilde{\mathbf{U}} \exp\Big(I\mathbf{k}\left(i_1\mathbf{B}_1 + i_2\mathbf{B}_2\right)\Big), \tag{3.23}$$

where $\tilde{\mathbf{U}}$ is a complex vector of dimension $2N_{SP}^{tri}$, independent of i_1 and i_2, and $\mathbf{k} = k(\cos\vartheta, \sin\vartheta)^\top$, k being the wavenumber of the harmonic wave and ϑ its orientation angle.
Using the non-dimensional quantities previously introduced, the discrete numerical solution is:

$$\hat{\mathbf{U}}^{i_1,i_2} = \tilde{\mathbf{U}} \exp\Big(I\kappa\big((i_1 + i_2\cos\mu)\cos\vartheta + i_2\sin\mu\sin\vartheta\big)\Big), \tag{3.24}$$

$\kappa = k\Delta x$ being the grid frequency. Injecting Eq. (3.24) into Eq. (3.22), one gets:

$$\begin{aligned}\frac{d\tilde{\mathbf{U}}}{dt} = -\frac{||\mathbf{c}||}{\Delta x}\Big[&\mathbf{M}^{0,0} + \mathbf{M}^{-1,0}\exp\Big(-I\kappa\cos\vartheta\Big)\\ &+ \mathbf{M}^{+1,0}\exp\Big(I\kappa\cos\vartheta\Big)\\ &+ \mathbf{M}^{0,-1}\exp\Big(-I\kappa(\cos\mu\cos\vartheta + \sin\mu\sin\vartheta)\Big)\\ &+ \mathbf{M}^{0,+1}\exp\Big(I\kappa(\cos\mu\cos\vartheta + \sin\mu\sin\vartheta)\Big)\Big]\,\tilde{\mathbf{U}}\\ = \frac{||\mathbf{c}||}{\Delta x}&\mathbf{M}_z\tilde{\mathbf{U}}.\end{aligned} \tag{3.25}$$

The complete spectrum of the SDRT spatial operator $\lambda_{\mathbf{M}_z}$ can be obtained by computing the eigenvalues of $\mathbf{M}_z$. The matrix $\mathbf{M}_z$ depends on:

- the SP location,
- the FP location,
- the advection angle $\theta \in [0, 2\pi]$,
- the grid frequency $\kappa \in [-\pi, \pi]$,
- the harmonic plane orientation $\vartheta \in [0, 2\pi]$,
- the skew angle $\mu \in [0, \pi/2]$.

Using the eigenvalue analysis, the SDRT spatial discretization is stable under a Fourier stability analysis if $\mathrm{Re}(\lambda_{\mathbf{M}_z})$ is non-positive. The Fourier analysis was applied to the SDRT scheme for triangular [142, 143] and hybrid grids [144]. In both cases, the SDRT scheme is found linearly stable up to $p = 3$ using the interior FP location given in section 3.3.2. Efforts were made to determine stable formulations for $p > 3$ but results were not successful.

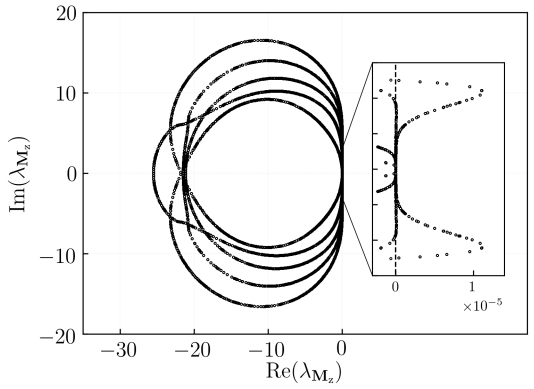
The spectrum of the spatial SDRT operator is computed for $p \in [\![4, 6]\!]$ using Fourier analysis for different implementations (i.e. different interior FP locations) for $\kappa \in [-\pi, \pi]$, $\vartheta \in [0, 2\pi]$, $\theta = 0$ and $\mu = \pi/2$. The SP location is set to the Williams-Shunn-Jameson quadrature points [174]. Values of $\max(\mathrm{Re}(\lambda_{\mathbf{M}_z}))$ are displayed in Table 3.1 for each SDRT implementation based on interior FP locations taken as the quadrature rules presented in section 3.3.2. The first observation is that all SDRT implementations show positive values of $\max(\mathrm{Re}(\lambda_{\mathbf{M}_z}))$, indicating that the spatial discretization is unstable. One can then note that only two quadrature rules are appropriated for both $p = 4$, $p = 5$ and $p = 6$: the Williams-Shunn-Jameson and the Vioreanu-Rokhlin. For the three polynomial degrees p, the use of the WSJ quadrature rule as the interior FP leads to smaller values of $\max(\mathrm{Re}(\lambda_{\mathbf{M}_z}))$

compared to the Vioreanu-Rokhlin quadrature rule. For the $SDRT_5$ scheme, two of the quadrature rules (Laursen-Gellert and Papanicolopulos) lead to very high values of $\max(\mathrm{Re}(\lambda_{\mathbf{M}_z}))$, whereas the smaller value is given by the Witherden-Vincent quadrature rule. For the $SDRT_6$ scheme, positive values of $\max(\mathrm{Re}(\lambda_{\mathbf{M}_z}))$ are obtained for each quadrature rule.

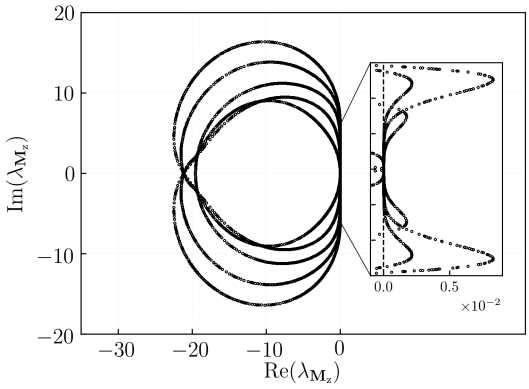
Quadrature rule	$SDRT_4$	$SDRT_5$	$SDRT_6$
Williams-Shunn-Jameson [174]	$1.11 \cdot 10^{-5}$	$5.85 \cdot 10^{-5}$	$2.87 \cdot 10^{-3}$
Vioreanu-Rokhlin [173]	$8.29 \cdot 10^{-3}$	$1.75 \cdot 10^{-2}$	$2.92 \cdot 10^{-2}$
Laursen-Gellert [169]	-	$> 10^{12}$	$9.82 \cdot 10^{-1}$
Witherden-Vincent [177]	-	$1.31 \cdot 10^{-5}$	-
Xiao-Gimbutas [172]	-	$7.33 \cdot 10^{-2}$	-
Papanicolopulos [178]	-	$> 10^{12}$	-

Table 3.1. – Values of $\max(\mathrm{Re}(\lambda_{\mathbf{M}_z}))$ for $\theta = 0$ using different quadrature rules as the interior FP locations

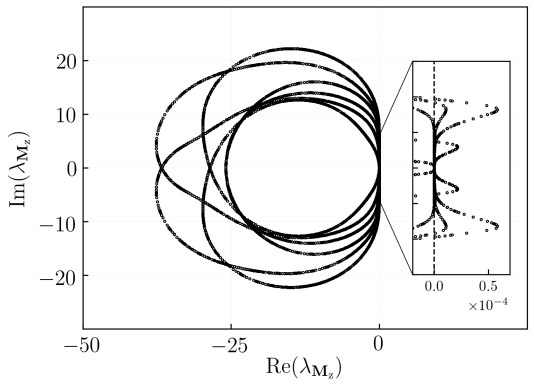
Spectra of unstable discretizations are plotted in Fig. 3.5 for $SDRT_4$ using Williams-Shunn-Jameson (Fig. 3.5a) and Vioreanu-Rokhlin (Fig. 3.5b) quadrature rules and for $SDRT_5$ using Williams-Shunn-Jameson (Fig. 3.5c) and Witherden-Vincent (Fig. 3.5d). A closer view on each spectra allows one to clearly see the positive eigenvalues real part of the spatial operator $\mathbf{M}_z$.



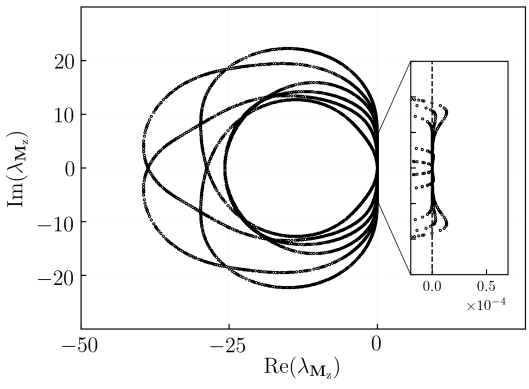
(a) $SDRT_4$, Williams-Shunn-Jameson rule



(b) $SDRT_4$, Vioreanu-Rokhlin rule



(c) $SDRT_5$, Williams-Shunn-Jameson rule



(d) $SDRT_5$, Witherden-Vincent rule

Figure 3.5. – Fourier footprint of the $SDRT_4$ and $SDRT_5$ spatial discretizations on triangles for $\theta = 0$ using different interior FP locations

3.3.4. Stability Analysis on a Fixed Number of Cells

The FNC approach introduced in section 3.1.3 is considered to seek stable SDRT schemes for $p > 3$ by studying multiple interior FP locations.

Analysis on Two Triangles The behavior of the SDRT scheme is studied on a regular mesh composed of two triangles T_1 and T_2 (Fig. 3.6) with periodic boundary conditions. It is recalled here that the main difference with the common Fourier analysis is that no plane harmonic wave is injected. The periodic boundary conditions are imposed through the numerical flux. The velocity vector is $\mathbf{c} = (\cos\theta, \sin\theta)^\top$. Using symmetry properties, the choice of the advection angle is reduced to $\theta \in [0, \pi/4]$.

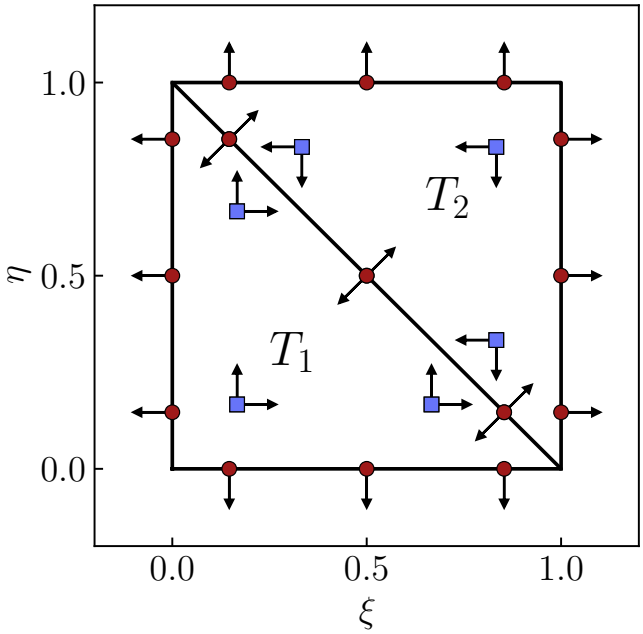


Figure 3.6. – Computational domain for the FNC linear stability analysis on triangles - Example of FP distribution for $p = 2$, $N = 2$ (edge: ●, interior: ■)

On the computational domain $\Omega = (T_1, T_2)$, Eq. (3.8) takes the form:

$$\begin{bmatrix} \dfrac{\partial \hat{\mathbf{U}}_j^{T_1}(t)}{\partial t} \\ \dfrac{\partial \hat{\mathbf{U}}_j^{T_2}(t)}{\partial t} \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_{jk} & 0 \\ 0 & \mathbf{D}_{jk} \end{bmatrix} \mathbf{C} \begin{bmatrix} \mathbf{T}_{kj} & 0 \\ 0 & \mathbf{T}_{kj} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{U}}_j^{T_1}(t) \\ \hat{\mathbf{U}}_j^{T_2}(t) \end{bmatrix}, \quad j \in [\![1, N_{SP}^{tri}]\!], \quad k \in [\![1, N_{FP}^{tri}]\!], \tag{3.26}$$

where $\hat{\mathbf{U}}_j^{T_1}$ and $\hat{\mathbf{U}}_j^{T_2}$ as the vectors collecting the solution values at SP for the triangle T_1 and T_2. The transfer matrix is given by Eq. (2.90):

$$\mathbf{T}_{kj} = \sum_{m=1}^{N_{SP}^{tri}} \left(\Phi_m(\boldsymbol{\xi}_j)\right)^{-1} \Phi_m(\boldsymbol{\xi}_k), \tag{3.27}$$

and the differentiation matrix by Eq. (2.101):

$$\mathbf{D}_{jk} = \sum_{n=1}^{N_{FP}^{tri}} \left(\boldsymbol{\phi}_n(\boldsymbol{\xi}_k) \cdot \hat{\mathbf{n}}_k\right)^{-1} \hat{\nabla} \cdot \boldsymbol{\phi}_n(\boldsymbol{\xi}_j). \tag{3.28}$$

The velocity matrix $\mathbf{C}$ should verify Eq. (3.6) while taking into account the mesh connectivity and the periodic boundary conditions. Following the FP numbering previously settled, it can be expressed as:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}^L & O_{3N_e,N_i} & \mathbf{C}^R & O_{3N_e,N_i} \\ O_{N_i,3N_e} & \mathbf{C}^I_{N_i} & O_{N_i,3N_e} & O_{N_i,N_i} \\ \mathbf{C}^R & O_{3N_e,N_i} & \mathbf{C}^L & O_{3N_e,N_i} \\ O_{N_i,3N_e} & O_{N_i,N_i} & O_{N_i,3N_e} & \mathbf{C}^I_{N_i} \end{bmatrix}, \tag{3.29}$$

where $\mathbf{C}^I$, $\mathbf{C}^L$ and $\mathbf{C}^R$ are defined by:

$$\mathbf{C}^I = \left[\mathrm{diag}(|J|J^{-1}(\mathbf{c}\cdot\hat{\mathbf{n}}))\right]_{N_i,N_i}, \tag{3.30}$$

$$\mathbf{C}^L = (\mathbf{c}\cdot\mathbf{n})\left[\mathrm{diag}(\frac{1+\mathrm{sign}(\mathbf{c}\cdot\mathbf{n})}{2})\right]_{3N_e,3N_e}, \tag{3.31}$$

and

$$\mathbf{C}^R = (\mathbf{c}\cdot\mathbf{n})\left[\mathrm{bdiag}(\begin{bmatrix} 0 & \dots & \frac{1-\mathrm{sign}(\mathbf{c}\cdot\mathbf{n})}{2} \\ \vdots & \ddots & \vdots \\ \frac{1-\mathrm{sign}(\mathbf{c}\cdot\mathbf{n})}{2} & \dots & 0 \end{bmatrix}_{N_e,N_e})\right]_{3,3}. \tag{3.32}$$

where, when applied to an arbitrary square matrix $\mathbf{A}$, the operator bdiag() gives a block diagonal matrix such that the main-diagonal blocks are the square matrix $\mathbf{A}$ and all off-diagonal blocks are zeros matrices. The stability of the scheme on two triangles is then study through the matrix $\mathbf{M}_\mathrm{n} = -\,\mathrm{diag}(\mathbf{D})\ \mathbf{C}\ \mathrm{diag}(\mathbf{T})$.

Optimization process To study multiple interior FP location, an optimization problem is considered. The aim is to minimize $\mathrm{Re}(\lambda_{\mathbf{M}_\mathrm{n}})$ by considering the interior FP location as the optimization parameters. The optimization algorithm used is the Differential Evolution algorithm from the SciPy library [179]. This algorithm optimizes a given problem by considering a population of candidate solutions (here, the interior FP location) and improves them according to a given measure of quality (here, $\mathrm{Re}(\lambda_{\mathbf{M}_\mathrm{n}})$). The choice of this algorithm is motivated by the ability of Differential Evolution to search very large spaces of candidate solutions.

To set up the interior FP position, a symmetry along $y = x$ is imposed in the triangle. The position of the interior FP located on this line is defined by a parameter $\alpha_i \in]0, 0.5[$. For $p = 4$ (respectively $p = 5$), the number of FP on this line is 2 (respectively 3). Note that this choice is arbitrary. To set up the position of the remaining interior FP, two parameters $(\beta_i, \gamma_i) \in]0, 1[\times]0, \beta_i/2[$ per points are needed. The parameter β_i is used to browse the triangle from 0 to $y = -x + 1$ using $y = -x + \beta_i$. The parameter γ_i is used to adjust the position of the point on the line $y = -x + \beta_i$. Because of the symmetry rule, one set (β_i, γ_i) gives two FP located at $(\beta_i/2 + \gamma_i, \beta_i/2 - \gamma_i)$ and $(\beta_i/2 - \gamma_i, \beta_i/2 + \gamma_i)$, thus the number of parameters in the optimization problem is equal to $N_i/2$. An example of the set up for $p = 3$ is given in Fig. 3.7 and the associated coordinate parameters are given in Table 3.2.

Flux Point	(ξ, η)
$\boldsymbol{\xi}_{10}$	(α_1, α_1)
$\boldsymbol{\xi}_{14}$	(α_2, α_2)
$\boldsymbol{\xi}_{11}$	$(\beta_1/2 + \gamma_1, \beta_1/2 - \gamma_1)$
$\boldsymbol{\xi}_{13}$	$(\beta_1/2 - \gamma_1, \beta_1/2 + \gamma_1)$
$\boldsymbol{\xi}_{12}$	$(\beta_2/2 + \gamma_2, \beta_2/2 - \gamma_2)$
$\boldsymbol{\xi}_{15}$	$(\beta_2/2 - \gamma_2, \beta_2/2 + \gamma_2)$

Table 3.2. – Interior FP coordinate parameters for SDRT_3

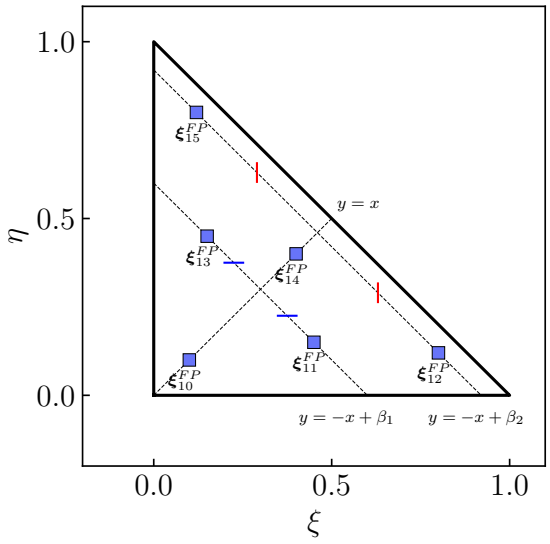
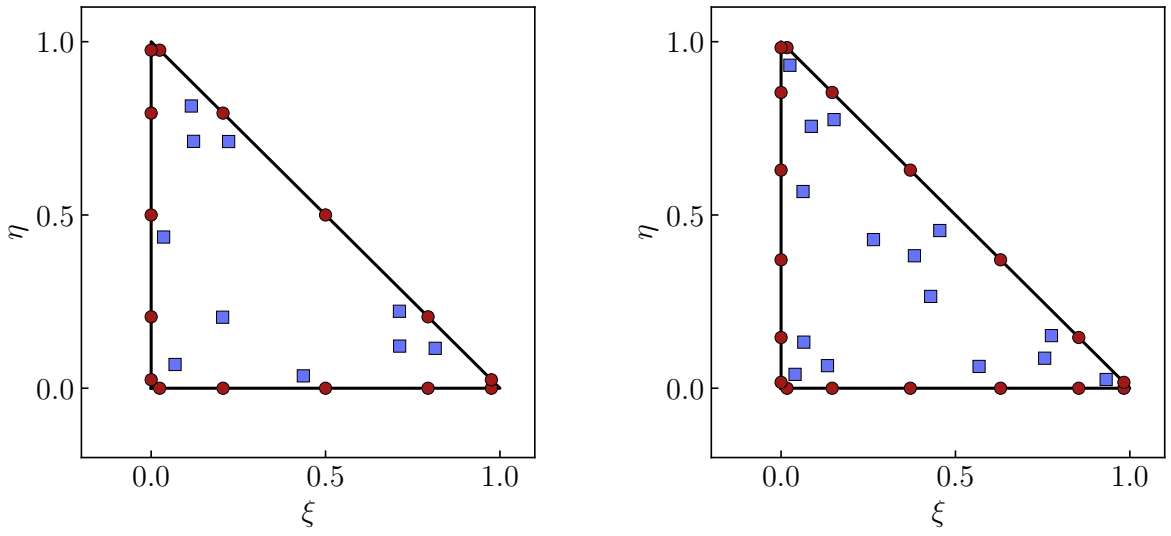


Figure 3.7. – Interior FP set up in the reference triangle for SDRT_3

The Differential Evolution algorithm is run for SDRT schemes up to the sixth-order ($p = 5$). For each scheme, several sets of interior FP lead a spectrum composed of non positive values of $\text{Re}(\lambda_{\mathbf{M}_n})$. This result indicates that the definition of a stable SDRT formulation on two triangles is not unique. One of the possible sets of coefficients to built the position of the interior FP is plotted in Fig. 3.8 for SDRT_4 and SDRT_5. From this figure, it can be noted that the interior FP are not evenly distributed. This is due to the fact that the Differential Evolution algorithm did not take into account an initial interior FP distribution, which could be given by a quadrature rule for example. The interior FP distribution determined by the Differential Evolution optimization process might then be associated with a significant extrapolation error.

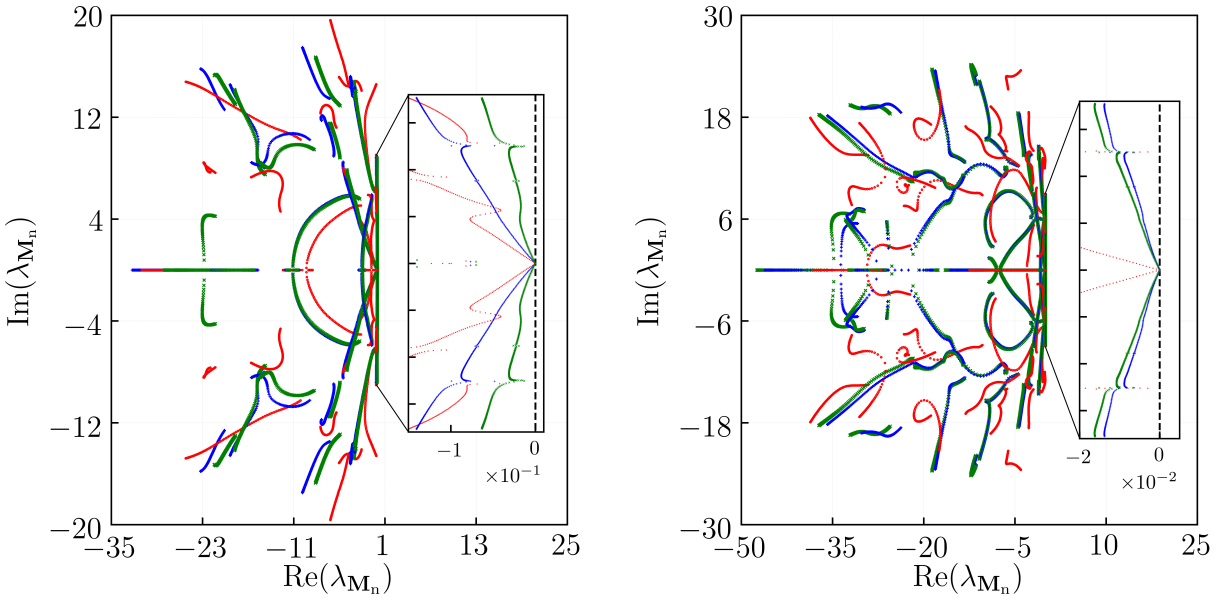


(a) SDRT_4 (b) SDRT_5

Figure 3.8. – Sets of FP determined using FNC analysis in the reference triangle

Quadrature rules While the optimization process was running, the capability of the published quadrature rules given in section 3.3.2 to lead to stable formulations for $p > 3$ on two triangles was also analyzed.

For $SDRT_4$ and $SDRT_5$ schemes, two quadrature rules per order lead to a linearly stable scheme: the Williams-Shunn-Jameson [174] and the Vioreanu-Rokhlin [173] quadrature rules for the $SDRT_4$; the Williams-Shunn-Jameson [174] and the Witherden-Vincent [177] quadrature rules for the $SDRT_5$. Spectra of SDRT formulations obtained with the FNC approach using the quadratures rules and the optimization process as interior FP are given in Fig. 3.9. All values of $\max(\mathrm{Re}(\lambda_{\mathbf{M}_n}))$ are negative. For SDRT implementations based on interior FP located at quadrature points, these results differ with the standard Fourier analysis results presented in the previous section.



(a) $SDRT_4$: Williams-Shunn-Jameson rule (+), Vioreanu-Rokhlin rule (×), Optimization (•)

(b) $SDRT_5$: Williams-Shunn-Jameson rule (+), Witherden-Vincent rule (×), Optimization (•)

Figure 3.9. – Spectra of matrix $\mathbf{M}_n$ obtained using the FNC method for $SDRT_4$ and $SDRT_5$ schemes

Comments on SDRT$_6$ For the $SDRT_6$ scheme, none of the tested quadrature rules led to a stable scheme on two triangles. When using the optimization process for $p = 6$, the optimization becomes too expensive in time: the CPU time per set of points is almost 3 times higher than for $p = 4$ and, given the large number of interior FP, the process has to be run longer. The differential algorithm did not manage to detect a non-positive spectrum before it was stopped. This does not state the fact that stable formulations could not be obtained in the future.

Differential Evolution Optimization using Fourier Analysis The Differential Evolution optimization process has also been run using the standard Fourier analysis. However, since it implies a grid frequency discretization, the computation of $\mathbf{M}_z$ eigenvalues is done repeatedly for each grid frequency. To compute eigenvalues on a triangular grid using $SDRT_4$ for a given velocity using $\mu = \pi/2$ and $\vartheta \in [0, 2\pi], \kappa \in [-\pi, \pi]$, each discretized using 11 values, the Fourier analysis is 15 times slower than the FNC method. The optimization process based on the Differential Evolution did not manage to found any stable sets of interior FP using the Fourier analysis.

Considering a small computational domain composed of two triangles, stable spectra of the spatial operator matrix $\mathbf{M}_\mathrm{n}$ were obtained for SDRT_4 and SDRT_5 by setting the interior FP position using two different means (through an optimization process and published quadrature rules). Results given by this analysis on two triangles differ with the standard Fourier analysis since spectra showing stable SDRT discretizations were obtained for SDRT implementations based on interior FP located at quadrature points.
The FNC approach is now used to investigate the stability of the SDRT discretization on computational domains composed of a larger number of cells. The aim is to examine the assumption that the stability on two triangles would lead to stability on any domain.

Analysis on $2\mathbf{N}^2$ Triangles The analysis performed on two triangles in the previous section is now extended to a regular domain composed of $2N^2$ triangles. An example of the considered domain is given in the case of $N = 5$ in Fig. 3.10. The aim is to verify that the FP locations previously determined lead to stable SDRT formulations when the number of cells increases.

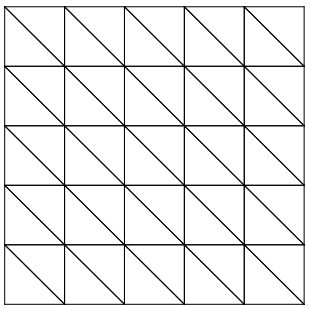
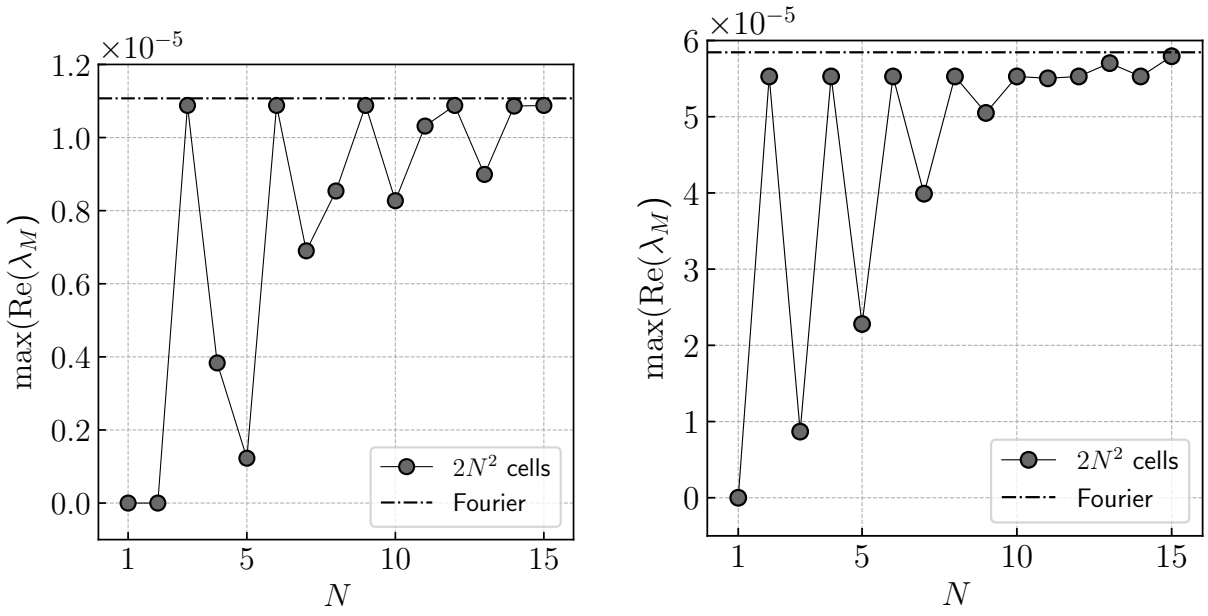


Figure 3.10. – Computational domain for the FNC linear stability analysis on triangles - Example for $N = 5$

In Fig. 3.11, the maximum of the real part of M eigenvalues is plotted as a function of N for the advection angle $\theta = 0$ for SDRT_4 (right) and SDRT_5 (left). The interior FP location is taken as the William-Shunn-Jameson quadrature points. The first observation that can be made is that when the number of cells increases, the spatial discretization becomes unstable, with a maximum eigenvalue real part of order 10^{-5}. In fact, as N increases, the maximum eigenvalue real part tends to the values given by the Fourier analysis. There is a dependency between the number of cells N and the value of $\max(\mathrm{Re}(\lambda_{\mathbf{M}_\mathrm{n}}))$. For $p = 4$, the scheme is stable for $N = 1$ and $N = 2$. When N increases, the spatial discretization becomes unstable. If N is a multiple of 3, then the same value is always obtained. For $p = 5$, the only stable case is $N = 1$. The same value of $\max(\mathrm{Re}(\lambda_{\mathbf{M}}))$ is obtained if N is even.

Analyses for interior FP location given by the optimization process or by other quadrature rules lead to the same conclusion: the SDRT spatial discretization is not stable when the number of cells in the computational domain increases. By considering a computational domain composed of a fixed number of cells, the stability results are strongly dependent on the number of cells. A pattern exists depending on the nature of N (multiple of 3 for $p = 4$, even for $p = 5$) but conclusions can not be extended to a general case. Similar observations were made when studying the CFL limit of the standard 1D scheme with the FNC approach. Those results, presented in Appendix C, show that the approach can recover the Fourier stability analysis results if the DoF number, given by $N_{DoF} = (p+1)N$, is even. For both the standard 1D SD and the SDRT scheme, the Fourier analysis gives a more general framework.



(a) $SDRT_4$ (b) $SDRT_5$

Figure 3.11. – Maximum eigenvalue real part of $\mathbf{M}_n$ determined using the FNC analysis for different domain size

To describe the stability behavior of the SDRT scheme in a general framework, the FNC approach is put aside in favor of the common Fourier analysis. The Fourier stability analysis presented in section 3.3.3 showed the instability of SDRT schemes for orders of accuracy higher than 4. To overcome this limitation, two different solutions will be explored.
The first possibility is based on the observation that using specific quadrature rule as the interior FP location, positive real parts of the SDRT spatial operator eigenvalues are of small order. These small positive values could then be dissipated by the numerical dissipation induced by the temporal discretization. Chapter 4 investigates this option.
The second possibility consists in finding interior FP locations leading to SDRT discretization proven as stable under a Fourier analysis. Optimization algorithms seem to be a promising technique to determine such sets of FP. However, the Differential Evolution algorithm introduced in this chapter is not compatible with the Fourier analysis because of a high computational time. Additionally, since no initial condition can be specified, there is no guarantee that interior FP determined using the Differential Evolution algorithm would lead to a negligible extrapolation error. To overcome those limitations, the use of a gradient descent algorithm could be the solution since this method is less costly in time and can take an initial solution into account. This alternative is addressed in Chapter 5.
Before that, the Fourier analysis is applied to tetrahedral elements.

3.4. Spectral Difference Scheme Using Raviart-Thomas Elements on Tetrahedral Elements

The previous section has shown that the Fourier analysis is the proper way to address the SDRT scheme stability in a general framework. It is thus the only analysis conducted to study the stability of the SDRT scheme on tetrahedral elements.

3.4.1. Reference Element Convention and Points Numbering

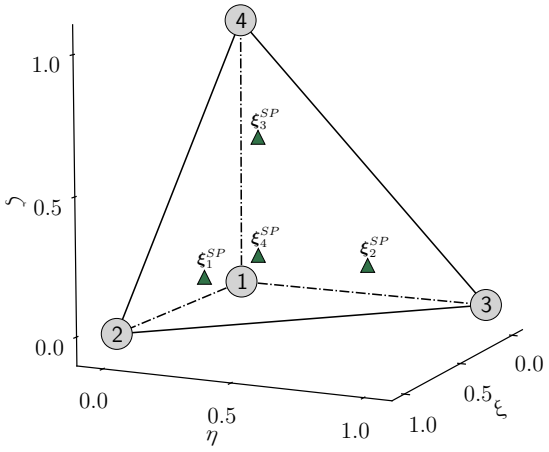
For clarity purposes, the SP and FP numbering and their associate normal vector in the reference tetrahedron are settled in this section. The reference tetrahedron is defined through its four nodes and four faces. The face numbering is chosen to be consistent with the CFD General Notation System (CGNS). The number of SP is $N_{SP}^{tet} = (p+1)(p+2)(p+3)/6$. They are represented by green triangles in Fig. 3.12a. On each face, there are $N_f^{tet} = (p+1)(p+2)/2$ FP, simply denoted N_f here since this section is dedicated to tetrahedral elements. They are denoted $\boldsymbol{\xi}_k$, where k is defined in the following, and associated with the normal vectors:

- Face 1: N_1, N_3, N_2, $k \in [\![1, N_f]\!]$, $\hat{\mathbf{n}} = (0, 0, -1)^\top$,
- Face 2: N_1, N_2, N_4, $k \in [\![N_f + 1, 2N_f]\!]$, $\hat{\mathbf{n}} = (0, -1, 0)^\top$,
- Face 3: N_2, N_3, N_4, $k \in [\![2N_f + 1, 3N_f]\!]$, $\hat{\mathbf{n}} = (1, 1, 1)^\top$,
- Face 4: N_3, N_1, N_4, $k \in [\![3N_f + 1, 4N_f]\!]$, $\hat{\mathbf{n}} = (-1, 0, 0)^\top$.

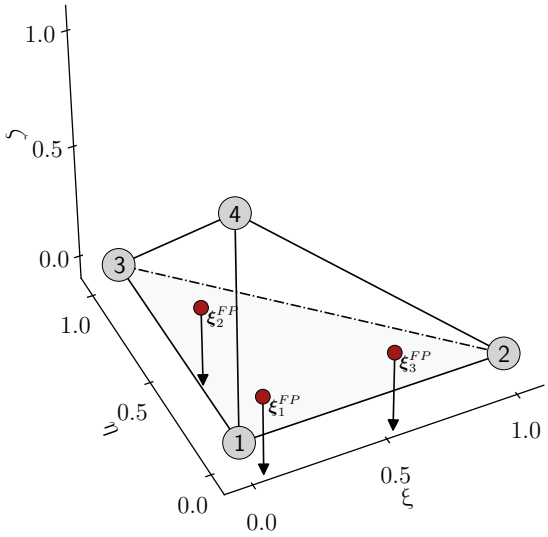
Additionally, a convention has to be chosen to set the distribution of FP on a same face. Let us consider a face defined by three nodes (N_A, N_B, N_C). FP will be numbered using a 2D spatial system formed by the vectors $\overrightarrow{N_A\,N_B}$ and $\overrightarrow{N_A\,N_C}$. The face origin is thus the point N_A. The index k will be increasing among the $\overrightarrow{N_A\,N_B}$ direction first, and then among $\overrightarrow{N_A\,N_C}$. On each face, the 2D spatial system is chosen as:

- Face 1: $(N_A, N_B, N_C) = (N_1, N_3, N_2)$,
- Face 2: $(N_A, N_B, N_C) = (N_1, N_2, N_4)$,
- Face 3: $(N_A, N_B, N_C) = (N_2, N_3, N_4)$,
- Face 4: $(N_A, N_B, N_C) = (N_1, N_4, N_3)$.

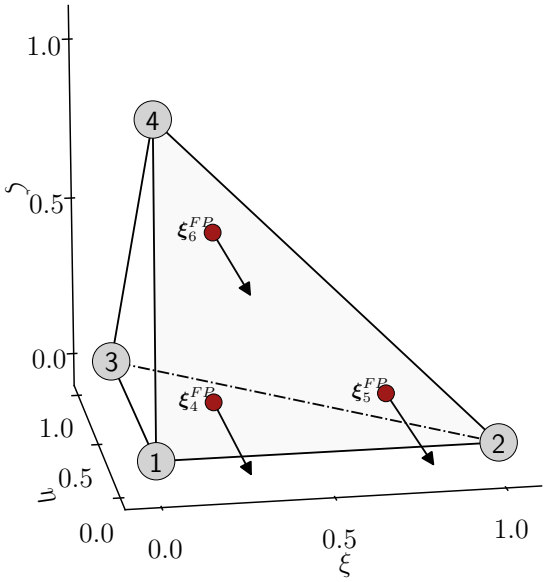
This convention is illustrated in Fig. 3.12b, 3.12c, 3.12d, 3.12e. The remaining $N_i^{tet} = p(p+1)(p+2)/2$ FP, simply denoted N_i, are located inside the tetrahedron. The number of physical FP is reduced to $N_i/3$ by considering each interior FP as three different degree of freedom associated with the normal vector $\hat{\mathbf{n}} = (1, 0, 0)^\top$, $\hat{\mathbf{n}} = (0, 1, 0)^\top$, $\hat{\mathbf{n}} = (0, 0, 1)^\top$ (Fig. 3.12f).



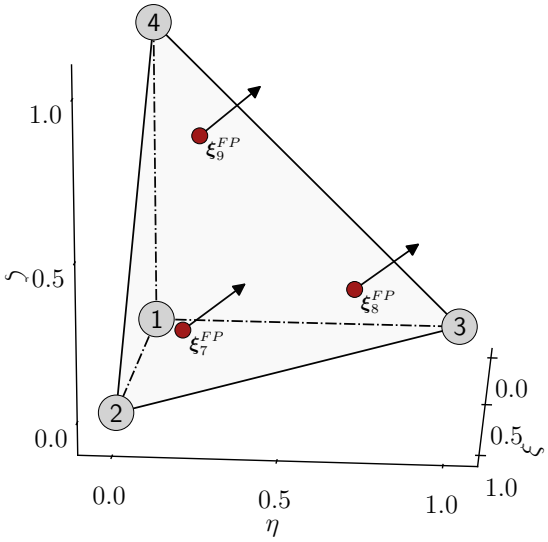
(a) SP



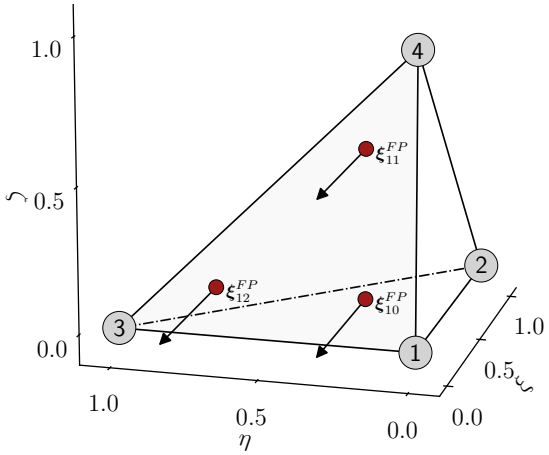
(b) FP, Face 1



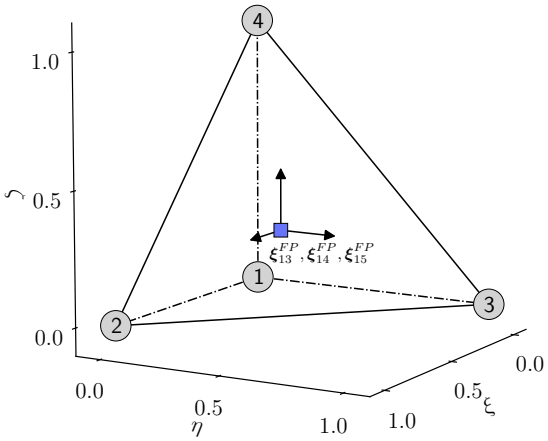
(c) FP, Face 2



(d) FP, Face 3



(e) FP, Face 4



(f) Interior FP

Figure 3.12. – Points distribution in the tetrahedral reference element for $p = 1$: SP $\boldsymbol{\xi}_j$ (▲) and FP $\boldsymbol{\xi}_k$ (face (●) and interior (■))

3.4.2. Choice of Solution Points and Flux Points Location

The location of SP and FP needs to be chosen for the reference tetrahedron. The number of SP is given by:

$$N_{SP}^{tet} = \frac{1}{6}(p+1)(p+2)(p+3). \tag{3.33}$$

The number of FP located on each face is equal to $(p+1)(p+2)/2$, which corresponds to the number of SP on a triangle. By choosing their location to be the same as the SP on a triangle, it is ensured that a tetrahedral and a prismatic element will share the same FP on faces, avoiding the need to apply mortar techniques as detailed in Section 3.1.1.2. The number of interior FP is given by:

$$N_{i}^{tet} = \frac{1}{2}p(p+1)(p+2). \tag{3.34}$$

Since each physical FP is counted as three separated DoF, the number of physical interior FP for which the position has to be settled is $N_{pi}^{tet} = \frac{1}{6}p(p+1)(p+2)$. The number of SP and physical interior FP is summarized in Table 3.3. It can be noted that the number of physical interior FP for a SDRT_p scheme correspond to the number of SP for a SDRT_{p-1} scheme.

p	N_{SP}^{tet}	N_{pi}^{tet}
1	4	1
2	10	4
3	20	10
4	35	20
5	56	35
6	84	56

Table 3.3. – Number of SP and physical interior FP for SDRT scheme on tetrahedral elements

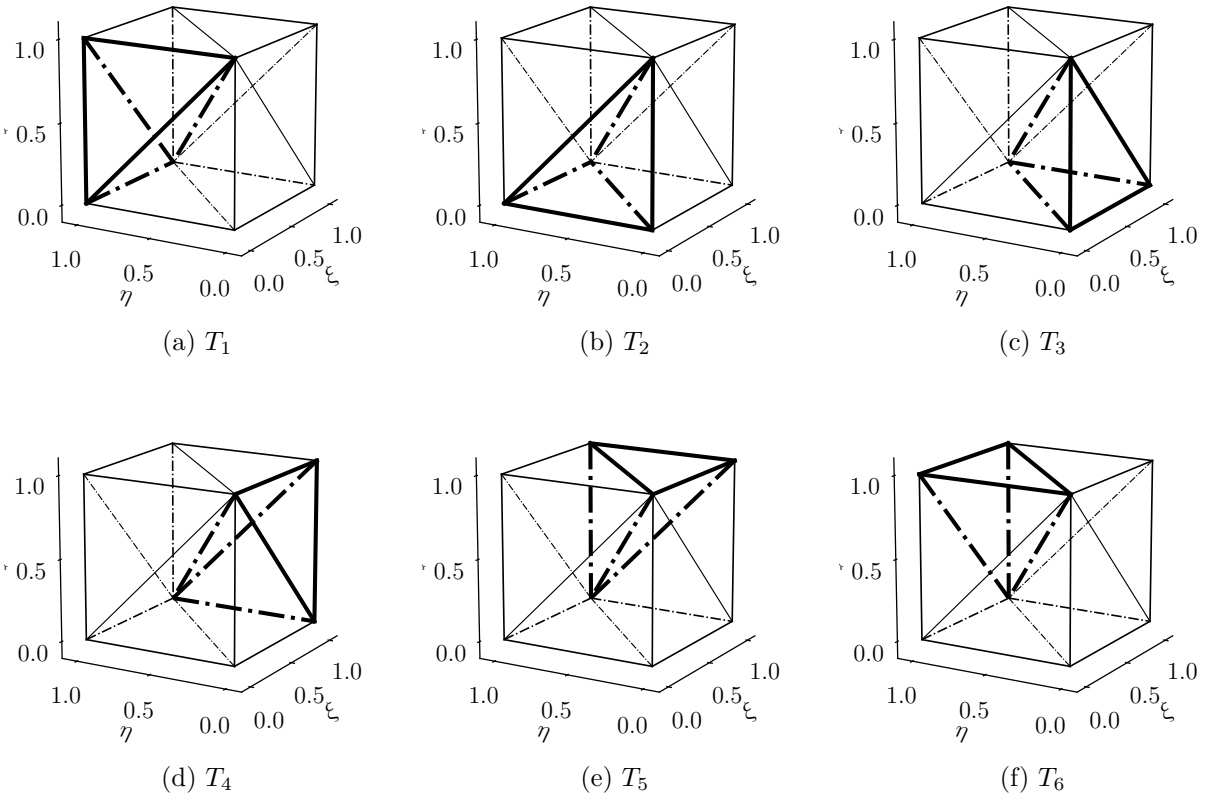
To set the SP and physical interior FP locations on tetrahedral elements, quadrature rules available in the literature are studied. To be suitable for the SDRT implementation, the quadrature rules should not have points located on corner, edge or face. Three quadrature rules are found to lead to the appropriate number of points for each degree p while fulfilling this requirement: the Newton-Cotes Open (NCO) [180], the Vioreanu-Rokhlin [173] and the Shunn-Ham [181] quadrature rules. Since those quadratures rules are suitable for each degree p, they can be used for both the SP and the physical interior FP by choosing the adequate quadrature order. Other quadrature rules can lead to the proper number of points for a given degree p and will be given below. The SP are chosen to be located at the Shunn-Ham quadrature points. For the physical interior FP:

- For $p = 1$, all the studied quadrature rules led to the same physical interior point located at $(\xi, \eta, \zeta) = (0.25, 0.25, 0.25)$.
- For $p = 2$, it is noted that several quadrature rules lead to the very same set of points (Keast [182], Vioreanu-Rokhlin [173], Shunn-Ham [181], Witherden-Vincent [177], Yu [183], Hammer-Marlowe-Stroud [168], Liu-Vinokur [184]). This set of point will be studied here as the Shunn-Ham quadrature rule. Three other quadrature rules containing four points will be studied: the Jaśkowiec-Sukumar [185], the Xiao-Gimbutas [172] and the NCO [180].
- For $p > 2$, the studied quadrature rules are Shunn-Ham, Vioreanu-Rokhlin and NCO, which contain the appropriate number of points for each p.

3.4.3. Matrix Form

Let us consider the 3D linear advection problem given by Eq. (3.1) on a domain $\Omega = [0, L]^3$ with periodic boundary conditions. The velocity vector is $\mathbf{c} = (\sin\theta_2 \cos\theta_1, \sin\theta_2 \sin\theta_1, \cos\theta_2)^\top$ where

$(\theta_1, \theta_2) \in [0, \pi/4]$. The domain Ω is meshed as a Cartesian mesh composed of $N_x \times N_y \times N_z$ hexahedral elements of size $\Delta x \times \Delta y \times \Delta z$, with $\Delta x = \Delta y = \Delta z$. Each hexahedral cell is then divided into tetrahedron. An hexahedron can be decomposed into a minimum of five tetrahedral elements, but to ensure the periodicity, six tetrahedron are required. The six tetrahedron of the hexahedral cell (i_1, i_2, i_3) are denoted $T^{i_1,i_2,i_3,1}, T^{i_1,i_2,i_3,2}, T^{i_1,i_2,i_3,3}, T^{i_1,i_2,i_3,4}, T^{i_1,i_2,i_3,5}, T^{i_1,i_2,i_3,6}$ and are represented in Fig. 3.13.



(a) T_1 (b) T_2 (c) T_3

(d) T_4 (e) T_5 (f) T_6

Figure 3.13. – Computational domain for the Fourier stability analysis on tetrahedral elements

Defining:

$$\hat{\mathbf{U}}_j^{i_1,i_2,i_3} = [\hat{\mathbf{U}}_j^{i_1,i_2,i_3,1}, \hat{\mathbf{U}}_j^{i_1,i_2,i_3,2}, \hat{\mathbf{U}}_j^{i_1,i_2,i_3,3}, \hat{\mathbf{U}}_j^{i_1,i_2,i_3,4}, \hat{\mathbf{U}}_j^{i_1,i_2,i_3,5}, \hat{\mathbf{U}}_j^{i_1,i_2,i_3,6}]^\top, \tag{3.35}$$

as the vector collecting the solution in the reference domain on the six tetrahedron for each SP $j \in [\![1, N_{SP}]\!]$ on cell (i_1, i_2, i_3), the SDRT spatial discretization using an upwind flux on this mesh takes the form:

$$\begin{aligned}\frac{d\hat{\mathbf{U}}_j^{i_1,i_2,i_3}}{dt} = -\frac{||\mathbf{c}||}{\Delta x}\Big[&\mathbf{M}^{0,0,0}\,\hat{\mathbf{U}}_j^{i_1,i_2,i_3} + \mathbf{M}^{-1,0,0}\,\hat{\mathbf{U}}_j^{i_1-1,i_2,i_3} \\ &+ \mathbf{M}^{+1,0,0}\,\hat{\mathbf{U}}_j^{i_1+1,i_2,i_3} \\ &+ \mathbf{M}^{0,-1,0}\,\hat{\mathbf{U}}_j^{i_1,i_2-1,i_3} \\ &+ \mathbf{M}^{0,+1,0}\,\hat{\mathbf{U}}_j^{i_1,i_2+1,i_3} \\ &+ \mathbf{M}^{0,0,-1}\,\hat{\mathbf{U}}_j^{i_1,i_2,i_3-1} \\ &+ \mathbf{M}^{0,0,+1}\,\hat{\mathbf{U}}_j^{i_1,i_2,i_3+1}\Big].\end{aligned} \tag{3.36}$$

In Eq. (3.22), $\mathbf{M}^{0,0,0}$, $\mathbf{M}^{-1,0,0}$, $\mathbf{M}^{+1,0,0}$, $\mathbf{M}^{0,-1,0}$, $\mathbf{M}^{0,+1,0}$, $\mathbf{M}^{0,0,-1}$ and $\mathbf{M}^{0,0,+1}$ are matrices of size $[6N_{SP}, 6N_{SP}]$ containing the three steps of the spatial discretization (extrapolation, flux computation and differentiation), which depend on the advection angles as well as on the SP and FP locations. The exact formulation of those matrices is given in Appendix D.2.

3.4.4. Fourier Stability Analysis

To perform the Fourier stability analysis on tetrahedral elements, the discretized numerical solution is assumed under the form of a planar harmonic wave:

$$\hat{\mathbf{U}}^{i_1,i_2,i_3} = \tilde{\mathbf{U}} \exp\Big(I\mathbf{k}\left(i_1\mathbf{x} + i_2\mathbf{y} + i_3\mathbf{z}\right)\Big), \tag{3.37}$$

where

$$(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \left(\begin{pmatrix}\Delta x\\0\\0\end{pmatrix}, \begin{pmatrix}0\\\Delta x\\0\end{pmatrix}, \begin{pmatrix}0\\0\\\Delta x\end{pmatrix}\right) \tag{3.38}$$

are the vectors defining the mesh, $\tilde{\mathbf{U}}$ is a complex vector of dimension $6N_{SP}^{tet}$, independent of i_1, i_2 and i_3, and

$$\mathbf{k} = k\begin{pmatrix}\cos\vartheta_1 \sin\vartheta_2\\ \sin\vartheta_1 \sin\vartheta_2\\ \cos\vartheta_2\end{pmatrix}, \tag{3.39}$$

k being the wave number of the harmonic wave and $(\vartheta_1, \vartheta_2)$ its orientation angles. Using non-dimensional quantities, Eq. (3.37) becomes:

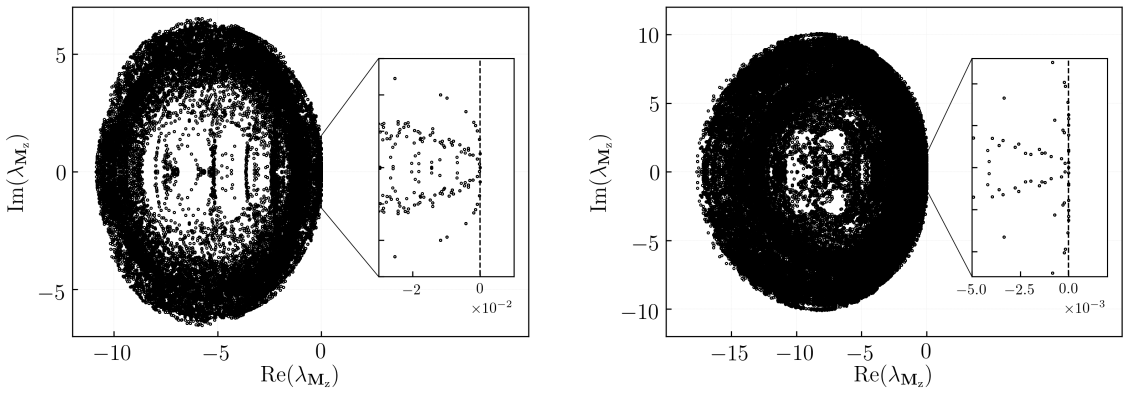
$$\hat{\mathbf{U}}^{i_1,i_2,i_3} = \tilde{\mathbf{U}} \exp\Big(I\kappa\left(i_1\cos\vartheta_1\sin\vartheta_2 + i_2\sin\vartheta_1\sin\vartheta_2 + i_3\cos\vartheta_2\right)\Big), \tag{3.40}$$

$\kappa = k\Delta x$ being the grid frequency. Injecting Eq. (3.40) into Eq. (3.36), one gets:

$$\begin{aligned}\frac{d\tilde{\mathbf{U}}}{dt} = -\frac{||\mathbf{c}||}{\Delta x}\Big[&\mathbf{M}^{0,0,0} + \mathbf{M}^{-1,0,0}\exp(-I\kappa\cos\vartheta_1\sin\vartheta_2)\\ &+ \mathbf{M}^{+1,0,0}\exp(I\kappa\cos\vartheta_1\sin\vartheta_2)\\ &+ \mathbf{M}^{0,-1,0}\exp(-I\kappa\sin\vartheta_1\sin\vartheta_2)\\ &+ \mathbf{M}^{0,+1,0}\exp(I\kappa\sin\vartheta_1\sin\vartheta_2)\\ &+ \mathbf{M}^{0,0,-1}\exp(-I\kappa\cos\vartheta_2)\\ &+ \mathbf{M}^{0,0,+1}\exp(I\kappa\cos\vartheta_2)\Big]\ \tilde{\mathbf{U}}\\ = \frac{||\mathbf{c}||}{\Delta x}\,\mathbf{M}_{\mathrm{z}}\,\tilde{\mathbf{U}}.&\end{aligned} \tag{3.41}$$

The complete spectrum of the SDRT spatial operator $\lambda_{\mathbf{M}_{\mathrm{z}}}$ is obtained by computing the eigenvalues of $\mathbf{M}_{\mathrm{z}}$ over the grid frequency $\kappa \in [-\pi, \pi]$ considering $(\vartheta_1, \vartheta_2) \in [0, 2\pi]$. For $p = 1$, the spectrum of $\mathbf{M}_{\mathrm{z}}$ is plotted in Fig. 3.14a for $(\theta_1, \theta_2) \in (0, \pi/8, \pi/4)^2$. A closer view on the spectrum allows to see the non-positivity of $\mathrm{Re}(\lambda_{\mathbf{M}_{\mathrm{z}}})$ and to establish the stability of the spatial discretization.

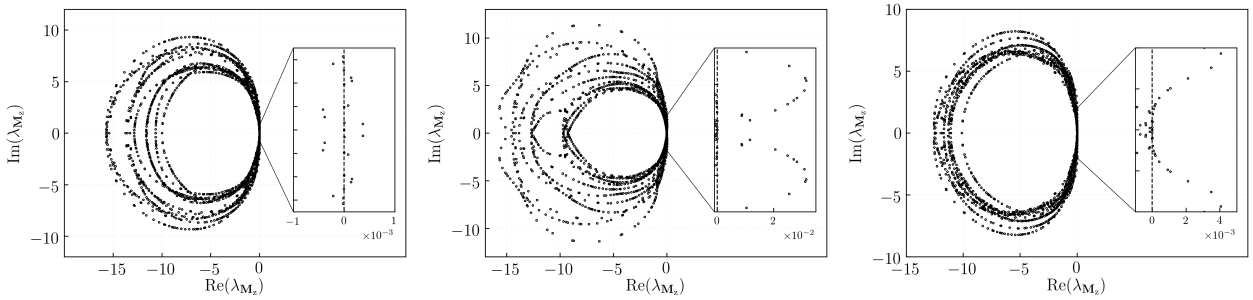
For $p = 2$, there are 4 interior physical FP. In Fig. 3.14b, the spectrum of the SDRT_2 discretization using the Shunn-Ham rule for interior FP is plotted for $(\theta_1, \theta_2) \in (0, \pi/8, \pi/4)^2$. A closer view on the spectrum allows to see the non-positivity of $\mathrm{Re}(\lambda_{\mathbf{M}_{\mathrm{z}}})$, indicating a stable SDRT scheme.



(a) SDRT_1 (b) SDRT_2, Shunn-Ham rule

Figure 3.14. – Spectrum of matrix $\mathbf{M_z}$ for stable SDRT schemes on tetrahedral elements, $(\theta_1, \theta_2) \in (0, \pi/8, \pi/4)^2$

In Fig. 3.15, the spectrum is plotted for the particular case $(\theta_1, \theta_2) = (0, 0)$ for the three other quadrature rules: the Jaśkowiec-Sukumar (Fig. 3.15a), the Xiao-Gimbutas (Fig. 3.15b) and the NCO (Fig. 3.15c). The SDRT scheme using those three quadrature rules is found unstable with $\max(\mathrm{Re}(\lambda_{\mathbf{M_z}})) \sim 3 \cdot 10^{-4}$ for Jaśkowiec-Sukumar, $\max(\mathrm{Re}(\lambda_{\mathbf{M_z}})) \sim 3 \cdot 10^{-2}$ for Xiao-Gimbutas and $\max(\mathrm{Re}(\lambda_{\mathbf{M_z}})) \sim 4 \cdot 10^{-3}$ for NCO. Using the NCO instead of the Shunn-Ham rule for the SP location did not influence the stability.



(a) Jaśkowiec-Sukumar rule (b) Xiao-Gimbutas rule (c) NCO rule

Figure 3.15. – Spectrum of matrix $\mathbf{M_z}$ for unstable SDRT_2 schemes on tetrahedral elements, $(\theta_1, \theta_2) = (0, 0)$

For $p > 2$, the SDRT stability was studied using the position of physical interior FP as Shunn-Ham, NCO and Vioreanu-Rokhlin quadrature rules. None of these rules were able to lead to stable formulations, with a $\max(\mathrm{Re}(\lambda_{\mathbf{M_z}}))$ of $\sim 80, 400$ and 5000 for $p = 3, 4$ and 5 (respectively).

The SDRT scheme on tetrahedral elements is demonstrated as linearly stable under a Fourier analysis up to the third-order of accuracy, provided that the FP location is defined according to the Shunn-Ham quadrature rule.

Chapter 4

Linear Stability Analysis for the Coupled Time-Space Discretization

The linear stability analysis of the spatial discretization conducted in the previous chapter led to the conclusion that the Fourier analysis is the proper way to determine an SD scheme stability. Under a Fourier analysis, the SDRT spatial discretization on triangles was shown as unstable for orders of accuracy higher than four. However, the positive eigenvalues real part value of the spatial operator spectrum are of small order.
This chapter extends the Fourier analysis to the study of the coupled time-space discretization to address two points. The first one examines the impact in terms of stability of the time integration scheme for spatially stable SDRT schemes. The aim is to ensure that the full discretization remains stable. The second one addresses the ability of time integration schemes to stabilize spatially unstable SDRT schemes through their numerical dissipation properties.
First, explicit Runge-Kutta time integration methods are presented and applied to the semi-discretized SD matrix form. The stability of the SDRT spatial scheme on triangles coupled with different Runge-Kutta methods is then analyzed for spatially stable ($SDRT_2$ and $SDRT_3$) and spatially unstable schemes ($SDRT_4$ and $SDRT_5$). Finally, the behavior of spatially stable $SDRT_1$ and $SDRT_2$ schemes on tetrahedral elements coupled with temporal schemes is studied.

Overview

4.1. Temporal Integration **66**
4.1.1. Runge-Kutta Integration Methods 66
4.1.2. Matrix Form of the Coupled Time-Space Discretization 67
4.1.3. Fourier Analysis of the Coupled Time-Space Discretization 67
4.2. Triangular Elements **69**
4.2.1. Spatially Stable Schemes: $SDRT_2$ and $SDRT_3$ 69
4.2.2. Spatially Unstable Schemes: $SDRT_4$ and $SDRT_5$ 73
4.3. Tetrahedral Elements **80**

4.1. Temporal Integration

4.1.1. Runge-Kutta Integration Methods

Let us considering a differential equation:

$$\frac{\partial u}{\partial t} = R(u). \tag{4.1}$$

A general m-stage Runge-Kutta (RK) method for Eq. (4.1) can be written as in [186]:

$$\begin{cases} u^{(l)} & = \sum_{k=0}^{l-1} \left(\alpha_{lk} u^{(k)} + \Delta t \beta_{lk} R(u^{(k)}) \right), \quad l = 1, ..., m, \\ u^{(0)} & = u^{(n)}, \quad u^{(m)} = u^{(n+1)}, \end{cases} \tag{4.2}$$

where, in the case of Eq. (3.1), $R(u^{(k)}) = -\nabla \cdot \mathbf{f}(u^{(k)})$.

An alternative form of Eq. (4.2) can be written as in [187]:

$$u^{(n+1)} = u^n + \sum_{j=1}^{m} \gamma_j \, \Delta t^j \, \frac{\partial^j u^n}{\partial t^j}. \tag{4.3}$$

Several time integration schemes are considered in this chapter and are listed below. For each scheme, the associated coefficients are given in Appendix F.

- The family of Total Variation Diminishing (TVD) time discretization [186] introduced by Gottlieb and Shu, later called Strong Stability Preserving (SSP) schemes [188]. The nonlinear stability property of these methods makes them particularly appropriated for the time-integration of hyperbolic partial differential equations. The class of SSP schemes was extended by Spiteri and Ruuth in [189]. The optimal m-stage SSP RK method of order n will be denoted **SSP*msno***. Coefficients (α, β) are given in Table F.1 for 2-stage to 4-stage schemes, and in Table F.2 for 5-stage schemes.
- The 4-stage RK scheme introduced by Jameson in [190] for multigrid calculations, denoted **RK4J** in this work. The set of coefficients is chosen to improve the damping of high frequency modes. They are given in Table F.1.
- The standard explicit 4-stage RK algorithm, denoted **RKs4s**. The low-storage formulation based on Eq. (4.3) is given in Table F.3.
- The low storage RK schemes given by Bogey and Bailly in [187], which were optimized to ensure low dissipation and low dispersion properties, denoted **RKo5s** and **RKo6s**. The formulation is based on Eq. (4.3) and the associated coefficients γ are given in Table F.3.

4.1.2. Matrix Form of the Coupled Time-Space Discretization

The system of linear differential equations given by Eq. (3.9) is time-integrated. The matrix form of the time-integrated Eq. (3.9) using Eq. (4.2) is:

$$\begin{cases} \mathbf{G}^{(l)} & = \sum_{k=0}^{l-1} \left(\alpha_{lk}\mathbf{I} + \Delta t \beta_{lk}\mathbf{M}\right)\mathbf{G}^{(k)}, \quad l = 1, ..., m, \\ \mathbf{G}^{(0)} & = \mathbf{I}, \quad \hat{\mathbf{U}}^{(n+1)} = \mathbf{G}\,\hat{\mathbf{U}}^{(n)}, \end{cases} \tag{4.4}$$

where $\hat{\mathbf{U}}^{(n)}$ is $\hat{\mathbf{U}}_j^i$ at time n. The amplification factor of the coupled time-space discretization from time n to $n+1$ is thus given by $\mathbf{G}^{(m)}$ which will simply denoted $\mathbf{G}$.
Using the formulation given by Eq. (4.3), the matrix form is:

$$\begin{aligned} \hat{\mathbf{U}}^{(n+1)} &= \left(\mathbf{I} + \sum_{j=1}^{m} \gamma_j \Delta t^j \mathbf{M}^j\right)\hat{\mathbf{U}}^{(n)} \\ \Leftrightarrow \hat{\mathbf{U}}^{(n+1)} &= \mathbf{G}\hat{\mathbf{U}}^{(n)}. \end{aligned} \tag{4.5}$$

4.1.3. Fourier Analysis of the Coupled Time-Space Discretization

The semi-discretized matrix form containing the planar harmonic wave given by Eq. (3.25) integrated in time using Eq. (4.4) is:

$$\begin{cases} \mathbf{G}^{(l)} & = \sum_{k=0}^{l-1} \left(\alpha_{lk}\mathbf{I} + \nu \beta_{lk}\mathbf{M}_{\mathrm{z}}\right)\mathbf{G}^{(k)}, \quad l = 1, ..., m, \\ \mathbf{G}^{(0)} & = \mathbf{I}, \quad \tilde{\mathbf{U}}^{(n+1)} = \mathbf{G}\,\tilde{\mathbf{U}}^{(n)}, \end{cases} \tag{4.6}$$

where $\mathbf{G} = \mathbf{G}^{(m)}$ and ν is the CFL number defined by:

$$\nu = \frac{||\mathbf{c}||\Delta t}{\Delta x} \tag{4.7}$$

Using Eq. (4.5), it becomes:

$$\begin{aligned} \tilde{\mathbf{U}}^{(n+1)} &= \left(\mathbf{I} + \sum_{j=1}^{m} \gamma_j \nu^j \mathbf{M}_{\mathrm{z}}^j\right)\tilde{\mathbf{U}}^{(n)} \\ \Leftrightarrow \tilde{\mathbf{U}}^{(n+1)} &= \mathbf{G}\tilde{\mathbf{U}}^{(n)}. \end{aligned} \tag{4.8}$$

Note that the CFL expression given by Eq. (Eq. (4.7)) is the classical definition of the CFL number. To compare CFL numbers used in high-order discontinuous methods with classical methods (like Finite Volume, Finite Element or Finite Difference), one can introduced an equivalent CFL number $\hat{\nu}$ defined as $\hat{\nu} = (p+1)\nu$ [133]. This equivalent CFL number makes sense in the one-dimensional case since $(p+1)$ is a length scale corresponding to the mean distance between two adjacent SP. However, this definition is not necessarily the most adequate one on triangles, as shown by Chalmers and Krivodonova [191] for the DG method. To the author' knowledge, there is no consensus on the definition on an equivalent CFL number for high-order discontinuous methods on simplex cells. The classical CFL definition is thus preferred in the manuscript.

The stability condition on the coupled time-space discretization is thus obtained by requiring that the amplitude of any harmonic does not grow in time, i.e.:

$$|\mathbf{G}| = \left| \frac{\tilde{\mathbf{U}}^{(n+1)}}{\tilde{\mathbf{U}}^{(n)}} \right| \leq 1. \tag{4.9}$$

In other words, to ensure a stable discretization, the spectral radius of the matrix $\mathbf{G}$, denoted $\rho_{\mathbf{G}}$ should be lower than 1, meaning that all the eigenvalues $\lambda_{\mathbf{G}}$ should be in the unit circle of the complex plane. The transfer matrix $\mathbf{G}$ between time steps n and $n+1$ is the amplification factor (or the Fourier symbol) of the full discretization.

4.2. Triangular Elements

The amplification matrix $\mathbf{G}$ depends on the SDRT spatial operator through the matrix $\mathbf{M}_z$ and on the CFL number $\nu = \|\mathbf{c}\|\Delta t/\Delta x$. The stability of the coupled time-space discretization using Fourier analysis is studied depending on the CFL number. The interior FP locations are taken as Williams-Shunn-Jameson [174] quadrature points. For each different temporal schemes introduced in section 4.1.1, the spectral radius $\rho_{\mathbf{G}}$ is computed as a function of the CFL number for the following parameters:

- the advection angle $\theta \in [0, 2\pi]$,
- the grid frequency $\kappa \in [-\pi, \pi]$,
- the harmonic plane orientation $\vartheta \in [0, 2\pi]$,
- the skew angle $\mu = (\pi/2, \pi/3, \pi/4)$.

The stability limit is determined by increasing the CFL number incrementally until the spectral radius $\rho_{\mathbf{G}}$ becomes greater than 1.

4.2.1. Spatially Stable Schemes: SDRT$_2$ and SDRT$_3$

First, the Fourier analysis of SDRT schemes proven as spatially stable, i.e. SDRT$_2$ and SDRT$_3$, is conducted. The aim is to study the stability properties of the coupled time-space discretizations.

4.2.1.1. Suitable Temporal Schemes

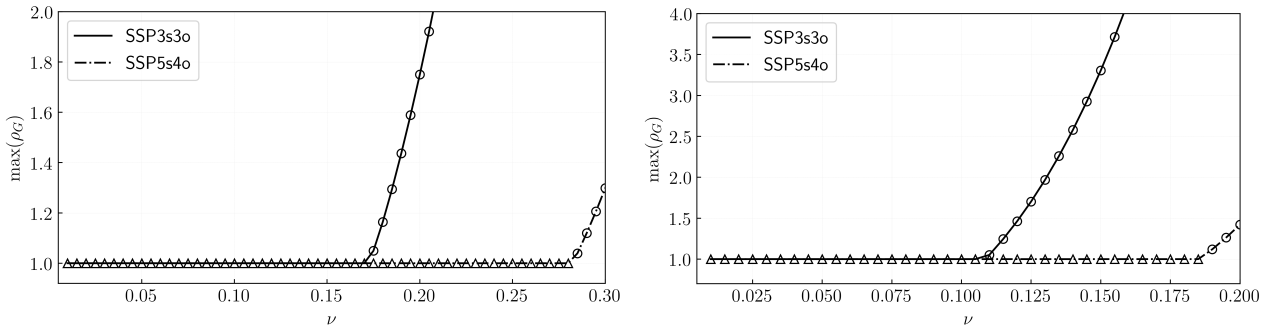
The Fourier analysis of the fully discretized linear advection equation is carried using temporal schemes introduced in section 4.1.1. For a large number of temporal schemes, the stability condition $\rho_{\mathbf{G}} \leq 1$ is verified up to a specific CFL value. Those schemes are thus suitable with the SDRT$_2$ and SDRT$_3$ spatial discretizations. Their stability limits are summarized in Table 4.1. The CFL numbers were determined numerically using the following parameters:

- $\theta \in [0, 2\pi], \Delta\theta = \pi/8$,
- $\kappa \in [-\pi, \pi], \Delta\kappa = \pi/8$,
- $\vartheta \in [0, 2\pi], \Delta\vartheta = \pi/8$,
- $\mu = (\pi/2, \pi/3, \pi/4)$.

Temporal scheme	SDRT$_2$	SDRT$_3$
SSP3s3o	0.172	0.108
SSP4s3o	0.252	0.166
SSP5s3o	0.330	0.229
SSP5s4o	0.282	0.185
RKs4s	0.192	0.120
RKo5s	0.231	0.144
RKo6s	0.286	0.179
RK4J	0.179	0.123

Table 4.1. – CFL stability limits for SDRT$_2$ and SDRT$_3$ on triangles coupled with different temporal schemes

The stability of the full discretization can be highlighted through the behavior of $\max(\rho_{\mathbf{G}})$ as the CFL number increases. This evolution is plotted on Fig. 4.1 for two temporal schemes - SSP3s3o and SSP5s4o - using SDRT_2 (Fig. 4.1a) and SDRT_3 (Fig. 4.1b). To ease the visualization, each $\max(\rho_{\mathbf{G}})$ value is associated with a marker. If $\max(\rho_{\mathbf{G}}) \leq 1$ (respectively $\max(\rho_{\mathbf{G}}) > 1$), the associated marker is a triangle (respectively a circle). From Fig. 4.1, it can be observed that the coupled time-space discretization is stable up to a given CFL number. Precise values of the maximum allowable CFL numbers associated with SSP3s3o and SSP5s4o schemes can be retrieved in Table 4.1.



(a) SDRT_2 (b) SDRT_3

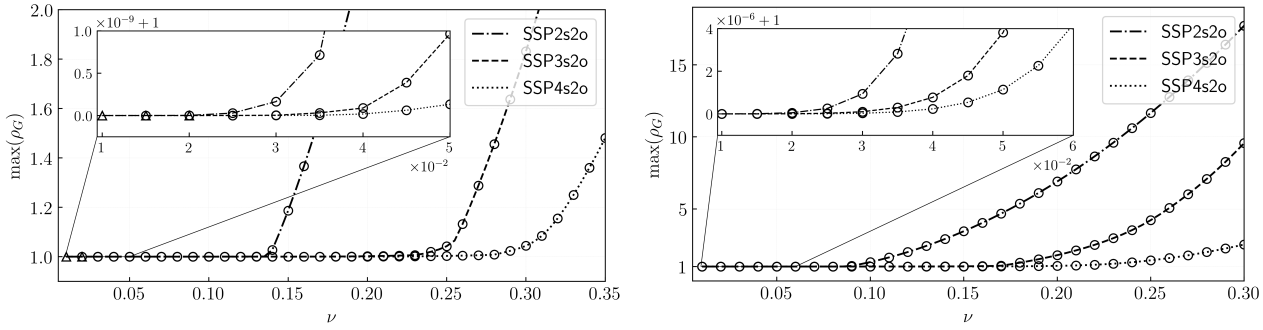
Figure 4.1. – Stability behavior of the coupled time-space discretizations for spatially stable SDRT schemes on triangles using SSP3s3o and SSP5s4o temporal schemes

4.2.1.2. Highlight of a Severe Stability Condition for Second-Order Temporal Schemes

While conducting the Fourier analysis, it was noted that a stable SDRT discretization can lead to an unstable coupled time-space discretization for a given temporal schemes for usual CFL numbers. This instability was noted by Balan *et al.* [142] when using SSP2s2o coupled with a SDRT_2 or SDRT_3, where a CFL number ensuring $\max(\rho_{\mathbf{G}}) \leq 1$ could not be determined. The behavior of $\max(\rho_{\mathbf{G}})$ as a function of the CFL number is plotted on Fig. 4.2 for different second-order temporal schemes - SSP2s2o, SSP3s2o and SSP4s2o - using SDRT_2 and SDRT_3.

For SDRT_2 (Fig. 4.2a), the triangular marker indicated that all temporal schemes are stable for $\nu \leq 0.01$. However, each scheme leads to unstable discretization for higher CFL values. When using SSP2s2o, $\max(\rho_{\mathbf{G}})$ is strictly higher than 1 for $\nu = 0.015$ with a value of approximately $1 + 2 \cdot 10^{-13}$, denoted by a circle. In fact, as the CFL number increases, this value grows up to $\sim 1 + 9 \cdot 10^{-4}$ (for $\nu = 0.135$) before it becomes noticeable on the plot. Using SSP3s2o (respectively SSP4s2o), $\max(\rho_{\mathbf{G}}) \leq 1$ is ensured until $\nu = 0.015$ (respectively $\nu = 0.02$).

For SDRT_3 (Fig. 4.2b), the three different second-order temporal schemes lead to $\max(\rho_{\mathbf{G}}) > 1$ even for $\nu = 0.01$. This extremely small stability limit makes the second-order temporal schemes presented here impractical.



(a) SDRT_2 (b) SDRT_3

Figure 4.2. – Stability behavior of the coupled time-space discretizations for spatially stable SDRT schemes on triangles using second-order temporal schemes

This results is verified numerically by considering the 2D linear advection of a sine. The initial solution of Eq. (3.1) is taken as:

$$u(x, y, 0) = \sin\left(\pi(x + y)\right), \tag{4.10}$$

and the velocity vector is $\mathbf{c} = (\cos\theta, \sin\theta)^\top$ where $\theta = \pi/8$. The computational domain is $\Omega = [-1, 1] \times [-1, 1]$ and periodic boundary conditions are used in both x and y directions. At interfaces, an upwind flux is used as the numerical flux. Following the notations introduced in section 3.3.3, Eq. (3.1) is solved on a regular mesh obtained using $N_x = N_y = 5$ and $\mu = \pi/2$. The initial solution is shown in Fig. 4.3. The computation is run using SDRT_2 associated with SSP2s2o and SSP3s3o. In both case, the CFL number is $\nu = 0.1$. Figure 4.4 shows the evolution of the solution maximum absolute value at SP obtained using SSP2s2o and SSP3s3o. At each time, the maximum value can be different depending on the location of the SP capturing the maximum value. Here, the interval in which different maximum values are included is small compared to the scale of $\max|u|$, leading to a thick straight line. In Fig. 4.4a, it can be clearly seen that the solution obtained using SSP2s2o grows up when the time increases, with a amplification of ~ 5 times the initial solution at $t_f = 10^4$ s. When using SSP3s3o (Fig. 4.4b), no amplification is observed: $\max|u|$ shows a decreasing behavior. The solution is lost due to numerical dissipation, which is expected since a very coarse mesh was used with a third-order spatial discretization.

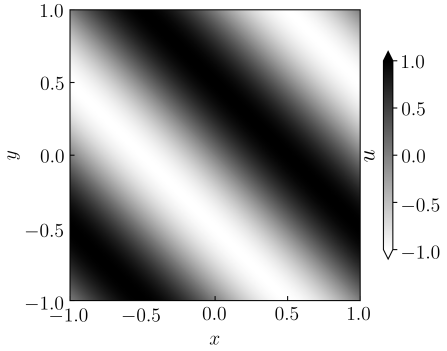
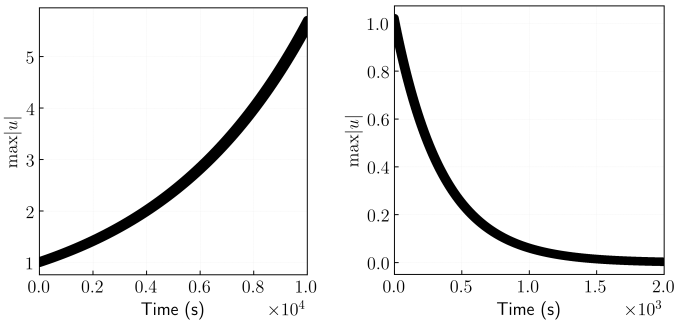


Figure 4.3. – Solution initialization for the sine advection



(a) SSP2s2o (b) SSP3s3o

Figure 4.4. – Evolution of $\max|u|$ using SDRT_2 on triangular mesh, $\theta = \pi/8$, $\nu = 0.1$

The analysis of the coupled time-space discretization for spatially stable SDRT schemes have highlighted the importance of the effect of temporal integration. Temporal schemes of the second-order SSP family associated with spatially stable SDRT schemes have revealed positive values of the amplification factor spectral radius for a CFL number $\nu > 0.02$. Even if positive values are of small order, they were shown as leading to an unstable computation. The severe condition on the CFL number should thus be respected but would lead to a high computational cost. Therefore, second-order SSP temporal schemes should be avoided for the benefit of time integration schemes leading to a stable time-space discretization. Suitable temporal schemes were given in section 4.2.1.1.
This section has demonstrated that to ensure stable computations, the Fourier analysis should be conducted on the fully (time and space) discretized equation.

4.2.2. Spatially Unstable Schemes: $SDRT_4$ and $SDRT_5$

For accuracy orders higher than 4, SDRT schemes were shown as spatially unstable in section 3.3.3. However, using the Williams-Shunn-Jameson (WSJ) quadrature points as the interior FP location, the positive eigenvalues are quite small ($\sim 10^{-5}$) for $SDRT_4$ and $SDRT_5$. The Fourier stability analysis of the coupled time-space discretization is thus conducted to determine if the numerical dissipation induced by the temporal scheme can stabilize the instability due to the spatial discretization. A first study is conducted for the advection angle $\theta = \pi/8$. This study aims to highlight the stability behaviors and to establish the coherence between the Fourier stability analysis and the numerical implementation of the SDRT method in JAGUAR.

4.2.2.1. Analysis for $\theta = \pi/8$

The Fourier analysis of the coupled time-space discretization is conducted using $\theta = \pi/8$, $\vartheta = \pi$, $\mu = \pi/2$ and $\kappa \in [-\pi, \pi]$. Results are plotted for the $SDRT_4$ (Fig.4.5) and $SDRT_5$ (Fig.4.6) space discretization coupled with different temporal schemes. For each scheme, the behavior of the maximum radius value is plotted for a CFL number $\nu \in [0.01, 0.1]$. A triangular marker indicates a stable scheme ($\max(\rho_{\mathbf{G}}) \leq 1$) whereas a circle is used to mark an unstable behavior ($\max(\rho_{\mathbf{G}}) > 1$). Three different behaviors are observed:

- The coupled time-space discretization is unstable for all CFL numbers and shows increasing $\max(\rho_{\mathbf{G}})$ values. Schemes associated with this behavior will be denoted as **unstable**.
- The coupled time-space discretization is unstable for small CFL numbers but becomes stable from a given CFL number until reaching a second CFL stability limit. Those schemes are **stable in an interval**, i.e. stable under a double CFL condition.
- The coupled time-space discretization is **stable** up to a CFL stability limit.

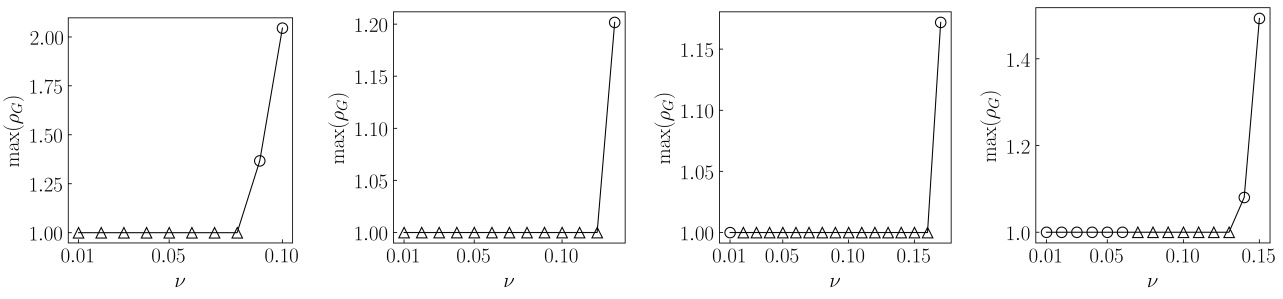
For both $SDRT_4$ and $SDRT_5$, the coupled time-space discretization is found unstable for all CFL numbers if the RKo6s is used. The RKo6s was optimized to minimize dispersion and dissipation error, thus it is not able to dissipate the instability due to the spatial SDRT discretization.

The RKo5s has also been designed as a low dissipation scheme, but it is shown in [187] that it leads to higher dissipation errors than the RKo6s (about one order of magnitude). The same study indicates that the RKs4s dissipation error is one order of magnitude higher than the RKo5s. Those two schemes are thus able to dissipate instabilities due to the spatial discretization for a high enough CFL number, before reaching a second CFL limit.

Concerning the SSP schemes, two of the third-order ones (SSP3s3o and SSP4s3o) were able to stabilize the $SDRT_4$ and $SDRT_5$ spatial discretization, even for small CFL numbers. The SSP5s3o however is unstable for $\nu < 0.02$ for both spatial discretizations. The four-order SSP scheme does not have enough dissipation for small CFL numbers for both spatial discretizations.

Additionally to the third-order SSP, the RK4J scheme also manage to dissipate the spatial discretization instabilities. This scheme was introduced by Jameson *'to improve the damping of high frequency modes'* [190], thus this result is consistent.

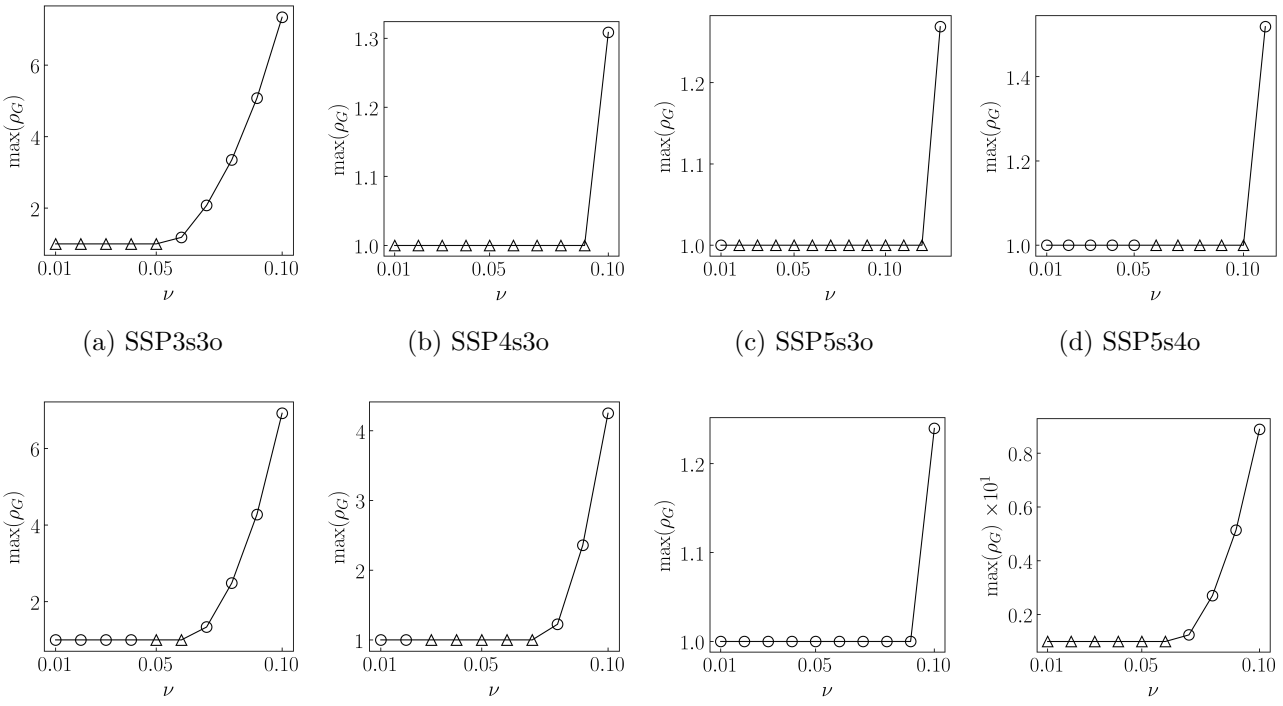
For the case $\theta = \pi/8$, it can be concluded that $SDRT_4$ and $SDRT_5$ spatial discretizations can be stabilized by the dissipation induced by the time-integration for specific temporal schemes.



(a) SSP3s3o (b) SSP4s3o (c) SSP5s3o (d) SSP5s4o

(e) RKs4s (f) RKo5s (g) RKo6s (h) RK4J

Figure 4.5. – Stability behavior of the coupled time-space discretizations for SDRT_4 on triangles for $\theta = \pi/8$



(a) SSP3s3o (b) SSP4s3o (c) SSP5s3o (d) SSP5s4o

(e) RKs4s (f) RKo5s (g) RKo6s (h) RK4J

Figure 4.6. – Stability behavior of the coupled time-space discretizations for SDRT_5 on triangles for $\theta = \pi/8$

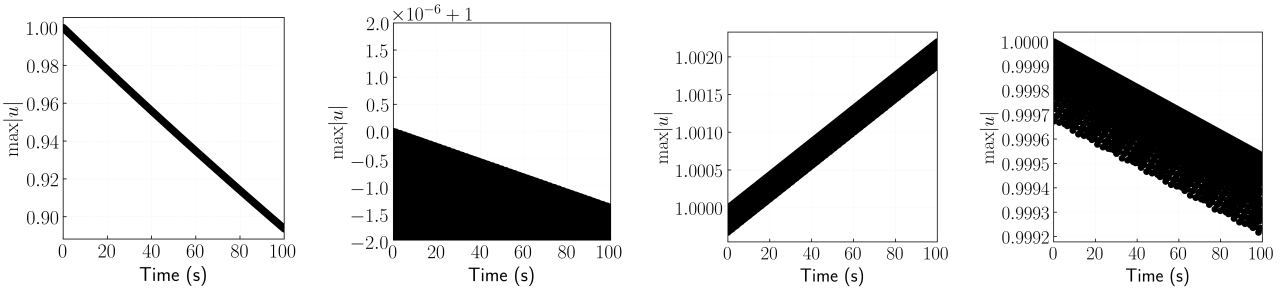
A summary of the stability limits in the case $\theta = \pi/8$ is given in Table 4.2. One could note that results obtained using SSP2s2o, SSP3s2o and SSP4s2o have not been presented in this section. The Fourier analysis of those schemes was conducted and showed an unstable behavior for all CFL numbers. Since it has already been underlined that those second-order temporal schemes could lead to unstable computations, even for spatially stable SDRT discretization, and given the large amount of studied temporal schemes, they were voluntary excluded from the study.

	$SDRT_4$		$SDRT_5$	
	Temporal Scheme	ν	Temporal Scheme	ν
Unstable	RKo6s	-	RKo6s	-
	SSP2s2o	-	SSP2s2o	-
	SSP3s2o	-	SSP3s2o	-
	SSP4s2o	-	SSP4s2o	-
Stable in an interval	SSP5s4o	$[0.07, 0.13]$	SSP5s4o	$[0.06, 0.1]$
	RKs4s	$[0.06, 0.09]$	RKs4s	$[0.05, 0.06]$
	RKo5s	$[0.03, 0.11]$	RKo5s	$[0.03, 0.07]$
	SSP5s3o	$[0.02, 0.16]$	SSP5s3o	$[0.02, 0.12]$
Stable	RK4J	0.08	RK4J	0.06
	SSP3s3o	0.08	SSP3s3o	0.05
	SSP4s3o	0.12	SSP4s3o	0.09

Table 4.2. – CFL stability limits for $SDRT_4$ and $SDRT_5$ on triangles coupled with different temporal schemes, $\theta = \pi/8$

To verify the consistency of the Fourier analysis results, a numerical study considering the advection of a sine introduced in section 4.2.1.2 is conducted. The computation is run using $SDRT_5$ on a mesh associated with the properties $N_x = N_y = 15, \mu = \pi/2$. Several temporal schemes are studied. First, the RK4J and the SSP4s3o, i.e. two schemes demonstrated as stable by the Fourier analysis for $\theta = \pi/8$ for all CFL numbers. The computation is run with a CFL number $\nu = 0.01$. The SSP5s3o, which was then demonstrated as stable in an interval of CFL, is used. Two computations are run using $\nu = 0.01$ (unstable) and $\nu = 0.1$ (stable). The stability of the computation is determined by monitoring the maximum absolute value of the solution $\max|u|$ at SP. Each computation is run up to $t_f = 100s$.

Figure 4.7 shows the evolution of $\max|u|$ at SP. As in the previous section, the maximum value can be different depending on the location of the SP capturing the maximum value at each time, leading to a thick line representation of $\max|u|$. The simulations using RK4J (Fig. 4.7a) and SSP4s3o (Fig. 4.7b) with $\nu = 0.01$ show that the maximum absolute value is decreasing when the time increases, indicating stable computations. It can be noted that using RK4J, the value of $\max|u|$ has decreased by $\sim 10\%$ of its value whereas using SSP4s3o allows one to preserve $\sim 99.99\%$ of the initial value. Results of the use of SSP5s3o are plotted for $\nu = 0.01$ in Fig. 4.7c and for $\nu = 0.1$ in Fig. 4.7d. As predicted by the Fourier analysis, using $\nu = 0.01$ leads to an unstable computation (amplification of the solution value) whereas using $\nu = 0.1$ gives decreasing solution values. At $t_f = 100\,\mathrm{s}$, $\sim 99.95\%$ of the solution is preserved. The numerical study results are consistent with the stability behavior predicted by the Fourier analysis.



(a) RK4J, $\nu = 0.01$ (b) SSP4s3o, $\nu = 0.01$ (c) SSP5s3o, $\nu = 0.01$ (d) SSP5s3o, $\nu = 0.1$

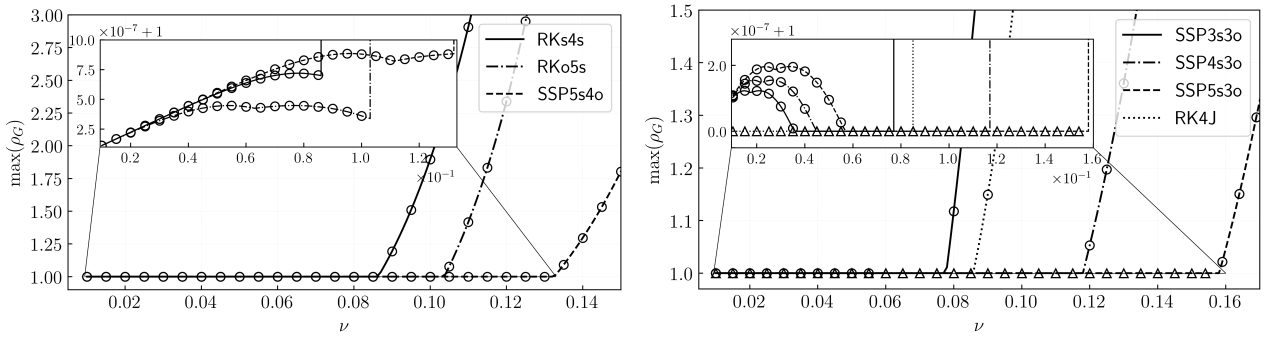
Figure 4.7. – Evolution of $\max|u|$ using SDRT_5 on triangles for $\theta = \pi/8$

4.2.2.2. Analysis in the General Case

The Fourier analysis is now extended to the general case, i.e. for $\theta \in [0, 2\pi]$. The following parameters are considered:

- $\theta \in [0, 2\pi], \Delta\theta = \pi/8$,
- $\kappa \in [0, \pi], \Delta\kappa = \pi/32$,
- $\vartheta \in [0, 2\pi], \Delta\vartheta = \pi/8$,
- $\mu = (\pi/2, \pi/3, \pi/4)$.

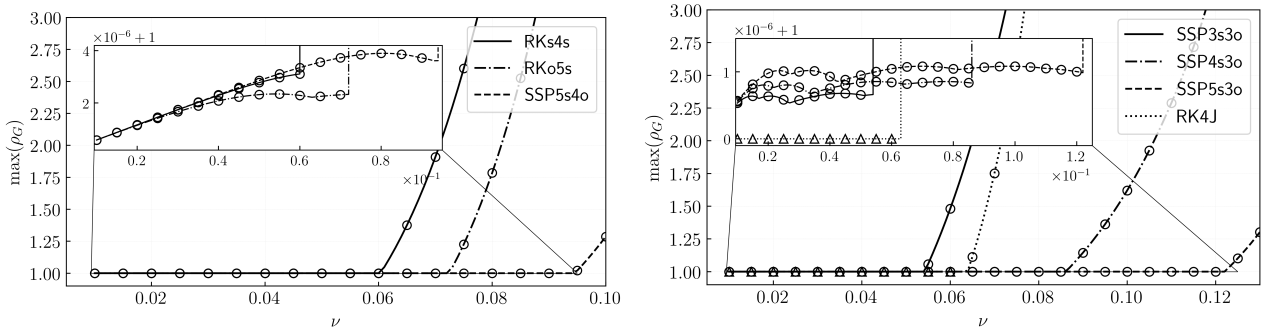
The grid frequency range is reduced to $[0, \pi]$ due to symmetry and $\Delta\kappa$ is taken smaller. Second-order temporal schemes and the RKo6s scheme, which were shown as unstable for $\theta = \pi/8$, will not be considered in the general analysis. The Fourier analysis results are plotted for SDRT_4 in Fig. 4.8. In Fig.4.8a, SSP5s4o, RKs4s and RKo5s show a diminishing - but still unstable - behavior. The spectral radius is of order $1+10^{-7}$ and this value remains bounded until a specific CFL ($\nu = 0.133$ for SSP5s4o, $\nu = 0.087$ for RKs4s and $\nu = 0.104$ for RKo5s) where $\max(\rho_{\mathbf{G}})$ becomes of order 10^{-2}. In Fig. 4.8b, the behavior of SSP3s3o, SSP4s3o, SSP5s3o and RK4J is plotted. For third-order RK schemes, the coupled time-space discretization is unstable ($\max(\rho_{\mathbf{G}}) > 1$, marked by circles) for small CFL numbers but becomes stable ($\max(\rho_{\mathbf{G}}) \leq 1$, marked by triangles) when the CFL number increases. The discretization then becomes unstable with higher spectral radius values when the upper CFL limit is attained. Finally, for the RK4J, the coupled time-space discretization is stable up to $\nu = 0.085$.



(a) (b)

Figure 4.8. – Stability behavior of the coupled time-space discretizations for SDRT_4 on triangles for $\theta \in [0, 2\pi]$

For $SDRT_5$, the Fourier analysis results are plotted in Fig. 4.9. The RK4J is stable up to $\nu = 0.063$. All other temporal schemes reveal an unstable behavior for all CFL number with $\max(\rho_{\mathbf{G}})$ of order $\sim 1 + 10^{-6}$. Table. 4.3 gives a summary of the Fourier analysis conclusions in the general case.



(a) (b)

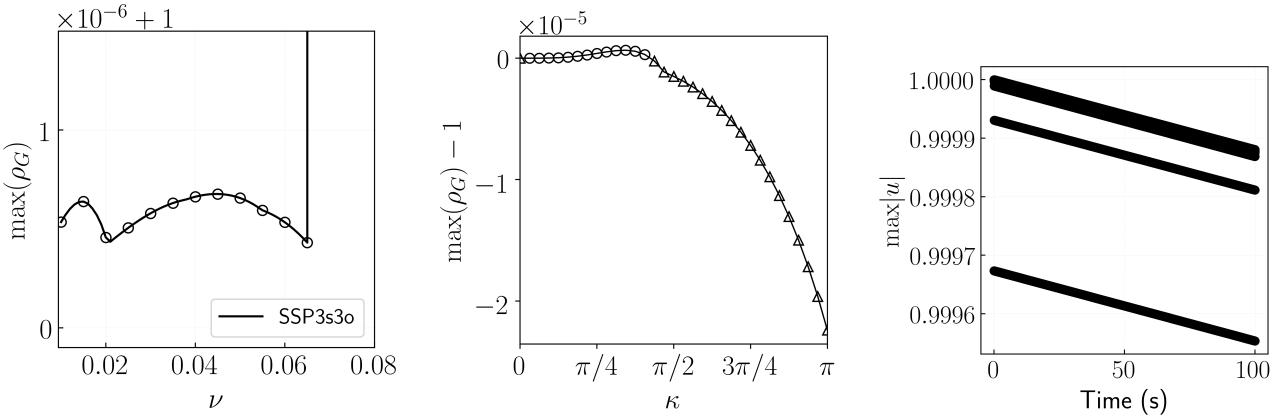
Figure 4.9. – Stability behavior of the coupled time-space discretizations for $SDRT_5$ on triangles for $\theta \in [0, 2\pi]$

	$SDRT_4$		$SDRT_5$	
	Temporal Scheme	ν	Temporal Scheme	ν
Stable in an interval	SSP3s3o	$[0.036, 0.077]$	-	-
	SSP4s3o	$[0.045, 0.117]$	-	-
	SSP5s3o	$[0.06, 0.158]$	-	-
Stable	RK4J	0.08	RK4J	0.06

Table 4.3. – CFL stability limits for $SDRT_4$ and $SDRT_5$ on triangles coupled with different temporal schemes, $\theta \in [0, 2\pi]$

4.2.2.3. Analysis for $SDRT_5$, $\theta = 0$

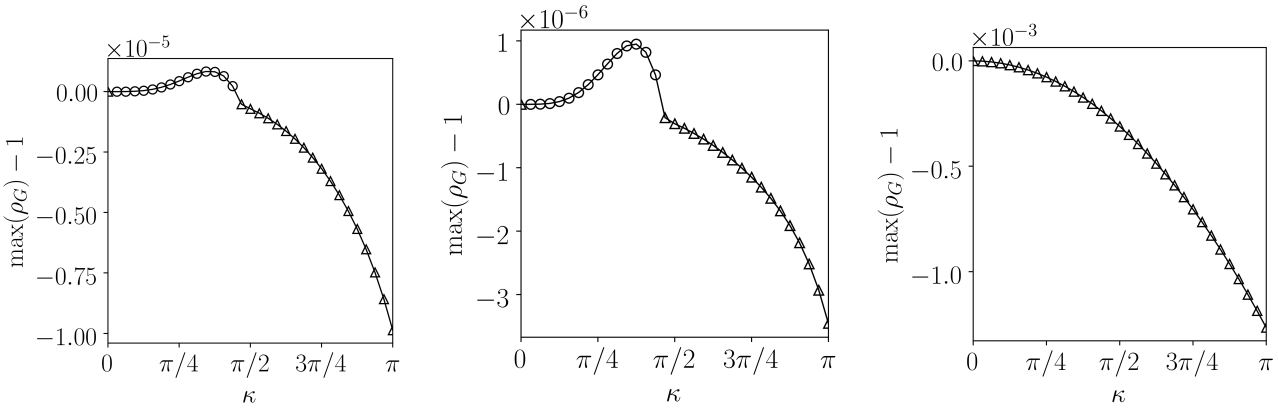
It was noted that the instability of the coupled time-space discretization is revealed for specific values of the advection angle ($\theta = 0$ or $\theta = \pi/2$ for example). A test case using $SDRT_5$ coupled with SSP3s3o using $\theta = \vartheta = 0$ and $\nu = 0.05$ is presented on Fig. 4.10. First, the maximum radius is plotted as a function of the CFL number on Fig. 4.10a. It can be seen that the coupled time-space discretization is unstable. The CFL number is then fixed to 0.05 and the maximum radius is plotted as a function of the grid frequency on Fig. 4.10b. Radius value higher than 1 come from $\kappa \in]0, \pi/2[$, but the discretization is stable for $\kappa \in [\pi/2, \pi]$. The advection of a sine is computed using $\nu = 0.05$ on a mesh associated with the properties $N_x = N_y = 15, \mu = \pi/2$, and the maximum absolute value of the solution at SP is plotted in Fig. 4.10c up to $t_f = 100$ s. As in previous computations, the maximum value varies depending on the location on the SP: due to the extrapolation step, some SP can not capture the maximum value of u. Since a periodic problem on a regular mesh is considered, the maximum absolute value of u is obtained from the same SP every N_{SP} iteration. This is highlighted by the formation of different parallel straight lines. All of them are decreasing, which indicate that the computation is stable. The same decreasing behavior is observed for $N_x = N_y \in [5, 20, 51, 100]$.



(a) Stability behavior (b) Stability behavior, $\nu = 0.05$ (c) Sine advection, $\nu = 0.05$

Figure 4.10. – Stability study of SDRT_5 coupled with SSP3s3o using $\theta = \vartheta = 0$ on triangular elements

The same stability study is conducted for SSP4s3o, SSP5s3o and RK4J (Fig. 4.11). For third-order schemes, the same behavior is observed: the scheme is unstable for $\kappa \in]0, \pi/2[$ and becomes stable for $\kappa \in [\pi/2, \pi]$. For $\kappa \in [\pi/2, \pi]$, dissipation errors are clearly reduced when the number of stages increases. Using the RK4J, the computation is stable for all grid frequencies. However, the dissipation error is of order 10^{-3}. Plots associated with RKs4s, RKo5s and SSP5s4o are not presented here since they all led to $\max(\rho_{\mathbf{G}}) > 1$ for $\kappa \in]0, \pi]$.



(a) SSP4s3o (b) SSP5s3o (c) RK4J

Figure 4.11. – Study of SDRT_5 on triangles coupled with different temporal schemes using $\theta = \vartheta = 0$, $\nu = 0.05$

4.2.2.4. Analysis of Other FP Locations

This study was conducted on alternative sets of interior FP considered in Chapter 3, which locations were taken according to other existing quadrature rules or determined using the Differential Evolution optimization process of the FNC analysis. For $SDRT_4$, the interior FP location taken as the Vioreanu-Rokhlin quadrature rule leads to an unstable time-space discretization in the general case $\theta \in [0, \pi]$ for all temporal schemes. For $SDRT_5$, the Witherden-Vincent rule was studied since the maximum value of $\max(\mathrm{Re}(\lambda_{\mathbf{M}_z}))$ is of order 10^{-5}. Other quadrature rules led to higher values and were then not considered. The coupled time-space discretization was found stable using Witherden-Vincent quadrature points as the interior FP locations if the spatial scheme is associated with the RK4J temporal scheme, with a maximum CFL limit $\nu = 0.06$. Both quadrature rules (Williams-Shunn-Jameson and Witherden-Vincent) thus lead to the same stability limit. Finally, the interior FP sets determined from the Differential Evolution optimization process of the FNC analysis led to unstable discretizations for all temporal schemes.

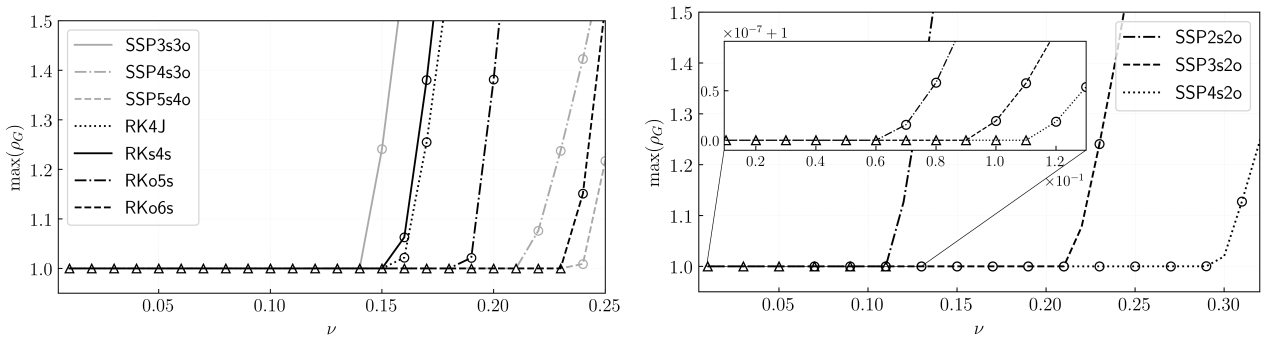
In section 3.3.3, the unstable behavior of the spatial SDRT operator on triangular grids for $p > 3$ was shown. In this section, the Fourier stability analysis of the SDRT spatial discretization previously detailed was extended for $p = 4$ and $p = 5$ to the coupled time-space discretization to study the effect of the time integration scheme on the stability.
It was demonstrated that for $SDRT_4$ and $SDRT_5$, it is possible to obtain stable discretizations by using the proper time integration scheme. For both schemes, the RK4J scheme is the only one that was proven as leading to a stable time-space discretization for all advection angles and all grid frequencies. The numerical dissipation induced by the RK4J can dissipate spatial instability; however, this high numerical dissipation error also means that the RK4J scheme probably won't be an appropriate choice to obtain high-order accuracy for unsteady problems. This statement will be verified numerically in Chapter 6, section 6.1.2.
Third-order SSP schemes (SSP3s3o, SSP4s3o and SSP5s3o) can stabilize the $SDRT_4$ scheme if the CFL number is chosen in the stability interval, i.e. high enough to dissipate instability due to the spatial discretization, but small enough to ensure a stable discretization. For $SDRT_5$ coupled to a third-order SSP scheme, it was shown that the discretization is stable for $\theta = \pi/8$ but not for all advection angles. The particular case of $\theta = 0$ has revealed that the instability is due to small grid frequencies ($\kappa \in]0, \pi/2[$).
Finally, the study of other sets of interior FP indicates that the Williams-Shunn-Jameson quadrature rule remains the most adequate for interior FP location. The SDRT implementation based on the interior FP located at Williams-Shunn-Jameson quadrature points will now be denoted the $SDRT^{WSJ}$ scheme.

4.3. Tetrahedral Elements

The Fourier analysis of the coupled time-space discretization is now conducted on tetrahedral elements for SDRT_1 and SDRT_2. The interior FP locations are in both case taken as following the Shunn-Ham quadrature rule. As for the analysis on triangles, the spectral radius $\rho_{\mathbf{G}}$ is computed based on the CFL number and the spatial SDRT operator $\mathbf{M}_z$ defined by Eq. (3.41). Additionally to the SP and FP locations, the matrix $\mathbf{M}_z$ depends on the advection velocity defined by (θ_1, θ_2), the grid frequency κ and the harmonic wave orientation angles $(\vartheta_1, \vartheta_2)$. These parameters are taken as:

- $\theta_1 \in [0, 2\pi], \Delta\theta_1 = \pi/8$,
- $\theta_2 \in [0, 2\pi], \Delta\theta_1 = \pi/8$,
- $\kappa \in [0, \pi], \Delta\kappa = \pi/8$,
- $\vartheta_1 \in [0, 2\pi], \Delta\vartheta_1 = \pi/8$,
- $\vartheta_2 \in [0, 2\pi], \Delta\vartheta_2 = \pi/8$.

All temporal schemes introduced in section 4.1.1 are studied. For SDRT_1, the same behavior is observed for all temporal schemes: the coupled time-space discretization is stable up to a stability limit CFL number. For SDRT_2, stability analysis results are plotted in Fig. 4.12. The second-order SSP schemes (Fig. 4.12b) have been separated from other temporal schemes (Fig. 4.12a) because of their different stability behavior. Temporal schemes on Fig. 4.12a present an usual behavior: schemes are stable ($\max(\rho_{\mathbf{G}}) \leq 1$, triangle markers) up to a CFL stability limit. At first sight, it seems that the same stability behavior is observed for second-order SSP schemes (Fig. 4.12b) with a CFL limit ($\nu = 0.11$ for SSP2s2o, $\nu = 0.21$ for SSP3s2o, $\nu = 0.29$ for SSP4s2o) but the circle markers indicates that the scheme is unstable ($\max(\rho_{\mathbf{G}}) > 1$). At the specified CFL numbers, the associated $\max(\rho_{\mathbf{G}})$ values are actually of order 10^{-6} (SSP2s2o), 10^{-5} (SSP3s2o) and 10^{-4} (SSP4s2o). A closer view at small CFL numbers allows one to determine the proper stability limits: $\nu = 0.06$ for SSP2s2o, $\nu = 0.09$ for SSP3s2o and $\nu = 0.11$ for SSP4s2o. The CFL stability limits are summarized for all temporal schemes in Table 4.4.



(a) (b)

Figure 4.12. – Stability behavior of the coupled time-space discretizations for SDRT_2 on tetrahedron

Temporal scheme	$SDRT_1$	$SDRT_2$
SSP2s2o	0.18	0.06
SSP3s2o	0.33	0.09
SSP4s2o	0.47	0.11
SSP3s3o	0.22	0.14
SSP4s3o	0.33	0.21
SSP5s3o	0.45	0.29
SSP5s4o	0.37	0.23
RKs4s	0.25	0.15
RKo5s	0.30	0.18
RKo6s	0.38	0.23
RK4J	0.24	0.15

Table 4.4. – CFL stability limits for $SDRT_1$ and $SDRT_2$ on tetrahedron coupled with different temporal schemes

Chapter 5

Optimization Process Based on the Fourier Analysis

In Chapter 4, a linear stability analysis for the coupled time-space discretization was presented. The study especially examines the behavior of spatially unstable SDRT implementations using Williams-Shunn-Jameson quadrature points as interior FP, denoted as the SDRT^{WSJ} scheme. It was shown that those schemes can be stabilized if a proper time integration scheme is used. Two different solutions were highlighted. The first consists in using the RK4J temporal scheme: stability is ensured for all advection angles but the error associated with numerical dissipation is high. The second one consists in using third-order SSP temporal schemes, which are associated with better dissipation and dispersion properties. However, the stability of the coupled time-space discretization is subject to conditions: for SDRT_4, the CFL number must lie into a given range, whereas for SDRT_5, the stability is not ensured for all advection angles. The fact that this second possibility is strongly dependent on the advection angle or on a double condition on the CFL number makes it impracticable. It will thus not be considered anymore in the following.
For a general framework, these conditions seem to be too restrictive. To overcome this limitation, this chapter presents new sets of interior FP leading to stable SDRT formulations for $p > 3$, called SDRT^{OPT}. These sets of points are determined through an optimization process based on the Fourier analysis for the spatial discretization similar to the one presented in Chapter 3. The main difference is that this chapter considers an optimization algorithm based on the gradient descent method. The optimization process is first presented for triangular elements. Sets of interior FP leading to stable SDRT formulations are then given and the Fourier analysis results of the spatial discretization associated with different temporal schemes are shown. The optimization process is then applied to the SDRT_6 scheme and finally to tetrahedral elements.

Overview

5.1. Triangular Elements **84**
5.1.1. Optimization Algorithm 84
5.1.2. Spatially Stable SDRT_4 and SDRT_5 Formulations 87
5.1.3. Study for the SDRT_6 Scheme 92
5.2. Tetrahedral Elements **93**
5.2.1. Optimization Algorithm 93
5.2.2. Fourier Analysis of the Spatial Discretization 95
5.2.3. Influence of Flux Points Located on Faces 95

5.1. Triangular Elements

5.1.1. Optimization Algorithm

The Fourier analysis for the spatial SDRT discretization on triangular elements presented in section 3.3.3 is considered. It is recalled here that the Fourier analysis leads to the definition of the matrix $\mathbf{M}_z$, which is fully defined through:

- the polynomial degree p,
- the SP location,
- the FP location,
- the advection angle $\theta \in [0, 2\pi]$,
- the grid frequency $\kappa \in [-\pi, \pi]$,
- the harmonic plane orientation $\vartheta \in [0, 2\pi]$,
- the skew angle $\mu \in [0, \pi/2]$.

To determine spatially stable SDRT formulations for orders of accuracy higher than four, the Fourier analysis is used in an optimization problem. The function to minimize is the maximum of the real part of the matrix $\mathbf{M}_z$ eigenvalues and the optimization parameters are the interior FP locations.

The difference with the algorithm presented in section 3.3.3 is that here, the optimization process solves the problem of minimizing a function locally using a gradient descent method called the L-BFGS-B method from the SciPy library [179]. The L-BFGS-B algorithm is part of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithms, which are iterative methods for solving unconstrained nonlinear optimization problems. The descent direction is determined by preconditioning the gradient with curvature information. Contrary to the Differential Evolution algorithm, using this optimization algorithm allows taking into account an initial value for the optimization parameters. Additionally, the computational time needed to perform one iteration is way faster with a descent method algorithm. This is due to the fact that for one iteration, the Differential Evolution algorithm has to evolve the entire population; the function is evaluated $[(\text{Population size}) * (\text{Number of parameters})]$ times. As an example, the time needed for one iteration of the optimization algorithm based on the Fourier analysis for SDRT_4 is approximately 32 seconds using the Differential Evolution algorithm versus 7 seconds using the L-BFGS-B method.

The full algorithm is detailed by Algorithm 1 for the SDRT_4 scheme. First, the constant parameters are settled: the polynomial degree p is set to 4, the SP location is set to the position given by the 15-points Williams-Shunn-Jameson quadrature rule and the position of FP located on edges is set to Gauss-Chebyshev points. As detailed in section 3.1.1.2, the SP location have no impact on the scheme stability and the edge FP location in a triangle needs to be the same as the edge FP location in a quadrilateral (Gauss-Chebyshev points) to avoid using mortar techniques in hybrid computations. The interior FP coordinates are then parametrized by a set of coefficients to ensure symmetry. The parametrization is given in Appendix E for $p \in [\![4, 6]\!]$. The initial interior FP location, stored in x0, is chosen as the 10-points Williams-Shunn-Jameson quadrature rule, expressed using the optimization parameters. Bounds are given to ensure that all interior FP are included in the triangle. The optimization is then run: the *scipy.optimize.minimize* function based on the L-BFGS-B method called the function MAIN, using x0 as the initial interior FP location and taking bounds into account.

The function MAIN returns the maximum of all eigenvalues real part of the matrix $\mathbf{M}_z$. The interior FP coordinates are first computed based on the optimization parameters, which allows to compute the transfer matrix $\mathbf{T}$ and the differentiation matrix $\mathbf{D}$. Using symmetry properties, the matrix $\mathbf{M}_z$ is computed for:

- the advection angle $\theta \in [0, \pi], \Delta\theta = \pi/8$,
- the grid frequency $\kappa \in [0, \pi] \Delta\kappa = \pi/32$,
- the harmonic plane orientation $\vartheta \in [0, \pi], \Delta\vartheta = \pi/8$,
- the skew angle $\mu = \pi/2$.

Finally, the maximum of the real part of all the eigenvalues of $\mathbf{M}_z$, denoted rm, is returned and will be minimized by the optimization algorithm.

Algorithm 1 Fourier analysis optimization algorithm on triangles for SDRT_4

Constants

$p \leftarrow 4$

SP location: $\boldsymbol{\xi}^{SP}_{1:15} \leftarrow$ WSJ 15-points rule

Edge FP location: $\boldsymbol{\xi}^{FP}_{1:5} \leftarrow$ Gauss-Chebyshev points ▷ Edge 1

$\boldsymbol{\xi}^{FP}_{6:10} \leftarrow$ Gauss-Chebyshev points ▷ Edge 2

$\boldsymbol{\xi}^{FP}_{11:15} \leftarrow$ Gauss-Chebyshev points ▷ Edge 3

Optimization Parameters

$\alpha_1 = 0.333333333333333, \alpha_2 = 0.055564052669793$

$\beta_1 = 0.365789252254277, \gamma_1 = 0.112639085608754$

$\beta_2 = 0.704466288264107, \gamma_2 = 0.281977603613669$ ▷ WSJ 10-points rule

$\beta_3 = 0.929744459481616, \gamma_3 = 0.169338518004915$

$\beta_4 = 0.944435947330207, \gamma_4 = 0.416653920995311$

x0 $= (\alpha_1, \alpha_2, \beta_1, \beta_2, \beta_3, \beta_4, \gamma_1, \gamma_2, \gamma_3, \gamma_4)$

Bounds $= (\alpha_1, \alpha_2 \in]0, 0.5[, \beta_1, \beta_2, \beta_3, \beta_4, \gamma_1, \gamma_2, \gamma_3, \gamma_4 \in]0, 1[)$

Optimization Process

call scipy.optimize.minimize(MAIN, x0, Bounds, method='L-BFGS-B')

function MAIN

$\boldsymbol{\xi}_{16} = (\beta_4/2 + \gamma_4, \beta_4/2 - \gamma_4)$, $\boldsymbol{\xi}_{17} = (\beta_1/2 + \gamma_1, \beta_1/2 - \gamma_1)$

$\boldsymbol{\xi}_{18} = (\beta_1/2 - \gamma_1, \beta_1/2 + \gamma_1)$, $\boldsymbol{\xi}_{19} = (\alpha_2, \alpha_2)$

$\boldsymbol{\xi}_{20} = (\beta_3/2 - \gamma_3, \beta_3/2 + \gamma_3)$, $\boldsymbol{\xi}_{21} = (\beta_2/2 + \gamma_2, \beta_2/2 - \gamma_2)$

$\boldsymbol{\xi}_{22} = (\beta_4/2 - \gamma_4, \beta_4/2 + \gamma_4)$, $\boldsymbol{\xi}_{23} = (\beta_3/2 + \gamma_3, \beta_3/2 - \gamma_3)$

$\boldsymbol{\xi}_{24} = (\alpha_1, \alpha_1)$, $\boldsymbol{\xi}_{25} = (\beta_2/2 - \gamma_2, \beta_2/2 + \gamma_2)$

$\boldsymbol{\xi}^{FP}_{26:35} = \boldsymbol{\xi}^{FP}_{16:25}$

Compute Transfer Matrix $\mathbf{T}$

Compute Differentiation Matrix $\mathbf{D}$

for $\mu = \pi/2$ **do** ▷ Skew angle

for $\vartheta \in [0, \pi], \Delta\vartheta = \pi/8$ **do** ▷ Harmonic plane orientation

for $\theta \in [0, \pi], \Delta\theta = \pi/8$ **do** ▷ Advection angle

Compute $\mathbf{M}^{0,0}, \mathbf{M}^{-1,0}, \mathbf{M}^{+1,0}, \mathbf{M}^{0,-1}, \mathbf{M}^{0,+1}$

for $\kappa \in [0, \pi], \Delta\kappa = \pi/32$ **do** ▷ Grid frequency

Compute $\mathbf{M}_z$ using:

$$\begin{aligned}\mathbf{M}_z = -\Big[&\mathbf{M}^{0,0} + \mathbf{M}^{-1,0}\exp\Big(-I\kappa\cos\vartheta\Big)\\ &+ \mathbf{M}^{+1,0}\exp\Big(I\kappa\cos\vartheta\Big)\\ &+ \mathbf{M}^{0,-1}\exp\Big(-I\kappa(\cos\mu\cos\vartheta + \sin\mu\sin\vartheta)\Big)\\ &+ \mathbf{M}^{0,+1}\exp\Big(I\kappa(\cos\mu\cos\vartheta + \sin\mu\sin\vartheta)\Big)\Big]\end{aligned}$$

Compute $\max(\mathrm{Re}(\lambda_{\mathbf{M}_z}))$

rm $= \max(\mathrm{rm}, \max(\mathrm{Re}(\lambda_{\mathbf{M}_z})))$

end for

end for

end for

end for

return rm

end function

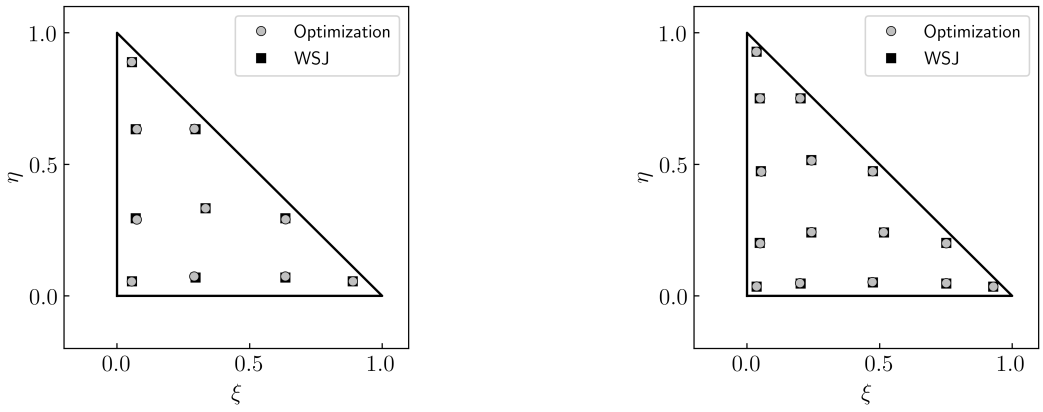
5.1.2. Spatially Stable $SDRT_4$ and $SDRT_5$ Formulations

5.1.2.1. Sets of Interior Flux Points

The optimization process based on the L-BFGS-B algorithm was able to determined spatially stable $SDRT_4$ and $SDRT_5$ formulations. Parameters leading to stable formulations are given in Table 5.1. It should be underlined that there is no proof of the uniqueness of the set of interior FP leading to stable SDRT formulations. The interior FP coordinates leading to stable SDRT formulations are actually very close to the coordinates given by the Williams-Shunn-Jameson quadrature rule. This was an expected result due to the local optimization process that looks for a stable formulation close from the initial guess. Both sets of points are compared in Fig. 5.1. However, as shown in the next section, stability conclusions are quite different.

	$SDRT_4$	
α_1	0.33366214220365053577666003548 1	–
α_2	0.05502032327765691427368111021 7	–
β_1, γ_1	0.36505900941934221748397249029 9	0.10825744697505322589048404324 8
β_2, γ_2	0.70838121841272838619119056602 4	0.28017810320268821122624558483 9
β_3, γ_3	0.92672898300009898253648543686 7	0.17186473732812543313563935498 7
β_4, γ_4	0.94480877497865967118428898174 9	0.41703166521315820913784477852 5
	$SDRT_5$	
α_1	0.03601638717092110059114773434 9	–
α_2	0.24288371116316528897094428884 9	–
α_3	0.47330280861806123260393519558 4	–
β_1, γ_1	0.24865327212126914213641271089 7	0.07537555948630439428548299929 4
β_2, γ_2	0.52610716826649672750448871738 6	0.20953863720661883296436656110 0
β_3, γ_3	0.75746307239073784600691396917 7	0.13620750036029358187583682138 2
β_4, γ_4	0.80019811864053436156751786256 8	0.35127172764319664066690052095 5
β_5, γ_5	0.95099538178119114029129832488 3	0.27556778867665415733156919486 6
β_6, γ_6	0.96387254267775313021360261700 5	0.44671648161944355059915778838 3

Table 5.1. – Coordinate parameters of interior FP determined using the optimization process on Fourier analysis



(a) $SDRT_4$ (b) $SDRT_5$

Figure 5.1. – Sets of FP determined using the optimization process on Fourier analysis compared to Williams-Shunn-Jameson sets

5.1.2.2. Fourier Analysis of the Spatial Discretization

The Fourier analysis of the spatial SDRT discretization based on the interior FP determined by the optimization process is conducted. The spectrum of $\mathbf{M}_z$ is first computed for:

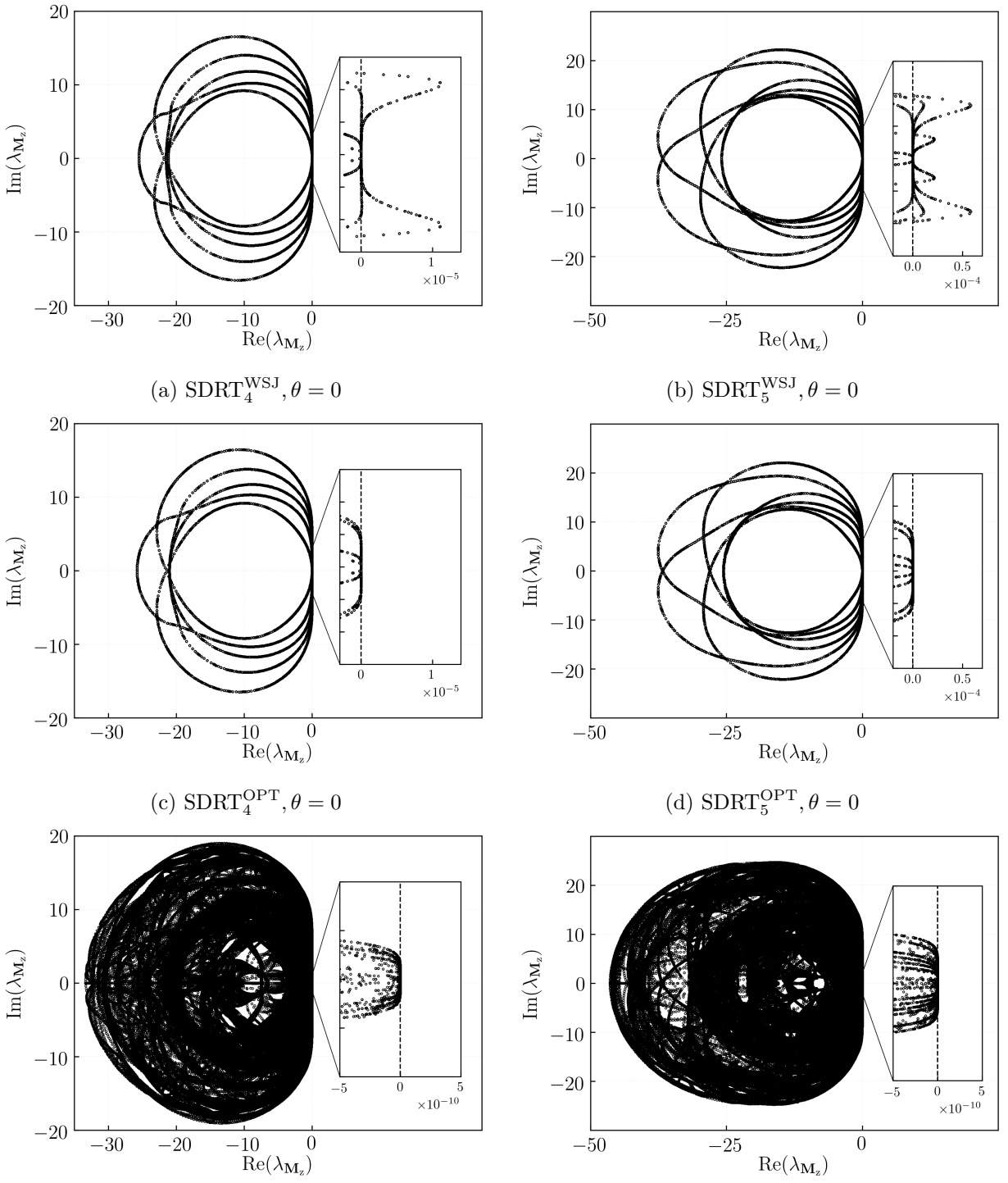
- the advection angle $\theta = 0$,
- the grid frequency $\kappa \in [-\pi, \pi]$,
- the harmonic plane orientation $\vartheta \in [0, 2\pi]$,
- the skew angle $\mu = \pi/2$.

These conditions are exactly the same as the ones used in section 3.3.3, where the SDRT^{WSJ} discretization was proven as unstable for $p = 4$ and $p = 5$. Spectra obtained for the unstable SDRT^{WSJ} scheme rule shown in Fig. 3.5 are shown again in Fig. 5.2a and Fig. 5.2b to easily compare the impact of the interior FP location. The spectrum of $\mathbf{M}_z$ using the interior FP determined by the optimization process is displayed in Fig. 5.2c for SDRT_4 and in Fig. 5.2d for SDRT_5. For each order, the Fourier footprint obtained using the SDRT^{OPT} is similar to the one obtained with the SDRT^{WSJ} scheme except that positive eigenvalues have been pushed to the negative side, leading to stable formulations.

Spectra are then computed in the general case, i.e. for:

- the advection angle $\theta \in [0, \pi]$,
- the grid frequency $\kappa \in [-\pi, \pi]$,
- the harmonic plane orientation $\vartheta \in [0, 2\pi]$,
- the skew angle $\mu \in [\pi/2, \pi/3, \pi/4]$.

Corresponding Fourier footprints are shown in Fig. 5.2e for SDRT_4 and in Fig. 5.2f for SDRT_5. From these spectra, the linear stability of the SDRT^{OPT} spatial discretization is clearly established since the real part of all eigenvalues is negative.



(a) $\mathrm{SDRT}_4^{\mathrm{WSJ}}, \theta = 0$

(b) $\mathrm{SDRT}_5^{\mathrm{WSJ}}, \theta = 0$

(c) $\mathrm{SDRT}_4^{\mathrm{OPT}}, \theta = 0$

(d) $\mathrm{SDRT}_5^{\mathrm{OPT}}, \theta = 0$

(e) $\mathrm{SDRT}_4^{\mathrm{OPT}}$, General Case

(f) $\mathrm{SDRT}_5^{\mathrm{OPT}}$, General Case

Figure 5.2. – Fourier footprint of the SDRT_4 (left) and SDRT_5 (right) spatial discretizations on triangles

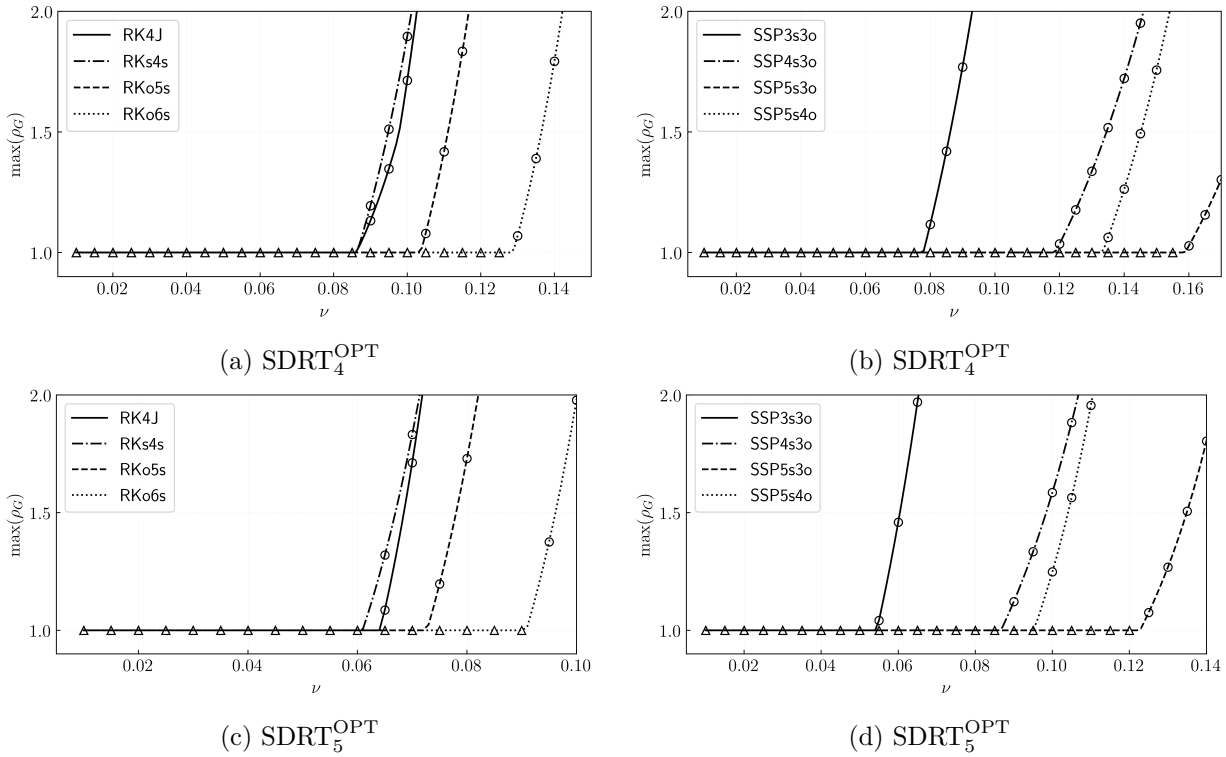
5.1.2.3. Fourier Analysis of the Coupled Time-Space Discretization

The linear stability of the coupled time-space discretization is now analyzed through the study of the spectral radius of the amplification factor $\rho_{\mathbf{G}}$ (see section 4.1.3). Since the SDRT$^{\text{OPT}}$ discretization is spatially stable, a stable behavior for the full discretization is expected for a CFL number in a range from zero to a maximum stability limit. This stability limit should also be determined for each temporal scheme.

The following parameters are considered in the study:

- $\theta \in [0, 2\pi], \Delta\theta = \pi/8$,
- $\kappa \in [-\pi, \pi], \Delta\kappa = \pi/32$,
- $\vartheta \in [0, 2\pi], \Delta\vartheta = \pi/8$,
- $\mu = (\pi/2, \pi/3, \pi/4)$.

The evolution of the maximum value of the spectral radius $\rho_{\mathbf{G}}$ as the CFL number ν increases is plotted in Fig. 5.3 for different temporal schemes. The visualization convention introduced in Chapter 4 is used here: each $\max(\rho_{\mathbf{G}})$ value is associated with a marker. If $\max(\rho_{\mathbf{G}}) \leq 1$ (respectively $\max(\rho_{\mathbf{G}}) > 1$), the associated marker is a triangle (respectively a circle). As expected due to the linear spatial stability, the evolution shows a fully stable behavior (triangular marker) up to a maximum CFL number. These stability limits are accurately given in Table 5.2.



(a) SDRT$_4^{\text{OPT}}$ (b) SDRT$_4^{\text{OPT}}$

(c) SDRT$_5^{\text{OPT}}$ (d) SDRT$_5^{\text{OPT}}$

Figure 5.3. – Stability behavior of the coupled time-space discretizations for SDRT$^{\text{OPT}}$ schemes on triangles

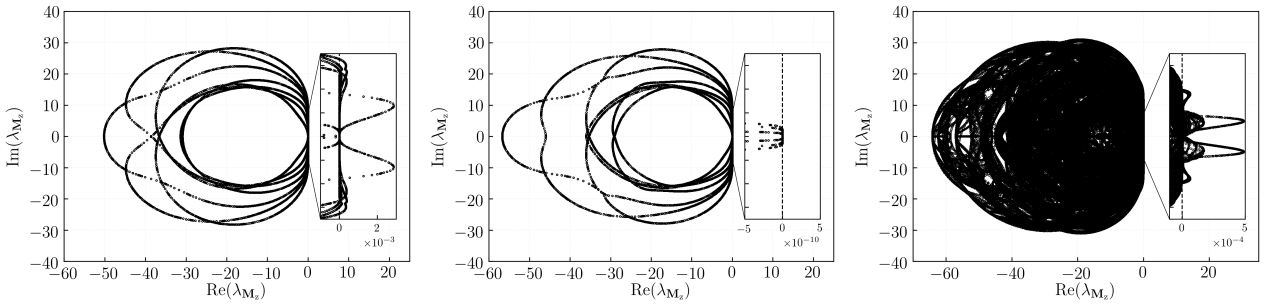
$\text{SDRT}_4^{\text{OPT}}$		$\text{SDRT}_5^{\text{OPT}}$	
Temporal Scheme	ν	Temporal Scheme	ν
RK4J	0.086	RK4J	0.064
RKs4s	0.086	RKs4s	0.060
RKo5s	0.103	RKo5s	0.072
RKo6s	0.128	RKo6s	0.090
SSP3s3o	0.077	SSP3s3o	0.054
SSP4s3o	0.118	SSP4s3o	0.086
SSP5s3o	0.158	SSP5s3o	0.122
SSP5s4o	0.133	SSP5s4o	0.095

Table 5.2. – CFL stability limits for SDRT^{OPT} schemes on triangles coupled with different temporal schemes

5.1.3. Study for the SDRT$_6$ Scheme

The linear stability analysis was conducted for the SDRT$_6^{\text{WSJ}}$ scheme in Chapter 3, section 3.3.3. The corresponding spectrum of the matrix $\mathbf{M}_z$ is here shown in Fig. 5.4a for an advection angle $\theta = 0$. The spatial discretization is unstable, with a maximum eigenvalues real part of order 10^{-3}.
The optimization process is run for the SDRT$_6$ scheme. The parametrization of the interior FP location is given in Appendix E. Given the large number of optimization parameters, the algorithm is first run for the case $\theta = 0$. A stable SDRT$_6^{\text{OPT}}$ formulation is found in this case. The corresponding $\mathbf{M}_z$ spectrum is shown in Fig. 5.4b and the associated optimization parameters are given in Table 5.3.

However, when considering the general case $\theta \in [0, 2\pi]$ (Fig. 5.4c), the interior FP determined using the optimization for $\theta = 0$ lead to an unstable SDRT formulation. The optimization process should then be run for $\theta \in [0, 2\pi]$. However, considering the large amount of optimization parameters, the computation becomes extremely costly and did not manage to find a stable SDRT$_6$ scheme. Finally, the study of the coupled time-space discretization shows that the RK4J was not able to stabilize the SDRT$_6$ scheme in a general framework ($\theta \in [0, 2\pi]$) using neither Williams-Shunn-Jameson quadrature points ($\max(\rho_{\mathbf{G}}) \sim 10^{-6}$) nor the optimization points ($\max(\rho_{\mathbf{G}}) \sim 10^{-7}$).



(a) $\theta = 0$, SDRT$_6^{\text{WSJ}}$ (b) $\theta = 0$, SDRT$_6^{\text{OPT}}$ (c) General Case, SDRT$_6^{\text{OPT}}$

Figure 5.4. – Fourier footprint of the SDRT$_6$ spatial discretizations on triangles

	SDRT$_6^{\text{OPT}}$	
α_1	0.025601729975188152604870239770	–
α_2	0.176111329503278755126416399 43	–
α_3	0.398061165124505511592190032388	–
β_1, γ_1	0.180524923708336115790729081709	0.059771932372879116313058034393
β_2, γ_2	0.393813927660607965286487797130	0.155450185809347496412868849802
β_3, γ_3	0.597070700120643449437807248614	0.100637371287577992440276375419
β_4, γ_4	0.644785985610853984439927444328	0.275192738170610518011471867794
β_5, γ_5	0.955432480804950801100972057611	0.126303970969899315157647379237
β_6, γ_6	0.820859390566581081927211016591	0.229642693685824350602331378468
β_7, γ_7	0.850697660514036169132623399491	0.384732707035698173747562123026
β_8, γ_8	0.966062410823000927706516449689	0.332382988539032642183457255669
β_9, γ_9	0.969853309157741039214783995703	0.465127005177082608611272007693

Table 5.3. – Coordinate parameters of interior FP determined using the optimization process on Fourier analysis for SDRT$_6$, $\theta = 0$

5.2. Tetrahedral Elements

5.2.1. Optimization Algorithm

The Fourier analysis for the spatial SDRT discretization on tetrahedral elements presented in section 3.4.4 is considered. The Fourier analysis leads to the definition of the matrix $\mathbf{M}_z$, which is fully defined through:

- the polynomial degree p,
- the position of the SP,
- the position of the FP,
- the advection angles $(\theta_1, \theta_2) \in [0, 2\pi]^2$,
- the grid frequency $\kappa \in [-\pi, \pi]$,
- the harmonic plane orientation angles $(\vartheta_1, \vartheta_2) \in [0, 2\pi]^2$.

The position of the interior FP is parametrized by four parameters $(\alpha, \beta, \gamma, \delta)$ in a similar definition as the Shunn-Ham quadrature rule and is given in Appendix E. Due to that simple parametrization, constraints have to be imposed on the optimization parameters in order for the interior FP to be located strictly inside the tetrahedron. To do so, the optimization process is based on the *scipy.optimize.minimize* function using the Sequential Least SQuares Programming (SLSQP) algorithm from the SciPy library [179]. The SLSQP optimizer is a sequential least squares programming algorithm that uses the Han–Powell quasi–Newton method and allows to impose constraints. As in the previous section, the function to minimize is the maximum of the real part of the eigenvalues of the matrix $\mathbf{M}_z$ and the optimization parameters are the interior FP locations.

The full algorithm is detailed by Algorithm 2 for the SDRT_3 scheme. The polynomial degree p is set to 3 and the SP location is set to the position given by the 20-points Shunn-Ham quadrature rule. The position of FP located on edges is set to the 10-points Williams-Shunn-Jameson quadrature rules to ensure that the FP located on a tetrahedral and a prismatic element match exactly. The optimization parameters are initialized using the 10-points Shunn-Ham quadrature rule. Interior FP coordinates are parametrized by a set of coefficients to ensure symmetry. Constraints are given to ensure that all interior FP are included in the tetrahedron. The optimization is then run: the optimization function based on the SLSQP method called the function MAIN, using x0 as the initial interior FP location and taking constraints into account.

The function MAIN returns the maximum of all eigenvalues real part of the matrix $\mathbf{M}_z$. The interior FP coordinates are first computed based on the optimization parameters, which allows to compute the transfer matrix $\mathbf{T}$ and the differentiation matrix $\mathbf{D}$. The matrix $\mathbf{M}_z$ is computed for the particular case $(\theta_1, \theta_2) = (0, 0)$, $(\vartheta_1, \vartheta_2) = (0, 0)$ and $\kappa \in [0, \pi], \Delta\kappa = \pi/32$. The maximum of the real part of all the eigenvalues of $\mathbf{M}_z$, denoted rm, is returned and will be minimized by the optimization algorithm.

Algorithm 2 Fourier analysis optimization algorithm on tetrahedral elements for SDRT_3

Constants

$p \leftarrow 3$

SP location: $\boldsymbol{\xi}^{SP}_{1:20} \leftarrow$ Shunn-Ham 20-points rule

Face FP location: $\boldsymbol{\xi}^{FP}_{1:10} \leftarrow$ Williams-Shunn-Jameson 10-points rule ▷ Face 1

$\boldsymbol{\xi}^{FP}_{11:20} \leftarrow$ Williams-Shunn-Jameson 10-points rule ▷ Face 2

$\boldsymbol{\xi}^{FP}_{21:30} \leftarrow$ Williams-Shunn-Jameson 10-points rule ▷ Face 3

$\boldsymbol{\xi}^{FP}_{31:40} \leftarrow$ Williams-Shunn-Jameson 10-points rule ▷ Face 4

Optimization Parameters

$\alpha = 0.7784952948213298$

$\beta = 0.0738349017262234$ ▷ Shunn-Ham 10-points rule

$\gamma = 0.4062443438840509$

$\delta = 0.0937556561159491$

x0 $= (\alpha, \beta, \gamma, \delta)$

Constraints $= \{0.25 < \alpha < 1, 0 < \beta < 0.25, 0.25 < \gamma < 0.5, 0 < \delta < 0.25,$
$(\alpha + \beta + \beta, \beta + \alpha + \beta, \beta + \beta + \alpha, \beta + \beta + \beta) < 1,$
$(\gamma + \gamma + \delta, \gamma + \delta + \gamma, \gamma + \delta + \delta, \delta + \gamma + \gamma) < 1,$
$(\delta + \gamma + \delta, \delta + \delta + \gamma) < 1\}$

Optimization Process

call scipy.optimize.minimize(MAIN, x0, Constraints, method='SLSQP')

function MAIN

$\boldsymbol{\xi}_{41} = (\alpha, \beta, \beta), \boldsymbol{\xi}_{42} = (\beta, \alpha, \beta), \boldsymbol{\xi}_{43} = (\beta, \beta, \alpha), \boldsymbol{\xi}_{44} = (\beta, \beta, \beta), \boldsymbol{\xi}_{45} = (\gamma, \gamma, \delta)$

$\boldsymbol{\xi}_{46} = (\gamma, \delta, \gamma), \boldsymbol{\xi}_{47} = (\gamma, \delta, \delta), \boldsymbol{\xi}_{48} = (\delta, \gamma, \gamma), \boldsymbol{\xi}_{49} = (\delta, \gamma, \delta), \boldsymbol{\xi}_{50} = (\delta, \delta, \gamma)$

$\boldsymbol{\xi}_{51:60} = \boldsymbol{\xi}_{41:50}$

$\boldsymbol{\xi}_{61:70} = \boldsymbol{\xi}_{51:60}$

Compute Transfer Matrix $\mathbf{T}$

Compute Differentiation Matrix $\mathbf{D}$

$(\vartheta_1, \vartheta_2) = (0, 0)$ ▷ Harmonic plane orientation angles

$(\theta_1, \theta_2) = (0, 0)$ ▷ Advection angles

Compute $\mathbf{M}^{0,0,0}, \mathbf{M}^{-1,0,0}, \mathbf{M}^{+1,0,0}, \mathbf{M}^{0,-1,0}, \mathbf{M}^{0,+1,0}, \mathbf{M}^{0,0,-1}, \mathbf{M}^{0,0,+1}$

for $\kappa \in [0, \pi], \Delta\kappa = \pi/32$ **do** ▷ Grid frequency

Compute $\mathbf{M}_\mathrm{z}$ using:

$$\begin{aligned}\mathbf{M}_\mathrm{z} = -\Big[&\mathbf{M}^{0,0,0} + \mathbf{M}^{-1,0,0}\exp(-I\kappa\cos\vartheta_1\sin\vartheta_2) + \mathbf{M}^{+1,0,0}\exp(I\kappa\cos\vartheta_1\sin\vartheta_2)\\ &+ \mathbf{M}^{0,-1,0}\exp(-I\kappa\sin\vartheta_1\sin\vartheta_2) + \mathbf{M}^{0,+1,0}\exp(I\kappa\sin\vartheta_1\sin\vartheta_2)\\ &+ \mathbf{M}^{0,0,-1}\exp(-I\kappa\cos\vartheta_2) + \mathbf{M}^{0,0,+1}\exp(I\kappa\cos\vartheta_2)\Big]\end{aligned}$$

Compute $\max(\mathrm{Re}(\lambda_{\mathbf{M}_\mathrm{z}}))$

rm $= \max(\mathrm{rm}, \max(\mathrm{Re}(\lambda_{\mathbf{M}_\mathrm{z}})))$

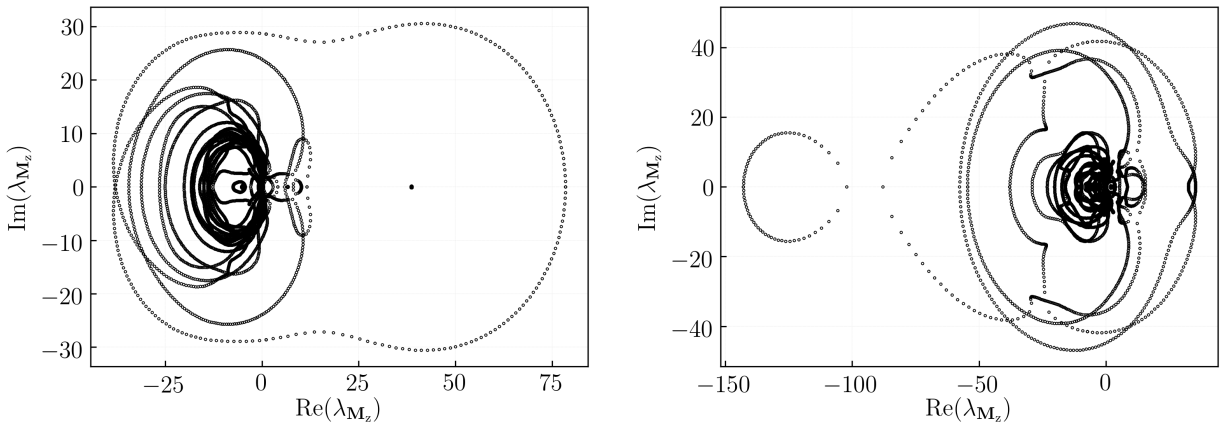
end for

return rm

end function

5.2.2. Fourier Analysis of the Spatial Discretization

The spectrum of $\mathbf{M}_z$ is plotted in Fig. 5.5a for the $SDRT_3$ using the Shunn-Ham rule as the interior FP location. As mentioned in Chapter 3, section3.4.4, $\max(\mathrm{Re}(\lambda_{\mathbf{M}_z}))$ is of order ~ 80. The optimization algorithm was able to determined a minimum for the MAIN function. The corresponding spectrum is shown in Fig. 5.5b. Even if there is a clear reduction of the maximum eigenvalue real part ($\max(\mathrm{Re}(\lambda_{\mathbf{M}_z})) \sim 35$), the spatial discretization remains unstable. The minimum found by the optimization algorithm could be a local minimum; however, changing the initial FP location (x0) led to the same value.



(a) Shunn-Ham interior FP

(b) Optimization interior FP

Figure 5.5. – Fourier footprint of the $SDRT_3$ spatial discretizations on tetrahedral elements, $(\theta_1, \theta_2, \vartheta_1, \vartheta_2) = (0, 0, 0, 0)$

5.2.3. Influence of Flux Points Located on Faces

It was mentioned in Chapter 3, section 3.4.4 that using different SP locations (either the NCO or the Shunn-Ham rule) did not influence the stability. Actually, it led to the exact same value of $\max(\mathrm{Re}(\lambda_{\mathbf{M}_z}))$. This result extends the statement that the SP location has no impact on the stability to tetrahedra. However, the influence of FP located on faces has not been studied yet. To do so, the SP location is fixed at the Shunn-Ham rule [181] whereas different locations of interior FP and FP located on faces are studied.

For $p = 3$, there are 10 FP located on each face (denoted Face FP). Several location are studied: the 10-points quadrature rules on a triangle from Williams-Shunn-Jameson (WSJ) [174] and Vioreanu-Rokhlin (VR) [173]) and the set of 10-points determined using the optimization process in the previous section given by Table 5.1, denoted OPT. For interior FP, the three quadrature rules on a tetrahedron introduced in section 3.4.4 are considered: the Shunn-Ham [181], the Vioreanu-Rokhlin (VR) [173] and the Newton-Cotes Open (NCO) [180] quadrature rules.

Table 5.4 shows values of $\max(\mathrm{Re}(\lambda_{\mathbf{M}_z}))$ for all the possible combinations of FP locations. They were obtained using $(\theta_1, \theta_2, \vartheta_1, \vartheta_2) = (0, 0, 0, 0)$ and $\kappa \in [0, \pi], \Delta\kappa = \pi/32$. From this table, the impact of the position of the FP located on faces is clearly highlighted. The impact of the Face FP is even more important than the impact of interior FP: the interval of values obtained by changing the Face FP location is larger than by changing the interior FP location.

Interior FP \ Face FP	WSJ [174]	VR [173]	OPT
Shunn-Ham [181]	78.41	156.04	70.53
VR [173]	79.17	153.82	72.59
NCO [180]	68.74	110.68	95.92

Table 5.4. – Impact of the FP located on faces: values of $\max(\mathrm{Re}(\lambda_{\mathbf{M}_z}))$ for SDRT_3, $(\theta_1, \theta_2, \vartheta_1, \vartheta_2) = (0, 0, 0, 0)$

In this chapter, an optimization process based on the Fourier analysis was conducted for triangular and tetrahedral elements.
For triangles, interior FP leading to stable SDRT schemes were determined for polynomial approximations of degree four and five. The implementation based on the interior FP located at the optimization points is called $\mathrm{SDRT}^{\mathrm{OPT}}$. The Fourier analysis of the time-space discretization verified that the full discretization remains stable under a CFL limit, which was numerically determined for several time integration schemes. The $\mathrm{SDRT}^{\mathrm{OPT}}$ implementation is stable for all advection angles up to a given CFL limit for temporal schemes associated with good dissipation and dispersion properties. For unsteady problems, it thus seems to be a more viable solution for $p = 4, 5$ than using the $\mathrm{SDRT}^{\mathrm{WSJ}}$. The two implementations will be compared in the next chapter for an unsteady test case in section 6.1. Note that for steady problems, the $\mathrm{SDRT}^{\mathrm{WSJ}}$ implementation associated with the RK4J temporal scheme remains a possibility for $p = 4, 5$. Finally, for the SDRT_6 scheme, improvements were shown by the optimization process for the particular case of $\theta = 0$ but an implementation considered stable under the Fourier analysis could not be determined at the moment.
For tetrahedral elements, the optimization process did not manage to determine stable SDRT formulation for a fourth-order of accuracy. However, the optimization problem presented in this chapter only considered the interior FP location as an optimization parameter whereas the FP located on faces were fixed at Williams-Shunn-Jameson quadrature points. As it was demonstrated, the position of FP located on faces has an important impact on the scheme stability. Therefore, an optimization problem considering both the interior FP location and the position of FP located on faces as optimization parameters could be able to determine stable SDRT formulations for $p > 2$.

Chapter 6

Validation for First-Order Partial Differential Equations

In this chapter, a set of numerical experiments are conducted for first-order PDEs. First, the 2D linear advection equation is considered to assess the accuracy of the two different SDRT implementations for an unsteady problem using polynomial approximations of degree four and five: the SDRT$^{\mathrm{WSJ}}$ stabilized by the RK4J temporal scheme proposed in Chapter 4 and the SDRT$^{\mathrm{OPT}}$ based on interior FP determined in Chapter 5. The Euler equations are then considered in two test cases. The first one is the 2D convection of an isentropic vortex (COVO) on regular triangular and hybrid grids. The order of accuracy is verified in the nonlinear case for both types of grids using $p \in [\![2, 5]\!]$. The second test case presents a convergence study for the 3D Euler equations using tetrahedral grids for $p \in [\![1, 2]\!]$.

Overview

6.1. 2D Linear Advection Equation . . . 98
6.1.1. Advection of a Sine . . . 98
6.1.2. Assessment of SDRT$^{\mathrm{WSJ}}$ - RK4J Accuracy . . . 99
6.1.3. Assessment of SDRT$^{\mathrm{OPT}}$ Accuracy . . . 100
6.2. Convection of an 2D Isentropic Vortex . . . 102
6.2.1. Triangular Grids . . . 103
6.2.2. Hybrid Grids . . . 105
6.3. 3D Euler Equations . . . 108

6.1. 2D Linear Advection Equation

6.1.1. Advection of a Sine

The 2D linear advection equation given by Eq. (3.1) is considered. The initial solution is taken as:

$$u(x, y, 0) = 1 + 0.5 \sin(\pi(x + y)), \tag{6.1}$$

and the velocity vector is $\mathbf{c} = (U, V)^\top = (\cos\theta, \sin\theta)^\top$ where $\theta = 0$. The analytical solution u_a can be computed at any time using:

$$u_a(x, y, t) = 1 + 0.5 \sin(\pi(x + y - t(U + V))). \tag{6.2}$$

The computational domain is $\Omega = [-1, 1]^2$. Computations are performed on a regular triangular mesh composed of $2 \cdot 15^2$ triangular cells shown in Fig. 6.1 and periodic boundary conditions are used in both x and y directions. At interfaces, an upwind flux is used as the numerical flux.

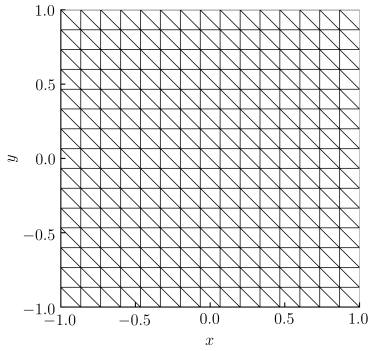


Figure 6.1. – Regular triangular grid used for the SDRT implementation comparison

To compare the accuracy of each SDRT implementation, the L_2 error is computed on the domain as:

$$L_2 = \sqrt{\frac{\int_\Omega (u_{ha} - u_{num})^2 \, d\Omega}{\int_\Omega d\Omega}}, \tag{6.3}$$

where u_{ha} is the polynomial approximation of the analytical solution u_a. In Eq. (6.3), the integral on the top can be expressed as the following sum on each cell:

$$\int_\Omega (u_{ha} - u_{num})^2 \, d\Omega = \sum_{i=1}^{N} \int_{\Omega_i} \left(u_{ha}^{(i)} - u_{num}^{(i)}\right)^2 d\Omega, \tag{6.4}$$

where N is the number of cells on the domain Ω. Integration is then performed in the reference domain using a quadrature rule such that:

$$\sum_{i=1}^{N} \int_{\Omega_i} \left(u_{ha}^{(i)} - u_{num}^{(i)}\right)^2 d\Omega = \sum_{i=1}^{N} \sum_{j=1}^{N_q} A \, |J^{(i,j)}| \, \omega_j \left(u_{ha}^{(i)}(\boldsymbol{\xi}_j) - u_{num}^{(i)}(\boldsymbol{\xi}_j)\right)^2, \tag{6.5}$$

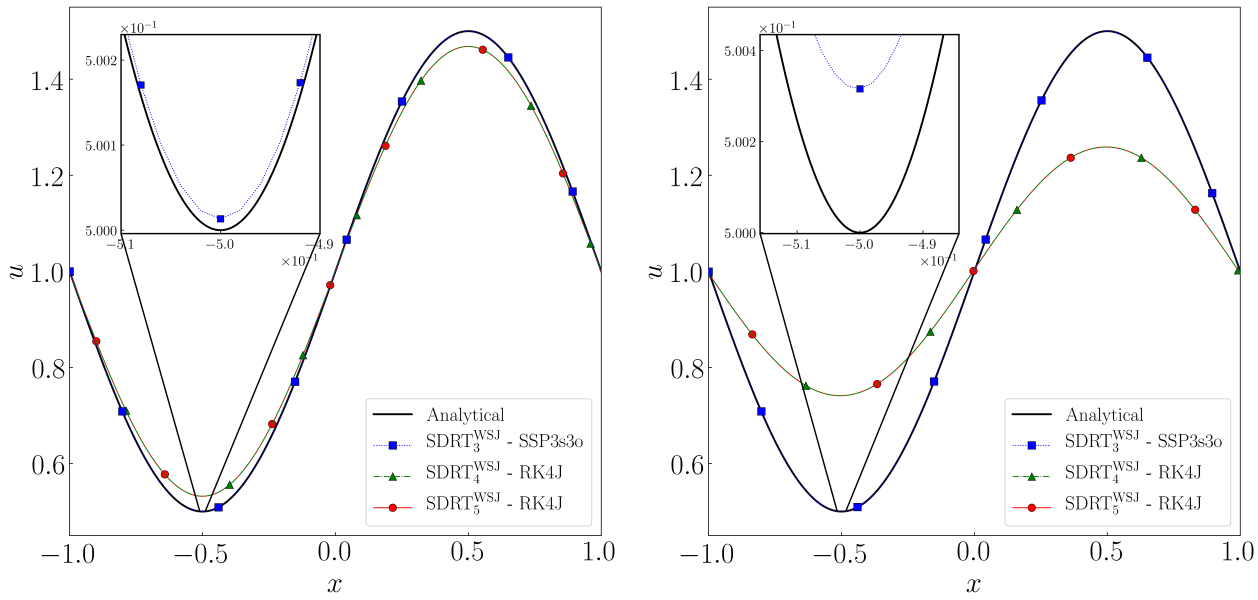
where A is the reference element area ($A = 1/2$ for a triangular element), $|J^{(i,j)}|$ is the Jacobian determinant at the j-th integration point of the i-th cell and N_q is the number of quadrature points. The quadrature points are located at $\boldsymbol{\xi}_j$ and associated with the weight ω_j. Since u_{ha} and u_{num} are polynomials of degree p, the term $\left(u_{ha}^{(i)} - u_{num}^{(i)}\right)^2$ should be approximated using a quadrature of order $2p$. On triangles, the integration is carried out using the 175-points symmetric quadrature given by Wandzura and Xiao [192], which can be used up to degree 30.

6.1.2. Assessment of SDRT^{WSJ} - RK4J Accuracy

The first implementation to be studied is the SDRT^{WSJ}. It is recalled here that the linear stability analysis of the coupled time-space discretization presented in Chapter 4 showed that spatially unstable SDRT$_4^{WSJ}$ and SDRT$_5^{WSJ}$ discretizations can be stabilized using the RK4J temporal scheme. However, it was noted that the numerical dissipation error induced by the RK4J scheme is higher than other temporal schemes.

The advection test case presented in section 6.1.1 is run using the SDRT$_4^{WSJ}$-RK4J and SDRT$_5^{WSJ}$-RK4J discretizations and compared to the solution obtained using a SDRT$_3^{WSJ}$ scheme associated with the SSP3s3o. The aim is to determined if it is better to use a high-order spatial scheme or a smaller polynomial approximation degree associated with a temporal scheme with better dissipation and dispersion properties.

Results obtained with a SDRT$_3^{WSJ}$ spatial discretization coupled with the SSP3s3o are compared with results obtain with a SDRT$_4^{WSJ}$ and SDRT$_5^{WSJ}$ spatial discretizations associated with the RK4J in Fig. 6.2. For both temporal schemes, the CFL number is $\nu = 0.05$. The computation is run for two final times: $t_f = 20\,\text{s}$ (Fig. 6.2a) and $t_f = 200\,\text{s}$ (Fig. 6.2b). In both case, the SDRT$_3^{WSJ}$-SSP3s3o lead to a very accurate solution and can barely be distinguished from the analytical solution. On the other hand, the use of SDRT$_4^{WSJ}$-RK4J or SDRT$_5^{WSJ}$-RK4J shows numerical dissipation: at $t_f = 20\,\text{s}$, 93% of the sine amplitude is preserved but at $t_f = 200\,\text{s}$, only half of the sine amplitude remains. At both time, solutions for SDRT$_4^{WSJ}$ and SDRT$_5^{WSJ}$ seems very closed: the same L_2 error is actually obtained using either $p = 4$ or $p = 5$ (Table 6.1). The numerical error is thus fully driven by the temporal scheme, and not by the spatial scheme. To reduce the error and get a better accuracy, the CFL number should be decreased, leading to a higher computational time. This comparison indicates that using the RK4J temporal scheme is not a practicable option to obtain high-order accuracy for unsteady computations at a reasonable cost.



(a) $t_f = 20\,\text{s}$ (b) $t_f = 200\,\text{s}$

Figure 6.2. – Assessment of the RK4J and SSP3s3o accuracy for the SDRT^{WSJ} implementation - Cross-section of the solution u at $y = 0$

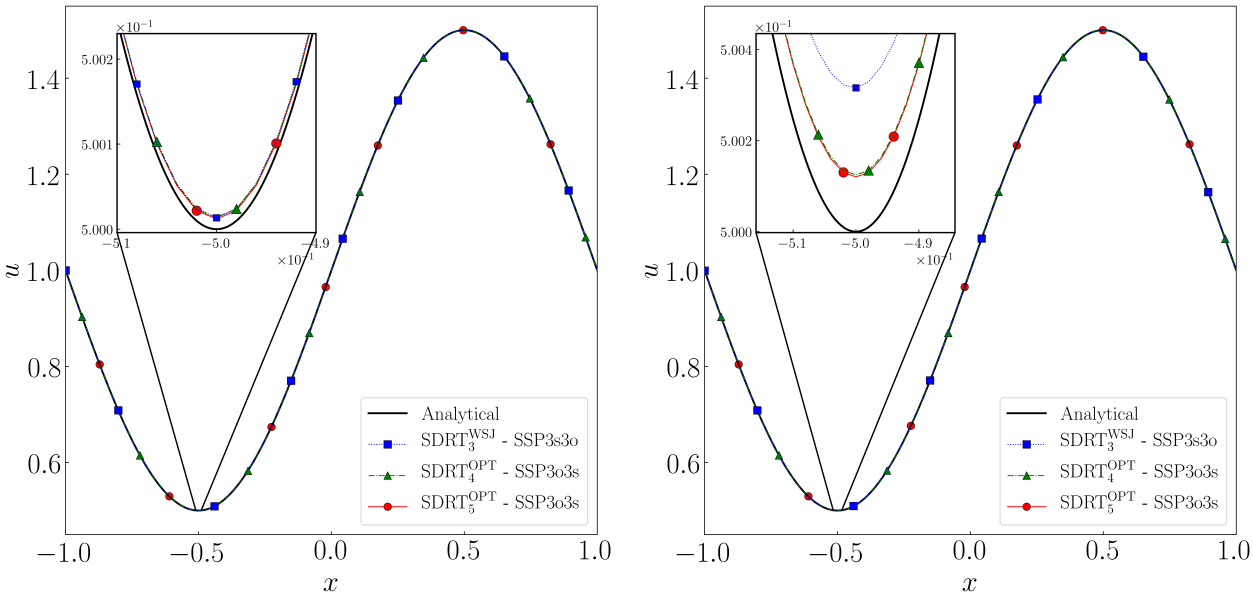
	$t_f = 20\,\mathrm{s}$	$t_f = 200\,\mathrm{s}$
$\mathrm{SDRT}_3^{\mathrm{WSJ}}$ - SSP3s3o	$5.08731 \cdot 10^{-5}$	$3.71342 \cdot 10^{-4}$
$\mathrm{SDRT}_4^{\mathrm{WSJ}}$ - RK4J	$2.25318 \cdot 10^{-2}$	$1.70558 \cdot 10^{-1}$
$\mathrm{SDRT}_5^{\mathrm{WSJ}}$ - RK4J	$2.25318 \cdot 10^{-2}$	$1.70558 \cdot 10^{-1}$

Table 6.1. – Assessment of the RK4J and SSP3s3o accuracy for the $\mathrm{SDRT}^{\mathrm{WSJ}}$ implementation - L_2 error

6.1.3. Assessment of $\mathrm{SDRT}^{\mathrm{OPT}}$ Accuracy

The same test case is now run using $\mathrm{SDRT}_4^{\mathrm{OPT}}$ and $\mathrm{SDRT}_5^{\mathrm{OPT}}$ discretizations. The aim is to verify that the small change in the interior FP location does not impact the scheme accuracy. To do so, solutions are compared with the $\mathrm{SDRT}_3^{\mathrm{WSJ}}$ scheme, whose interior FP at located at the WSJ quadrature points.

For the three SDRT schemes, solutions are time-integrated using the SSP3s3o temporal scheme and the CFL number is $\nu = 0.05$. Results are presented in Fig. 6.3. At the final time $t_f = 20\,\mathrm{s}$ (Fig. 6.3a), all of the SDRT schemes perfectly match the analytical solution. In the close view, a very small amount of dissipation can be seen. The three schemes seem to lead to the same accuracy, but the L_2 error values presented in Table 6.2 allows one to see that a better solution is obtained with $\mathrm{SDRT}_4^{\mathrm{OPT}}$ and $\mathrm{SDRT}_5^{\mathrm{OPT}}$ schemes. At $t_f = 200\,\mathrm{s}$ (Fig. 6.3b), the accuracy difference becomes more visible on the close view, where the solution obtained with $\mathrm{SDRT}_3^{\mathrm{WSJ}}$ has dissipated a little more. The L_2 error values corroborate this statement: the accuracy increases with the polynomial approximation degree. This result indicates that slightly changing the interior FP location has a negligible impact on the scheme accuracy and validates the proposed $\mathrm{SDRT}^{\mathrm{OPT}}$ implementation to obtain high-order accuracy for unsteady problems using a polynomial approximation of degree four or five.



(a) $t_f = 20\,\mathrm{s}$ (b) $t_f = 200\,\mathrm{s}$

Figure 6.3. – Assessment of the $\mathrm{SDRT}^{\mathrm{OPT}}$ implementation accuracy - Cross-section of the solution u at $y = 0$

	$t_f = 20\,\mathrm{s}$	$t_f = 200\,\mathrm{s}$
$\mathrm{SDRT}_3^{\mathrm{WSJ}}$ - SSP3s3o	$5.08731 \cdot 10^{-5}$	$3.71342 \cdot 10^{-4}$
$\mathrm{SDRT}_4^{\mathrm{OPT}}$ - SSP3s3o	$8.70316 \cdot 10^{-6}$	$8.62779 \cdot 10^{-5}$
$\mathrm{SDRT}_5^{\mathrm{OPT}}$ - SSP3s3o	$8.50374 \cdot 10^{-6}$	$8.50279 \cdot 10^{-5}$

Table 6.2. – Assessment of the $\mathrm{SDRT}^{\mathrm{OPT}}$ implementation accuracy - L_2 error

This first test case has allowed us to numerically assess the accuracy of the two proposed SDRT implementations. For unsteady problems, the $\mathrm{SDRT}^{\mathrm{WSJ}}$-RK4J is not a practicable option due to the high dissipation error induced by the temporal scheme. It however remains a possible SDRT implementation for steady flows and obviously for $p \leq 3$ associated with other temporal schemes. On the other hand, $\mathrm{SDRT}^{\mathrm{OPT}}$ schemes based on a polynomial approximation of degree $p = 4$ and $p = 5$ can be associated with temporal schemes with good dissipation properties and led to very accurate results. Their accuracy order, which should be $p + 1$, will be determined in the next section using a 2D Euler test case.

6.2. Convection of an 2D Isentropic Vortex

A nonlinear case is now studied by considering the two-dimensional Euler equations:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = 0, \quad \text{in } \Omega \times [0, t_f], \tag{6.6}$$

where $\mathbf{u}$, $\mathbf{f}$ and $\mathbf{g}$ are given by:

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho U \\ \rho V \\ E \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho U \\ \rho U^2 + P \\ \rho UV \\ U(E+P) \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \rho V \\ \rho VU \\ \rho V^2 + P \\ V(E+P) \end{pmatrix}. \tag{6.7}$$

In Eq. (6.7), ρ is the density, U (respectively V) is the velocity component in the x (respectively y) direction, E is the total energy and P is the pressure determined from the following equation of state:

$$P = (\gamma - 1)\left(E - \frac{1}{2}\rho(U^2 + V^2)\right), \tag{6.8}$$

where the constant ratio of specific heats γ is equal to 1.4 for air.

To assess the SDRT scheme accuracy and capability to preserve vorticity in an unsteady inviscid flow, the convection of an isentropic vortex (COVO) test case from the International Workshop on High-Order CFD Methods [193] is studied. An isentropic vortex is transported by an inviscid uniform flow defined by $P_\infty = 10^5$ Pa, $T_\infty = 300$ K, $M_\infty = U_\infty/\sqrt{\gamma R_{gas} T_\infty} = 0.5$. The fluid is assumed to be a perfect gas, with a constant ratio of specific heats $\gamma = 1.4$. An isentropic vortex of characteristic radius $R = 0.005$ m and strength $\beta = 0.2$ is added to this mean flow around the point of coordinates $(X_c, Y_c) = (0.05, 0.05)$ through perturbation in U, V and the temperature T. The computation is thus initialized by the local velocity components U_0 and V_0 as well as temperature T_0:

$$U_0 = U_\infty \left(1 - \beta \, \frac{(y - Y_c)}{R} \, \exp\left(-r^2/2\right)\right), \tag{6.9}$$

$$V_0 = U_\infty \beta \, \frac{(x - X_c)}{R} \, \exp\left(-r^2/2\right), \tag{6.10}$$

$$T_0 = T_\infty - \frac{U_\infty^2 \beta^2}{2R_{gas}} \frac{(\gamma - 1)}{\gamma} \exp\left(-r^2\right), \tag{6.11}$$

where

$$r = \frac{\sqrt{(x - X_c)^2 + (y - Y_c)^2}}{R}. \tag{6.12}$$

Since the vortex is isentropic, the density can be computed using:

$$\rho_0 = \rho_\infty \left(\frac{T_0}{T_\infty}\right)^{\frac{1}{\gamma - 1}}, \tag{6.13}$$

where $\rho_\infty = \dfrac{P_\infty}{R_{gas} T_\infty}$. Euler equations are solved on the computational domain $\Omega = [0, L]^2$ where $L = 0.1$ m. Translational periodic boundary conditions are imposed for the left/right and top/bottom boundaries respectively. The SSP3s3o scheme introduced for the coupled time-space analysis is considered for the simulations. The time step is chosen sufficiently small so that the error from the time discretization is negligible compared to the spatial discretization error by setting the CFL number to

10^{-2}. At interfaces, Roe's Riemann solver [194] was used to compute the numerical flux. The SDRT implementation is as follow: the SDRT$^{\mathrm{WSJ}}$ is used for $p = 2$ and $p = 3$ whereas the SDRT$^{\mathrm{OPT}}$ is used for $p = 4$ and $p = 5$. Two different types of grids are considered: triangular and hybrid (quadrilaterals and triangles). The numerical scheme used for quadrilateral cells is the standard SD method based on the interior FP located at the zeros of the corresponding Legendre polynomials.

6.2.1. Triangular Grids

6.2.1.1. Comparison at 50 Periods

To assess the accuracy difference between the different SDRT$_p$ schemes, the COVO test case is first run on triangular grids until 50 periods using approximately 18,000 DOF. Computations are performed on different mesh refinements so that the number of DoF between orders is as equivalent as possible. Considering a regular mesh composed of $2N^2$ triangles, chosen grids are $N = (40, 30, 25, 20)$ for $p = (2, 3, 4, 5)$, leading to a total number of DOF of $(19200, 18000, 18750, 16800)$. Figure 6.4 gives the initial solution by showing ρV contours (product of the density ρ and the y-velocity V) on the mesh used for the SDRT$_5$ scheme ($N = 20$).

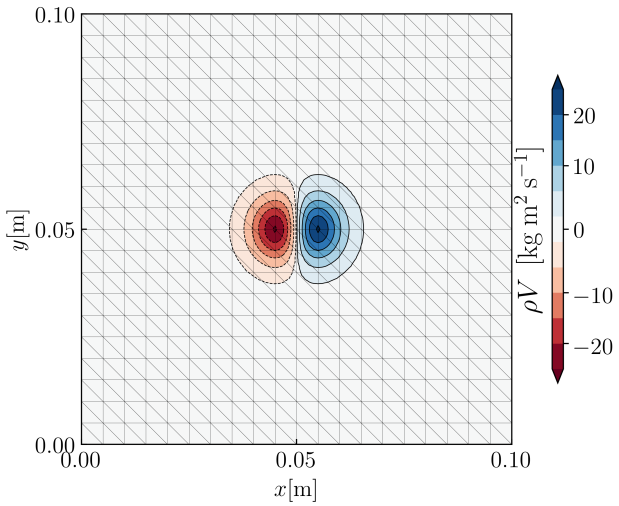


Figure 6.4. – Initialization of the COVO test case using SDRT$_5$ and $N = 20$ on a triangular grid

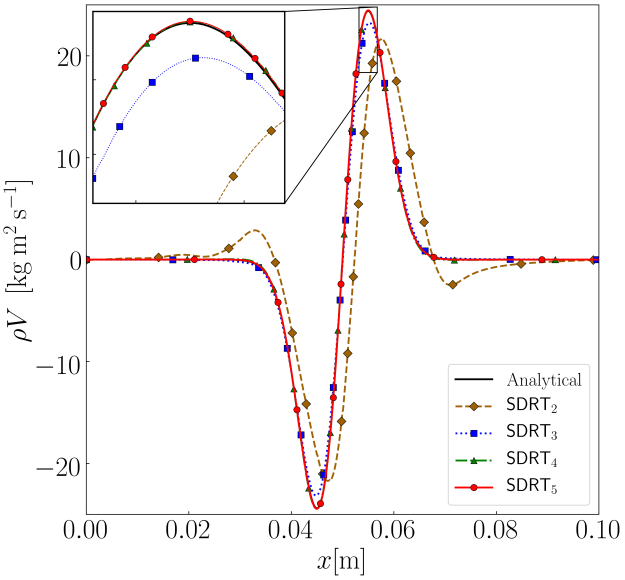


Figure 6.5. – Cross-section of ρV at $y = 0.05$ m after 50 periods using SDRT$_p$ schemes, $p \in [\![2, 5]\!]$, on triangular grids for $\sim 18,000$ DOF

Results obtained with each SDRT scheme after 50 periods are compared in Fig. 6.5 by plotting the product ρV compared to the analytical solution on a cross-section at $y = 0.05$ m. The visualization process on triangular grids is done by extrapolating the solution at SP to high-order Lagrange elements nodes. Results obtained with SDRT$_2$ shows dissipation and dispersion whereas the SDRT$_3$ scheme only leads to a small amount of dissipation and no dispersion. Both SDRT$_4$ and SDRT$_5$ lead to a very accurate solution; a close view shows that both of them exactly match the analytical solution.

In Fig. 6.6, ρV contours obtained with each SDRT scheme are displayed, where the accuracy improvement when the SDRT scheme order increases is highlighted.

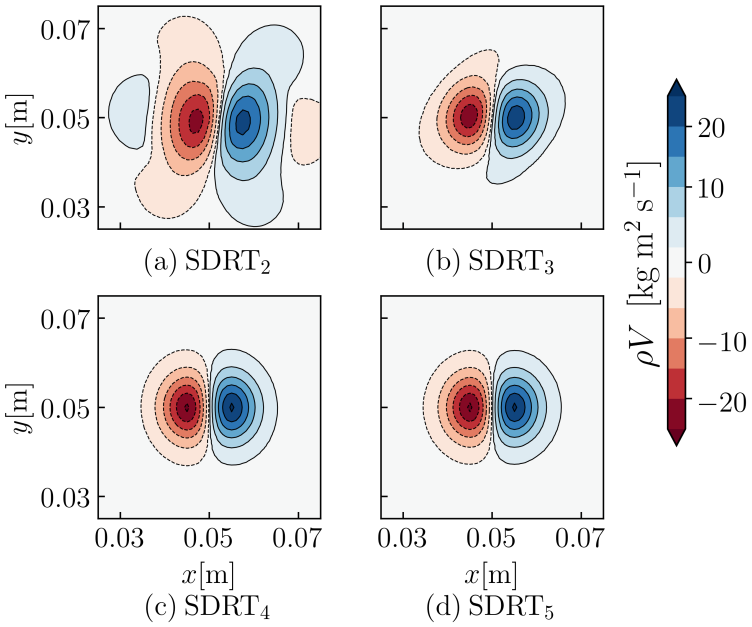


Figure 6.6. – ρV contours after 50 periods using SDRT$_p$ schemes, $p \in [\![2, 5]\!]$, on triangular grids for $\sim 18,000$ DOF

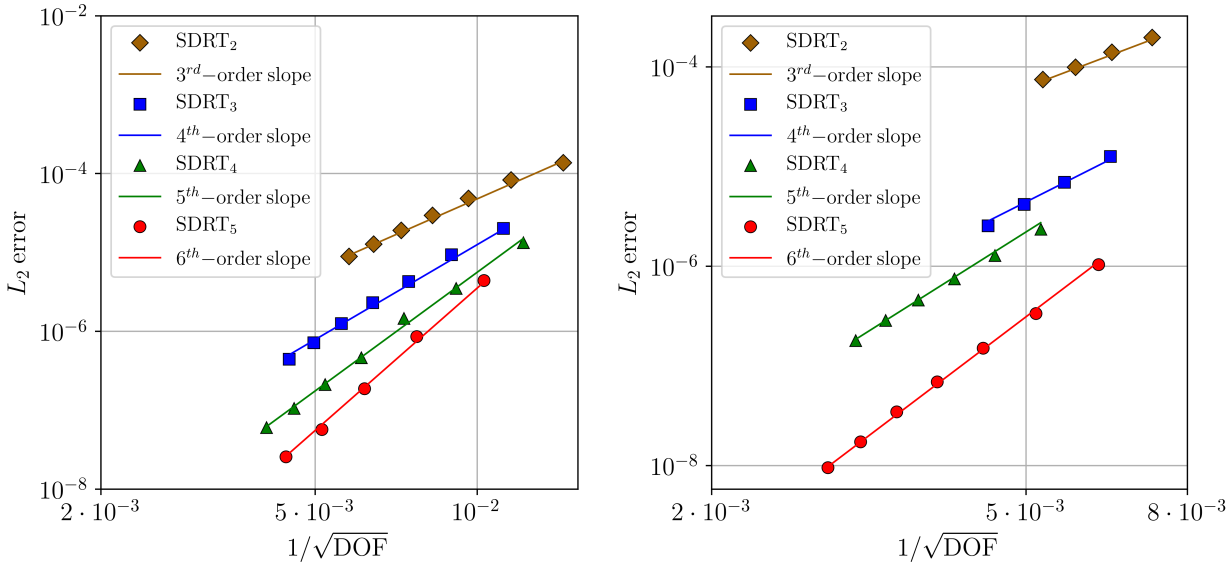
6.2.1.2. Convergence Study

A mesh refinement study is then performed using regular triangular grids of different size. To verify the order of accuracy of the SDRT scheme, the L_2 error on the density can be computed at each period on the domain as:

$$L_2 = \sqrt{\frac{\int_\Omega \left(\rho_{h_0} - \rho_{num}\right)^2 d\Omega}{\int_\Omega d\Omega}}, \tag{6.14}$$

where ρ_{h_0} is the polynomial approximation of the initial solution ρ_0. Integration is performed as detailed by Eq. (6.5) in section 6.1.1 using the 175-points symmetric quadrature given by Wandzura and Xiao [192].

The L_2 norm of the density error is plotted in Fig. 6.7 after 5 and 50 time-periods. For each SDRT scheme, the L_2 error is compared with the expected order slope. For both final times, the expected order of accuracy $p + 1$ is retrieved. The precise overall order of accuracy, computed using a least-squares polynomial fit of degree one, is given in Table 6.3)



(a) After 5 periods

(b) After 50 periods

Figure 6.7. – L_2 error and theoretical order of accuracy slopes for the COVO test case on triangular grids

Scheme	Order of accuracy
SDRT$_2$	3.03
SDRT$_3$	4.02
SDRT$_4$	4.98
SDRT$_5$	6.19

(a) After 5 periods

Scheme	Order of accuracy
SDRT$_2$	3.08
SDRT$_3$	4.46
SDRT$_4$	4.95
SDRT$_5$	6.11

(b) After 50 periods

Table 6.3. – Overall accuracy orders for the COVO test case on triangular grids

6.2.2. Hybrid Grids

6.2.2.1. Comparison at 5 Periods

The COVO test case is now run using 2D regular hybrid grids. The left part of the computational domain ($x \in [0, 0.05]$) is meshed using quadrilateral elements whereas the right part ($x \in [0.05, 0.1]$) is meshed with triangles. The mesh is composed of $\frac{1}{2}N^2$ quadrilaterals and N^2 triangles. An example is given in Fig. 6.8 for $N = 16$ where the initial solution obtained using a SD/SDRT$_5$ scheme is also displayed. Computations are performed on different grids so that the number of DoF between orders is approximately $10,000$. The very same polynomial degree is used for both triangular and quadrilateral elements. For $p = (2, 3, 4, 5)$, $N = (32, 24, 18, 16)$ is used, leading to a total number of DOF of respectively $(10752, 10368, 9360, 9984)$. Note that the mesh used for $p = 4$ should lead to 8910 DOF but for some reason, the mesh generator built a grid composed of 180 quadrilateral cells and 324 triangles.

Results obtained with each SD/SDRT$_p$ schemes after 5 periods are compared in Fig. 6.9 by plotting the product ρV compared to the analytical solution on a cross-section at $y = 0.05\,\text{m}$. The first thing to note is that the visualization process is here done using a Delaunay triangulation from the solution at SP. The resulting curve is then less smooth than for full triangular grids, where high-order Lagrange

elements were used. Apart from that, using hybrid grids leads to the same results as triangular grids: the accuracy increases with the scheme polynomial degree, as shown on the close view.

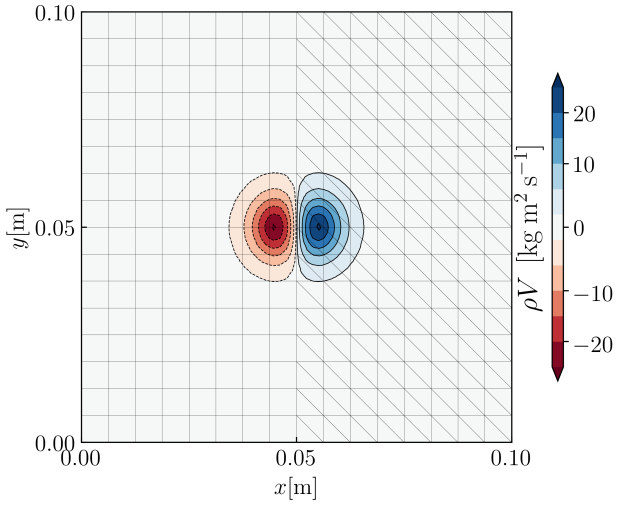


Figure 6.8. – Initialization of the COVO test case using SD/SDRT$_5$ and $N = 16$ on a hybrid grid

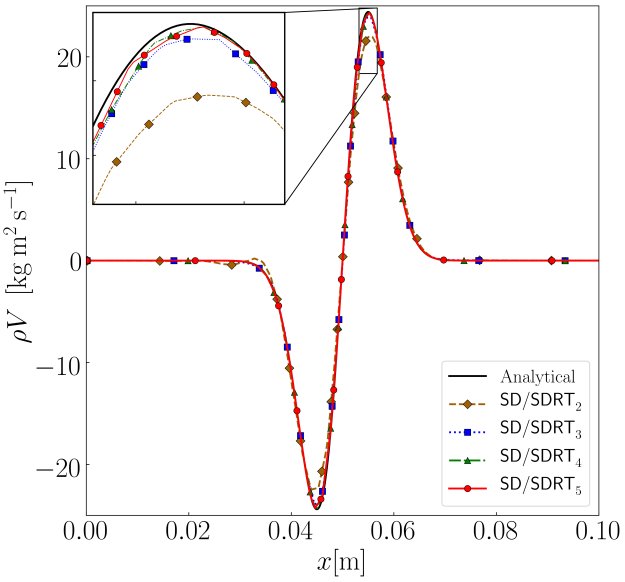


Figure 6.9. – Cross-section of ρV at $y = 0.05$ m after 5 periods using SD/SDRT$_p$ schemes, $p \in [\![2, 5]\!]$, on hybrid grids for $\sim 10,000$ DOF

Contours of ρV obtained with each SD/SDRT scheme are displayed in Fig. 6.10. The solution given by using $p = 2$ shows numerical dispersion and dissipation. From $p = 3$, an acceptable solution is obtained, even if negative and positive maximum values are only captured by $p = 4$ and $p = 5$.

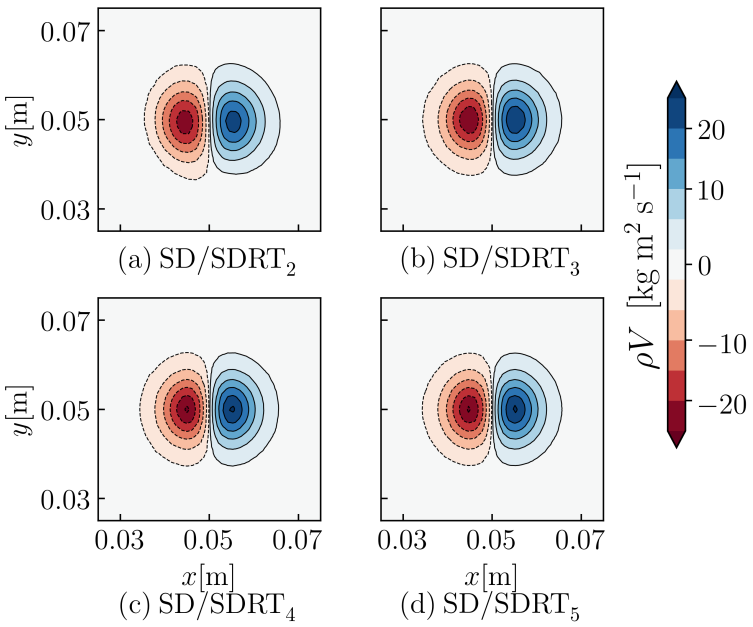


Figure 6.10. – ρV contours after 5 periods using SD/SDRT$_p$ schemes, $p \in [\![2, 5]\!]$, on hybrid grids for $\sim 10,000$ DOF

6.2.2.2. Convergence Study

As for triangular grids, a convergence study is now performed using hybrid grids of different size. The L_2 error is computed using Eq. (6.14) and Eq. (6.5) with $A = 1$ and $A = 1/2$ for a quadrilateral and a triangular element, respectively. As in the previous section, the integration on triangles is carried out using the 175-points symmetric quadrature given by Wandzura and Xiao [192]. On quadrilaterals, the integration is performed using the tensor product of two 1D integration at SP, with the appropriate Gauss-Chebyshev weights.

The L_2 norm of the density error is plotted in Fig. 6.11 after 5 time-periods and compared with the expected order slope. For all schemes, the expected order of accuracy $p + 1$ is perfectly retrieved, as shown precisely in Table 6.4 where the overall order of accuracy computed using a least-squares polynomial fit of degree one is given.

Scheme	Order of accuracy
SD/SDRT$_2$	3.00
SD/SDRT$_3$	4.02
SD/SDRT$_4$	5.05
SD/SDRT$_5$	6.09

Table 6.4. – Overall accuracy orders for the COVO test case on hybrid grids after 5 periods

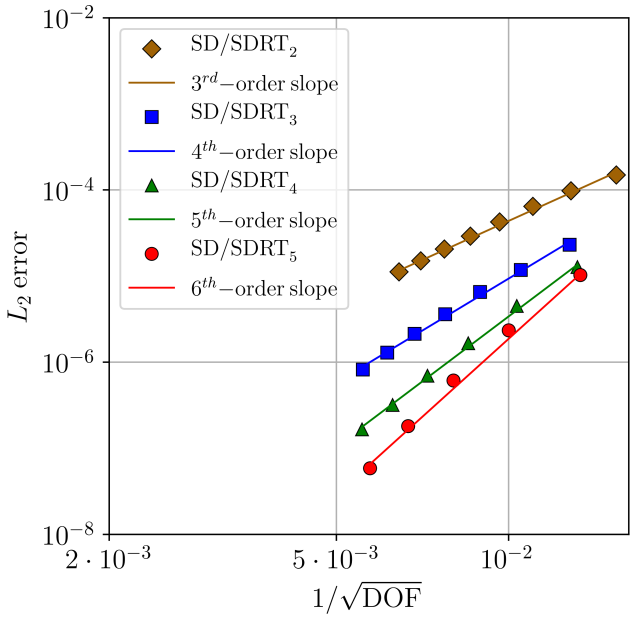


Figure 6.11. – L_2 error and theoretical order of accuracy slopes for the COVO test case on hybrid grids after 5 periods

6.3. 3D Euler Equations

Let us consider the three-dimensional Euler equations:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} + \frac{\partial \mathbf{h}}{\partial z} = 0, \quad \text{in } \Omega \times [0, t_f], \tag{6.15}$$

where $\mathbf{u}$, $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{h}$ are given by:

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho U \\ \rho V \\ \rho W \\ E \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho U \\ \rho U^2 + P \\ \rho UV \\ \rho UW \\ U(E+P) \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \rho V \\ \rho VU \\ \rho V^2 + P \\ \rho VW \\ V(E+P) \end{pmatrix}, \quad \mathbf{h} = \begin{pmatrix} \rho W \\ \rho WU \\ \rho WV \\ \rho W^2 + P \\ W(E+P) \end{pmatrix}. \tag{6.16}$$

In Eq. (6.16), ρ is the density, U (respectively V, W) is the velocity component in the x (respectively y, z) direction, E is the total energy and P is the pressure determined from the following equation of state:

$$P = (\gamma - 1)\left(E - \frac{1}{2}\rho(u^2 + v^2 + w^2)\right), \tag{6.17}$$

where the constant ratio of specific heats γ is equal to 1.4 for air.

To validate the SDRT scheme implementation on tetrahedral elements, a 3D periodic Euler test case from [195, 196] is considered. The 3D Euler equations are solved on a cubic computational domain $\Omega = [0, L]^3$ where $L = 2$ m with periodic boundary conditions. The initial solution is:

$$\rho = 1 + 0.2\sin(\pi(x + y + z)), \quad (U, V, W, P) = (1, 1, 1, 1). \tag{6.18}$$

The density initial condition is displayed on Fig. 6.12 for a regular $10 \times 10 \times 10$ tetrahedral grid using $p = 2$.

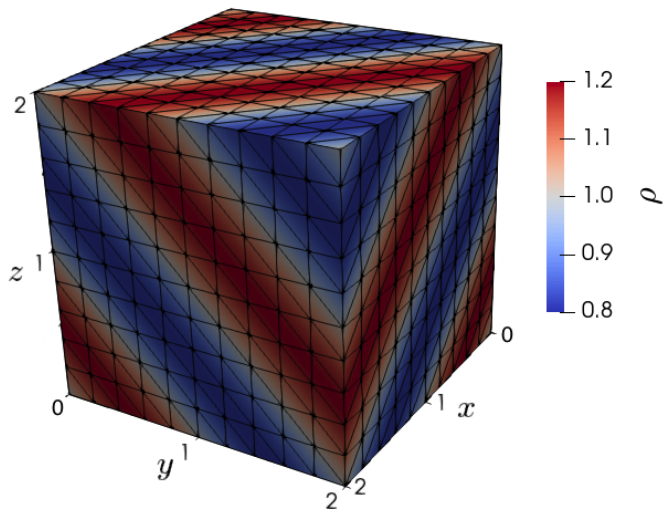


Figure 6.12. – Initial density condition on a $10 \times 10 \times 10$ regular tetrahedral grid using $p = 2$

The solution is advanced in time using the RKo6s temporal scheme and the simulation is carried out until $t_f = 1\,\mathrm{s}$. The time step is chosen sufficiently small so that the error from the time discretization is negligible compared to the spatial discretization error by setting the CFL number to 10^{-2}. The analytical solution of the density at any time t [196] is:

$$\rho_a = 1 + 0.2\sin(\pi(x + y + z - t(U + V + W))), \tag{6.19}$$

whereas velocity and pressure remain constant in time.

The convergence study was conducted on two types of grids (regular and irregular). Each mesh is refined several times and the order of accuracy is verified by computing the density L_2 error using the 84 points quadrature rule from [174]. Table 6.5 shows the L_2 errors and orders of accuracy for the two different types of grids. For both second-order and third-order schemes, a $p+1$ order of accuracy is recovered.

	Regular mesh			Irregular mesh		
p	DoF number	L_2 error	Order of accuracy	DoF number	L_2 error	Order of accuracy
1	3000	2.183E-02	-	2132	7.713E-02	-
	24000	5.754E-03	1.92	11792	2.988E-02	1.66
	81000	2.582E-03	1.98	36460	1.437E-02	1.95
	192000	1.458E-03	1.99	82164	8.349E-03	2.00
	375000	9.345E-04	1.99	151448	5.448E-03	2.09
2	7500	1.366E-03	-	5330	2.156E-02	-
	60000	1.508E-04	3.18	29480	3.277E-03	3.30
	202500	4.343E-05	3.07	91150	9.946E-04	3.17
	480000	1.813E-05	3.04	205410	4.148E-04	3.23
	937500	9.233E-06	3.02	378620	2.173E-04	3.17

Table 6.5. – L_2 error and order of accuracy values for regular and irregular tetrahedral grids

Test cases presented in this chapter allow us to validate the implementation of the SDRT scheme for first-order PDEs on triangular, tetrahedral and 2D hybrid grids. The accuracy of SDRT$^{\mathrm{OPT}}$ schemes based on a polynomial approximation of degree $p = 4$ and $p = 5$ on triangular and 2D hybrid grids was shown as equal to $p+1$, as expected. On tetrahedral grids, the SDRT implementation is validated up to the third-order of accuracy.

Chapter 7

Extension and validation for Second-Order Partial Differential Equations

Numerical experiments are now conducted on second-order PDEs, i.e. Navier-Stokes equations. The procedure to deal with viscous terms using the SDRT approach is first presented. A viscous flow over a NACA0012 airfoil using quadratic triangular elements is then studied considering three different viscous flow conditions: symmetric subsonic, asymmetric subsonic and transonic flow. A 2D hybrid grid is then used to solve a laminar flow around a cylinder at $Re = 20$. Finally, the Taylor-Green Vortex is used to validate the SDRT implementation for viscous flows on tetrahedral grids.

Overview

7.1. Extension of the Spectral Difference Approach for Navier-Stokes Equations on Triangular and Tetrahedral Elements . . . 112
7.2. 2D Viscous Flow Over a NACA0012 Airfoil . . . 114
7.2.1. Case A: Symmetric Subsonic Flow . . . 115
7.2.2. Case B: Asymmetric Subsonic Flow . . . 118
7.2.3. Case C: Transonic Flow . . . 120
7.3. 2D Viscous Flow Around a Circular Cylinder . . . 122
7.4. 3D Taylor-Green Vortex . . . 125

7.1. Extension of the Spectral Difference Approach for Navier-Stokes Equations on Triangular and Tetrahedral Elements

Let us consider a 2D scalar conservation law in the reference domain:

$$\frac{\partial \hat{u}(\boldsymbol{\xi}, t)}{\partial t} + \hat{\nabla} \cdot \hat{\mathbf{f}} = 0, \tag{7.1}$$

except that now, the flux is defined by :

$$\hat{\mathbf{f}} = |J| J^{-1} \mathbf{f}(u, \nabla u), \tag{7.2}$$

leading to a second-order PDE. For the Navier-Stokes equations, the flux can be expressed as:

$$\mathbf{f} = \mathbf{f}^i(u) - \mathbf{f}^v(u, \nabla u), \tag{7.3}$$

where $\mathbf{f}^i$ is the inviscid flux and $\mathbf{f}^v$ is the viscous flux. The viscous flux depends not only on the solution u but also on its first spatial derivative ∇u. Eq. (7.1) is solved following the very same procedure as for a first-order PDE except for the determination of the flux values at FP $\hat{f}_k$. The scalar flux values are now given by:

$$\hat{f}_k = \hat{f}_k^i - \hat{f}_k^v. \tag{7.4}$$

The inviscid flux values $\hat{f}_k^i$ are computed using Eq. (2.99) since the inviscid flux only depends on the solution:

$$\hat{f}_k^i = \begin{cases} \hat{\mathbf{f}^i} \cdot \hat{\mathbf{n}}_k = |J| J^{-1} \mathbf{f}_k^i(u_h(\boldsymbol{\xi}_k)) \cdot \hat{\mathbf{n}}_k, & \boldsymbol{\xi}_k \in \mathcal{T} \setminus \partial\mathcal{T} \quad (\text{or} \in \mathcal{T}_e \setminus \partial\mathcal{T}_e), \\ \left(\hat{\mathbf{f}}_k^i \cdot \hat{\mathbf{n}}_k\right)^* = \left(\mathbf{f}_k^i \cdot |J| (J^{-1})^\top \hat{\mathbf{n}}_k\right)^*, & \boldsymbol{\xi}_k \in \partial\mathcal{T} \quad (\text{or} \in \partial\mathcal{T}_e). \end{cases} \tag{7.5}$$

To compute $\hat{f}_k^v$, which relies on the solution and its gradient, the following procedure, based on a centered formulation [110] is used. From the approximated solution in the reference domain, the physical approximated solution $u_h(\boldsymbol{\xi}_k)$ is first computed at FP:

$$u_h(\boldsymbol{\xi}_k) = \frac{1}{|J|} \hat{u}_h(\boldsymbol{\xi}_k) = \frac{1}{|J|} \mathbf{T}_{kj} \, \hat{u}_j. \tag{7.6}$$

From those values, a polynomial interpolation of degree $p+1$ should be reconstructed for the solution but this polynomial would be discontinuous at cell interfaces. For this reason, a centered scheme is used to uniquely define the solution at each flux point by averaging the values from the left and the right cells, leading to a continuous polynomial interpolation u_h^c:

$$u_h^c(\boldsymbol{\xi}_k) = \begin{cases} u_h(\boldsymbol{\xi}_k), & \boldsymbol{\xi}_k \in \mathcal{T} \setminus \partial\mathcal{T} \quad (\text{or} \in \mathcal{T}_e \setminus \partial\mathcal{T}_e), \\ \frac{1}{2}\left(u_h^L(\boldsymbol{\xi}_k) + u_h^R(\boldsymbol{\xi}_k)\right), & \boldsymbol{\xi}_k \in \partial\mathcal{T} \quad (\text{or} \in \partial\mathcal{T}_e). \end{cases} \tag{7.7}$$

The solution gradient in the reference domain is given as [144]:

$$\left(\hat{\nabla}\hat{u}\right)(\boldsymbol{\xi}_j) = \mathbf{D}_{jk} \, \hat{\mathbf{n}}_k \, \hat{u}_h^c(\boldsymbol{\xi}_k). \tag{7.8}$$

In the physical domain, the solution gradient can be expressed as:

$$\nabla u = \frac{1}{|J|} \left(J^{-1}\right)^\top \hat{\nabla}\hat{u}, \tag{7.9}$$

and since

$$\hat{u}_h^c = |J| u_h^c, \tag{7.10}$$

one gets the expression of the solution gradient (∇u) in the physical domain:

$$(\nabla u)(\boldsymbol{\xi}_j) = \frac{1}{|J|} \mathbf{D}_{jk} \left(u_h^c(\boldsymbol{\xi}_k) \left(|J| J^{-1} \right)^\top \hat{\mathbf{n}}_k \right). \tag{7.11}$$

From the solution gradient at SP in the reference domain, the solution gradient in the physical domain can be interpolated at FP:

$$(\nabla u)_h(\boldsymbol{\xi}_k) = \mathbf{T}_{kj} \ (\nabla u)(\boldsymbol{\xi}_j). \tag{7.12}$$

The polynomial approximation of the solution gradient $(\nabla u)_h$ is discontinuous at cell interfaces. As it was done for the solution, a centered scheme is used to defined a single value at cell interface:

$$(\nabla u)_h^c(\boldsymbol{\xi}_k) = \begin{cases} (\nabla u)_h(\boldsymbol{\xi}_k), & \boldsymbol{\xi}_k \in \mathcal{T} \setminus \partial\mathcal{T} \quad (\text{or} \in \mathcal{T}_e \setminus \partial\mathcal{T}_e), \\ \dfrac{1}{2}\left((\nabla u)_h^L(\boldsymbol{\xi}_k) + (\nabla u)_h^R(\boldsymbol{\xi}_k) \right), & \boldsymbol{\xi}_k \in \partial\mathcal{T} \quad (\text{or} \in \partial\mathcal{T}_e). \end{cases} \tag{7.13}$$

The continuous solution u_h^c and the continuous solution gradient $(\nabla u)_h^c$ in the physical domain are used to compute the viscous flux values:

$$\mathbf{f}_k^v = \mathbf{f}^v(u_h^c(\boldsymbol{\xi}_k), (\nabla u)_h^c(\boldsymbol{\xi}_k)). \tag{7.14}$$

The viscous flux values in the reference domain are finally given as:

$$\hat{f}_k^v = \begin{cases} |J| J^{-1} \mathbf{f}_k^v \cdot \hat{\mathbf{n}}_k, & \boldsymbol{\xi}_k \in \mathcal{T} \setminus \partial\mathcal{T} \quad (\text{or} \in \mathcal{T}_e \setminus \partial\mathcal{T}_e), \\ \mathbf{f}_k^v \cdot |J| (J^{-1})^\top \hat{\mathbf{n}}_k, & \boldsymbol{\xi}_k \in \partial\mathcal{T} \quad (\text{or} \in \partial\mathcal{T}_e). \end{cases} \tag{7.15}$$

To summarize, the flux values read:

$$\hat{f}_k = \begin{cases} |J| J^{-1} \left(\mathbf{f}_k^i - \mathbf{f}_k^v \right) \cdot \hat{\mathbf{n}}_k, & \boldsymbol{\xi}_k \in \mathcal{T} \setminus \partial\mathcal{T} \quad (\text{or} \in \mathcal{T}_e \setminus \partial\mathcal{T}_e), \\ \left(\mathbf{f}_k^i \cdot |J| (J^{-1})^\top \hat{\mathbf{n}}_k \right)^* - \mathbf{f}_k^v \cdot |J| (J^{-1})^\top \hat{\mathbf{n}}_k, & \boldsymbol{\xi}_k \in \partial\mathcal{T} \quad (\text{or} \in \partial\mathcal{T}_e). \end{cases} \tag{7.16}$$

The flux polynomial based on the flux values $\hat{f}_k = \hat{f}_k^i - \hat{f}_k^v$ is then differentiated by multiplying it by the differentiation matrix $\mathbf{D}_{jk}$ and the semi-discrete equation is integrated in time.

7.2. 2D Viscous Flow Over a NACA0012 Airfoil

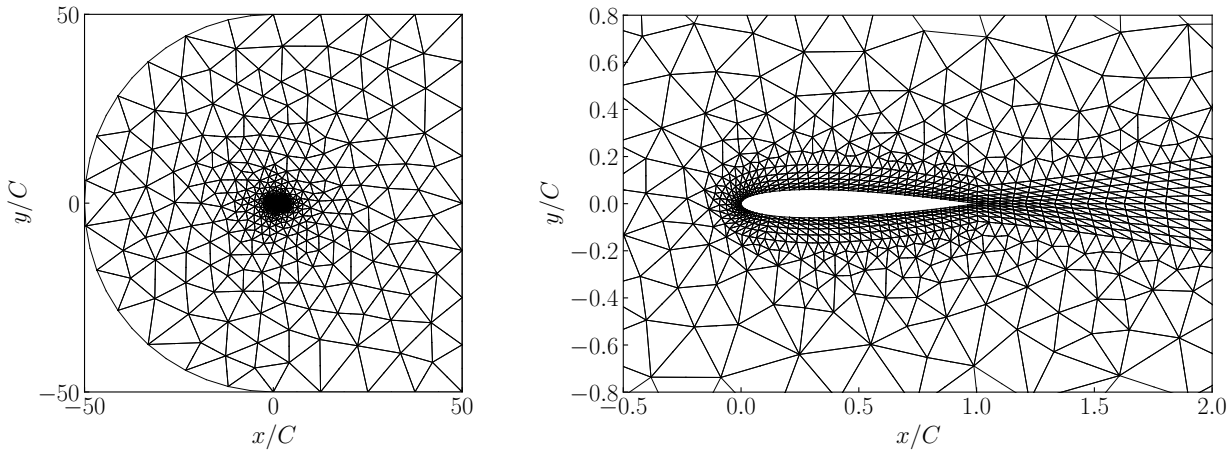
This test case aims to validate the method for the computation of viscous flow with a high-order triangular curved boundary representation. The compressible Navier-Stokes equations are solved and a laminar viscous flow over the NACA0012 airfoil is considered. The computational setup is defined by the angle of attack α, the Mach number M_∞ and the Reynolds number $\mathrm{Re} = \rho_\infty U_\infty C/\mu_{d\infty}$, where C is the airfoil chord. Three different laminar flow conditions chosen from the NASA technical report [197] are considered:

- Case A: Symmetric subsonic flow, $\mathrm{M}_\infty = 0.5$, $\alpha = 0°$, $\mathrm{Re} = 5000$,
- Case B: Asymmetric subsonic flow, $\mathrm{M}_\infty = 0.5$, $\alpha = 2°$, $\mathrm{Re} = 5000$,
- Case C: Transonic flow, $\mathrm{M}_\infty = 0.8$, $\alpha = 10°$, $\mathrm{Re} = 500$.

The NACA0012 airfoil equation used is:

$$y = \pm 0.6 \left(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4\right), \tag{7.17}$$

so the trailing edge has a zero thickness. At the airfoil, a no-slip adiabatic wall condition is imposed. To avoid spurious reflections on the boundary conditions, the farfield boundary is located 50 chords away from the airfoil. On the farfield boundary, pressure and temperature are imposed at $P_\infty = 101325\,\mathrm{Pa}$ and $T_\infty = 293.15\,\mathrm{K}$ and the velocity is imposed depending on the Mach number. Interface flux is then obtained by applying the approximated Riemann solver at the interface using the prescribed state outside and the extrapolated internal state. The computational domain is meshed with a C-type topology and has a total number of 2407 quadratic triangular elements (with 62 cells on the airfoil). The resulting mesh is shown in Fig. 7.1.



(a) Farfield view

(b) Close view

Figure 7.1. – Unstructured mesh around a NACA0012 airfoil - 2407 quadratic triangular elements

The SDRT_2 and SDRT_3 implementation is the $\mathrm{SDRT}^{\mathrm{WSJ}}$ whereas the $\mathrm{SDRT}^{\mathrm{OPT}}$ is used for SDRT_4 and SDRT_5. Solutions are time-integrated using the SSP3s3o temporal scheme and as before the convection flux is Roe's scheme. Results are presented for SDRT schemes from the third- to the sixth-order ($p \in [\![2, 5]\!]$). The CFL number is chosen based on the maximum one affordable using $p = 5$ for a $\mathrm{SDRT}^{\mathrm{OPT}}$ scheme associated with the SSP3s3o temporal scheme, i.e. $\nu = 0.05$. The visualization process on triangular grids is done by extrapolating the solution at SP to high-order Lagrange elements nodes.

To monitor the computation convergence, the L_2 norm of the residual on the density between iteration n and $n+1$ is computed using:

$$||\text{Res}||_2 = \sqrt{\frac{\int_\Omega (\rho^{n+1} - \rho^n)^2 \, d\Omega}{\int_\Omega d\Omega}}. \tag{7.18}$$

Integration is performed using the 175-points symmetric quadrature given by Wandzura and Xiao [192]. The decay of the residual against number of iteration for SDRT$_p$ schemes, $p \in [\![2, 5]\!]$, is shown for the transonic case in Fig. 7.2. Despite the use of an explicit time-integration scheme, a very good convergence is obtained.

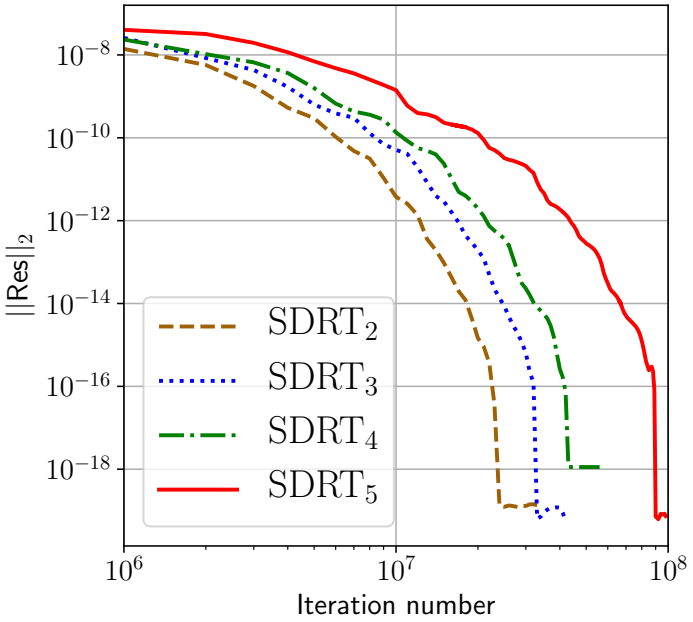
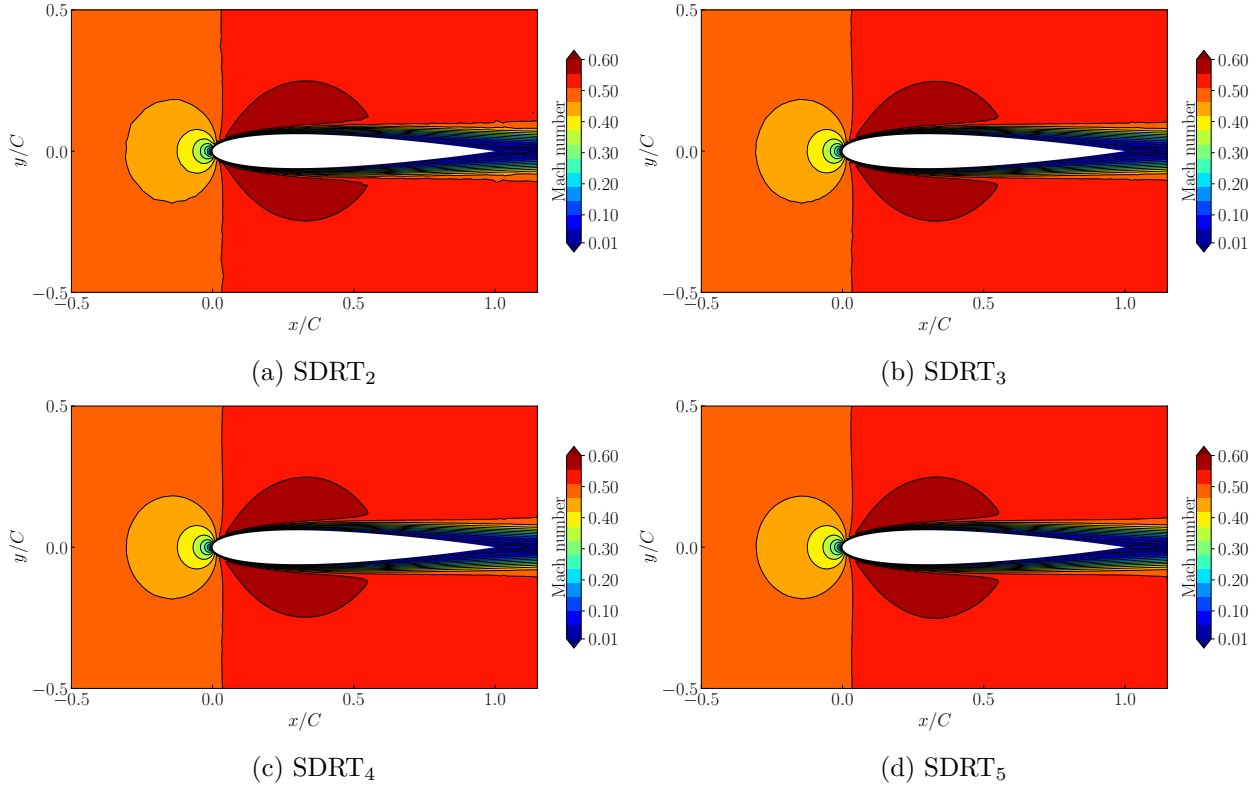


Figure 7.2. – Convergence of the residual for the transonic viscous flow over a NACA0012 airfoil

7.2.1. Case A: Symmetric Subsonic Flow

Results are first presented for Case A, which consists in a symmetric laminar flow around the NACA0012 airfoil defined by $M_\infty = 0.5$, $\alpha = 0°$, $Re = 5000$. Mach contours obtained using SDRT$_p$ schemes, $p \in [\![2, 5]\!]$ for the Case A are shown in Fig. 7.3. The separation region occurs close to the trailing edge, leading to the formation of a symmetric recirculation bubble where the flow is associated with a low Mach number.

As the degree of the polynomial reconstruction increases, the solution becomes smoother and thus more accurate. For SDRT$_2$ and SDRT$_3$ schemes (Fig. 7.3a, 7.3b), discontinuous contour lines can be observed. Those discontinuities are induced by the visualization process, which is done independently on each triangular element, leading to different solution values at cell interfaces, and express a low resolution. Mach contour lines become smoother for SDRT$_4$ (Fig. 7.3c) and perfectly continuous for SDRT$_5$ (7.3d). All SDRT schemes converge to the same steady solution.



(a) SDRT_2 (b) SDRT_3

(c) SDRT_4 (d) SDRT_5

Figure 7.3. – Case A: Mach number contours using SDRT_p schemes, $p \in [\![2, 5]\!]$

The surface skin-friction coefficient C_f and the surface pressure coefficient C_p distributions are plotted in Fig. 7.4 and Fig. 7.5 (respectively) for each accuracy order and compared to NASA results from [197] (finite volume solver RK/Implicit scheme with matrix dissipation on structured grid) on the most refined mesh (1024×512 elements).

Coefficients C_f and C_p are computed using:

$$C_f = \frac{\tau_\omega}{1/2\,\gamma\,\mathrm{M}_\infty^2\,P_\infty}, \tag{7.19}$$

$$C_p = \frac{P - P_\infty}{1/2\,\gamma\,\mathrm{M}_\infty^2\,P_\infty}. \tag{7.20}$$

In Eq. (7.19) and Eq. (7.20), the skin shear stress τ_ω and the pressure P need to be computed on the airfoil surface. They are obtained from an extrapolation of the variables at SP to points located at the surface. In the following results, variables are extrapolated to 100 surface points per edge and an averaging procedure is applied to obtain a single value at each cell interface.

From Fig. 7.4, the main effect of the SDRT scheme order of accuracy can be observed at the airfoil leading edge: the peak C_f value is overestimated by 13.8% when using SDRT_2 and 12.7% using SDRT_3. This overshoot could be due to the interpolation post-processing step performed independently on each cell and is most visible for smaller orders since less information per cell is available. For SDRT_4 and SDRT_5, a closer view shows a proper estimation of the maximum C_f value, with an overestimation of 0.8% for SDRT_4 and an underestimation of 2.5% for SDRT_5. For $p = 2, 3$ and 4, the maximum C_f value is slightly shifted: it is located at $x/C = 0.009$ whereas the NASA data led to a maximum C_f value located at $x/C = 0.007$. This offset is could due to the mesh refinement at the leading edge. The

$SDRT_5$ leads to a better estimation of the peak shape. Additionally, the $SDRT_2$ scheme overestimates the C_f coefficient around $x/C = 0.1$, whereas higher-order SDRT schemes are matching NASA data. The surface pressure coefficient plot (Fig. 7.5) shows a good agreement with the NASA data. A closer view highlights the convergence of the results when the accuracy order increases.

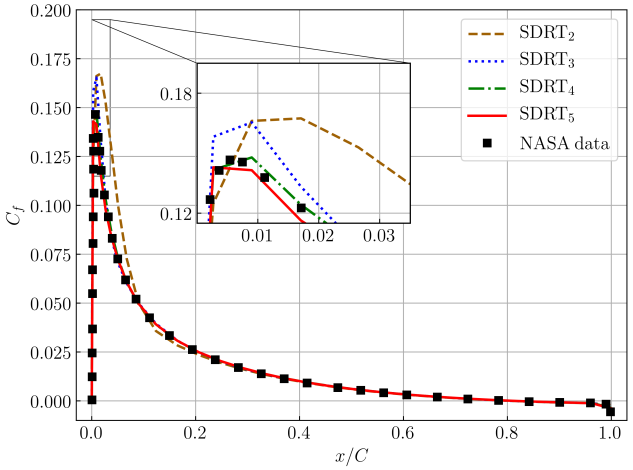


Figure 7.4. – Case A: Surface skin-friction coefficient C_f

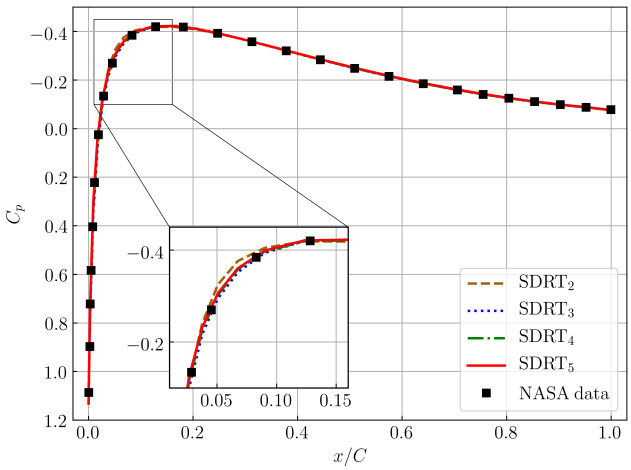


Figure 7.5. – Case A: Surface pressure coefficient C_p

The pressure drag coefficient $(C_D)_p$, the skin-friction drag coefficient $(C_D)_f$ and the total drag coefficient C_D are computed from the curves using [198]:

$$(C_D)_p = \left(\int_0^1 (C_{p_u} - C_{p_l})dy\right)\cos(\alpha) - \left(\int_0^1 (C_{p_u} - C_{p_l})dx\right)\sin(\alpha), \tag{7.21}$$

$$(C_D)_f = \int_0^1 (C_{f_u} + C_{f_l})dx, \tag{7.22}$$

$$C_D = (C_D)_p + (C_D)_f, \tag{7.23}$$

where the subscript u (respectively l) denote the upper (respectively lower) section of the airfoil, associated with $y > 0$ (respectively $y < 0$). Integrals are computed using the trapezoidal rule with more than 100 control points per edge. Their computed values and the location of the separation point relative to chord length are given in Table 7.1 and compared with NASA data and results using a fifth-order SD scheme on quadrilaterals.

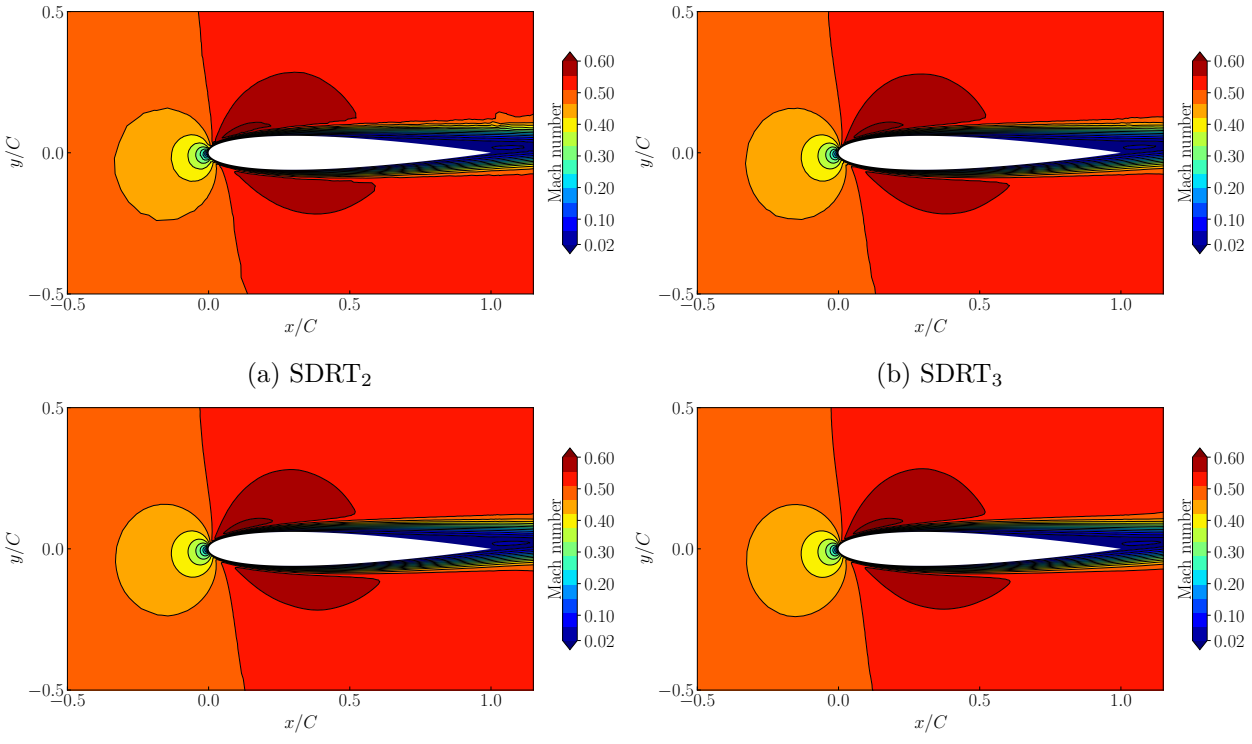
The overestimation of the surface skin-friction coefficient previously observed for $SDRT_2$ is retrieved on the $(C_D)_f$ value. Even if the $(C_D)_p$ value is in good agreement with the NASA data using the refined grid and the fifth-order SD scheme, this leads to an overestimated total drag coefficient. Values of the different coefficients converge as the order of accuracy increases. Results using $SDRT_4$ are in excellent agreement with the NASA data using the refined grid (difference of 0.1%), whereas $SDRT_5$ matches results obtained using the fifth-order SD scheme (difference of 0.7%). All of the separation point locations predicted by SDRT schemes lies in the interval given by the three references up to the third decimal.

	DoF Number	C_D	$(C_D)_p$	$(C_D)_f$	x_{sep}/C
$SDRT_2$	14,442	0.0583644	0.0220679	0.0362965	0.809
$SDRT_3$	24,070	0.0573619	0.0239848	0.0333771	0.808
$SDRT_4$	36,105	0.0554990	0.0225962	0.0329028	0.810
$SDRT_5$	50,547	0.0543702	0.0218255	0.0325447	0.814
NASA [197]	8,192	0.0560794	0.0237143	0.0323651	0.809
	524,288	0.0555743	0.0227887	0.0327855	0.808
Fifth-order SD [113]	43,200	0.05476	0.02225	0.03251	0.814

Table 7.1. – Case A: Comparison of drag coefficients and separation point location

7.2.2. Case B: Asymmetric Subsonic Flow

Results are now presented the subsonic laminar flow ($M_\infty = 0.5$, $Re = 5000$) with an incidence angle of $\alpha = 2°$. The Mach contours presented in Fig. 7.6 indicate the formation of a recirculating region, which length is larger than for Case A. The flow separation seems to occur further from the trailing edge. As for Case A, all SDRT schemes converge to the same steady solution, with better accuracy when the order increases.



(a) $SDRT_2$ (b) $SDRT_3$

(c) $SDRT_4$ (d) $SDRT_5$

Figure 7.6. – Case B: Mach number contours using $SDRT_p$ schemes, $p \in [\![2, 5]\!]$

The surface skin-friction coefficient C_f and the surface pressure coefficient C_p distributions are plotted in Fig. 7.7 and Fig. 7.8 (respectively) for each accuracy order and compared to NASA results from [197] on the most refined mesh (1024×512 elements). Both plots highlight the non-symmetry property of the flow for Case B.
Fig. 7.7 allows us to confirm the first comment on the separation point location made from the Mach contour plot: compared to Case A, the separation occurs approximately at mid-chord, on the up-

per surface. Starting from $x/C = 0.3$, all SDRT schemes lead to a good agreement with the NASA data. For $x/C < 0.3$, the SDRT_2 and SDRT_3 schemes do not manage to match the reference data. On the other hand, a close view at the leading edge allows us to see that SDRT_4 leads to a proper value of the surface skin-friction coefficient at the upper airfoil section and a slightly overestimated one for at the lower section. Finally, the SDRT_5 scheme was able to properly capture maximum C_f values in both the upper and the lower airfoil sections. In Fig. 7.8, the surface pressure coefficient plot obtained using SDRT schemes shows an acceptable shape but seems to be a little bit shifted by a positive value. As for Case A, a close view shows the convergence of the solution as the order increases.

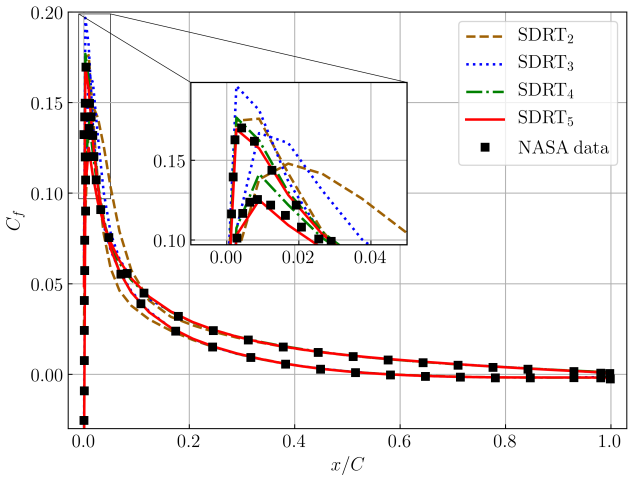


Figure 7.7. – Case B: Surface skin-friction coefficient C_f

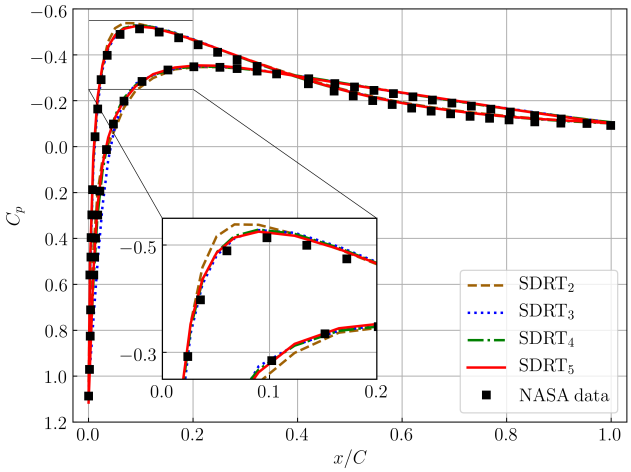


Figure 7.8. – Case B: Surface pressure coefficient C_p

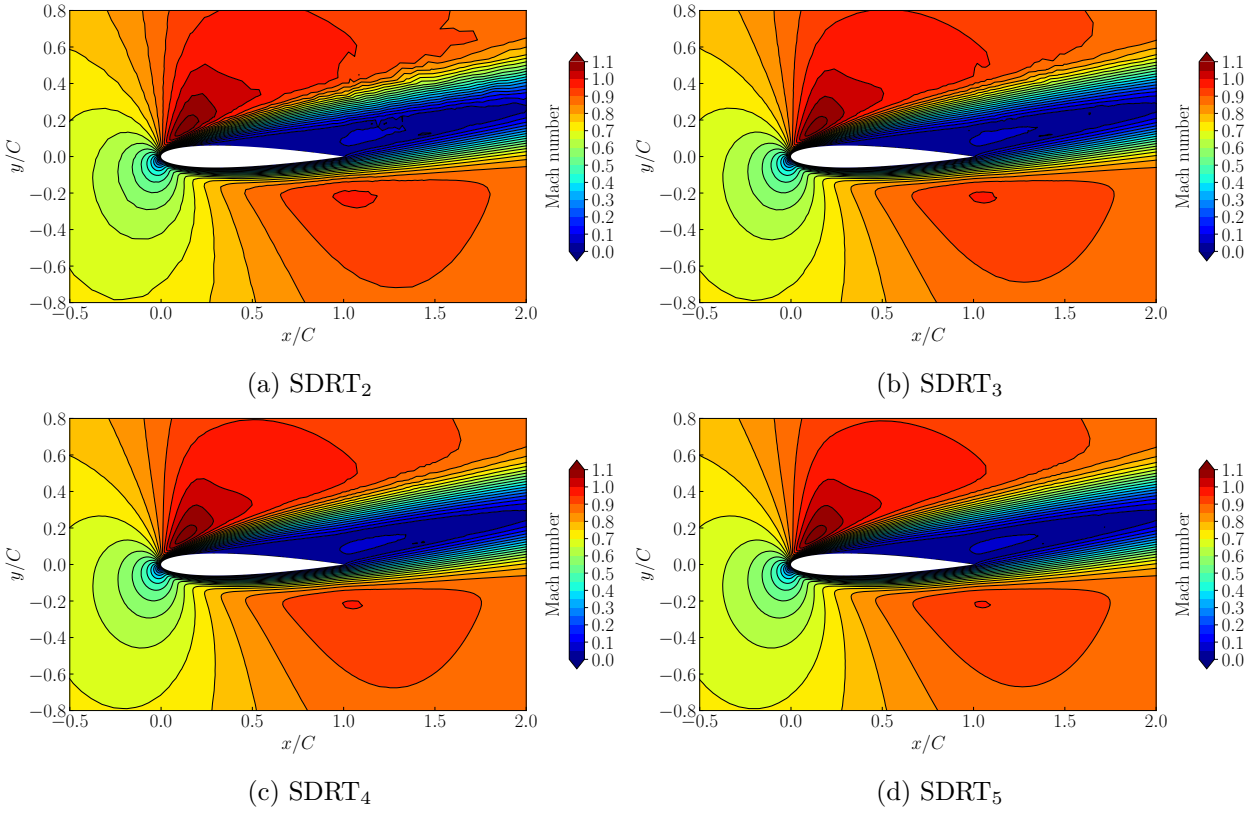
The aerodynamic coefficients are computed for each SDRT scheme and compared to NASA references in Table 7.2. The SDRT_2 scheme leads to a good prediction of C_D (1% of difference), but the surface skin-friction and pressure coefficients are not accurately determined ($\sim$ 4.5% of difference). For higher-order SDRT schemes, the C_D value converges to the NASA reference for the refined grid (13% for SDRT_3, 2.8% for SDRT_4 and 0.8% for SDRT_5), with close values for $(C_D)_p$ and $(C_D)_f$. The same behavior is observed for the separation point location, even if it is slightly overestimated (2.8% for $p = 5$).

	DoF Number	C_D	$(C_D)_p$	$(C_D)_f$	x_{sep}/C
SDRT_2	14,442	0.0574735	0.0233123	0.0341613	0.584
SDRT_3	24,070	0.0644525	0.0298255	0.0346270	0.580
SDRT_4	36,105	0.0584788	0.0252599	0.0332190	0.579
SDRT_5	50,547	0.0573627	0.0245549	0.0328078	0.577
NASA [197]	8,192	0.0572544	0.0251267	0.0321277	0.552
	524,288	0.0568914	0.0243173	0.0325741	0.561

Table 7.2. – Case B: Comparison of drag coefficients and separation point location

7.2.3. Case C: Transonic Flow

Eventually, results for the transonic flow (Case C) defined by $M_\infty = 0.8$, $\alpha = 10°$, $Re = 500$ are presented. Fig. 7.9 shows the Mach contours obtained using $SDRT_p$ schemes, $p \in [\![2, 5]\!]$. The flow is accelerated at the airfoil upper surface and creates a small supersonic zone ($M > 1$). However, as expected for this case, there is no shock wave developing. Accuracy improvement when using higher-order SDRT scheme is clearly visible for this case, especially for $SDRT_2$ (Fig. 7.9a) which reveals highly discontinuous Mach contours. The Mach contours given by the $SDRT_4$ and $SDRT_5$ schemes (Fig. 7.9c, 7.9d) show continuous lines for most of the domain. The remaining discontinuities located around the position $(x/C, y/C) = (1.4, 0.4)$ are due to the fact that the mesh used is refined for the wake given with an angle of attack $\alpha = 0°$. Apart from this region, the Mach contours obtained show that the $SDRT_4$ and $SDRT_5$ schemes converge to the same solution.



(a) $SDRT_2$ (b) $SDRT_3$

(c) $SDRT_4$ (d) $SDRT_5$

Figure 7.9. – Case C: Mach number contours using $SDRT_p$ schemes, $p \in [\![2, 5]\!]$

The surface skin-friction coefficient C_f and the surface pressure coefficient C_p distributions are plotted in Fig. 7.10 and Fig. 7.11. For both coefficients, there is good agreement between the results obtained from $SDRT_p$ schemes and the NASA data. In Fig. 7.10, a closer view shows that all SDRT schemes were able to capture the maximum value of the surface skin-friction coefficient at the leading edge. Small discontinuities between cells are observed, resulting from the interpolation post-processing step performed independently on each cell. The peak at the leading edge is quite accurately represented. A difference with the NASA data can be noticed at the trailing edge, where SDRT schemes did not manage to capture the maximum values due to low mesh refinement. A close view indicates that the SDRT C_f value gets a bit closer to the reference one when the order of accuracy increases. The surface pressure coefficient plot (Fig. 7.11) shows that all $SDRT_p$ schemes lead to excellent agreement with the NASA data, including at leading and trailing edges. As for the first two cases, results converge when the order increases.

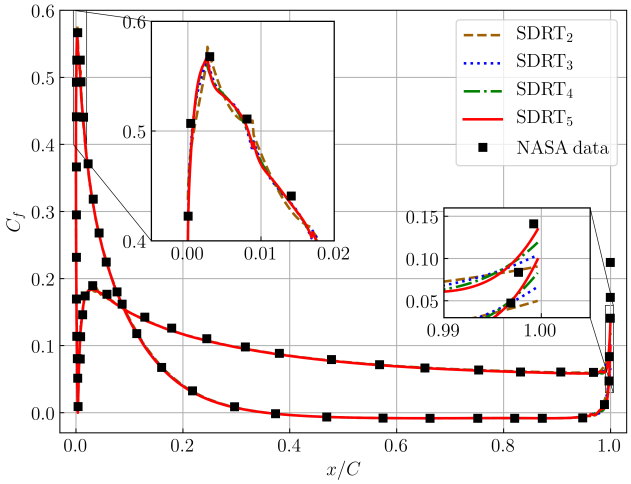


Figure 7.10. – Case C: Surface skin-friction coefficient C_f

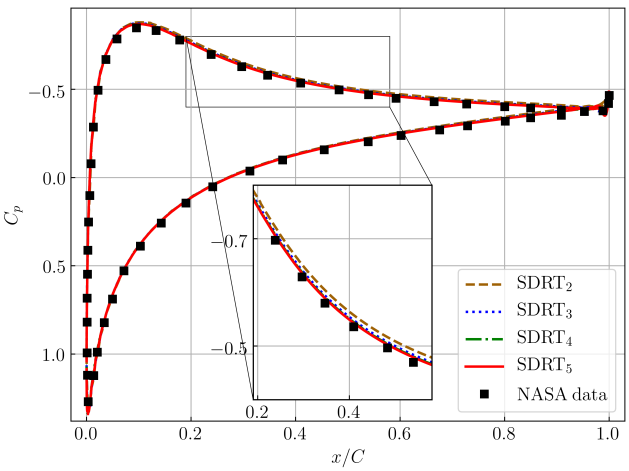


Figure 7.11. – Case C: Surface pressure coefficient C_p

The C_D, $(C_D)_p$, $(C_D)_f$, x_{sep}/C values are presented for SDRT$_p$ schemes and compared to the NASA values in Table 7.3. Results for the total drag coefficient are in excellent agreement with the NASA reference on the refined mesh, with a difference of $< 2\%$ for all schemes. For $p > 2$, all coefficients are converged up to the third decimal. The location of the separation point is slightly smaller than the reference ones but remains in good agreement.

	DoF Number	C_D	$(C_D)_p$	$(C_D)_f$	x_{sep}/C
SDRT$_2$	14,442	0.272783	0.148275	0.124508	0.357
SDRT$_3$	24,070	0.270953	0.147087	0.123867	0.356
SDRT$_4$	36,105	0.270255	0.146632	0.123622	0.355
SDRT$_5$	50,547	0.270224	0.146715	0.123509	0.355
NASA [197]	8,192	0.268071	0.146177	0.121894	0.386
	524,288	0.275155	0.147544	0.127611	0.362

Table 7.3. – Case C: Comparison of drag coefficients and separation point location

7.3. 2D Viscous Flow Around a Circular Cylinder

This test case aims to validate the method for the computation of viscous flow using 2D hybrid mesh. Let us consider a steady laminar viscous flow at $\mathrm{Re} = 20$ around a cylinder. The Mach number is $M_\infty = 0.1$ and the Reynolds number is defined by $\mathrm{Re} = \rho_\infty U_\infty d/\mu_{d\infty}$, where the dynamic viscosity is $\mu_{d\infty} = 1.853 \cdot 10^{-3}$ Pa s, and the cylinder diameter is $d = 1$ m. The density ρ_∞ and the velocity U_∞ can be deduced from the temperature $T = 300$ K and the constant ratio of specific heats $\gamma = 1.4$. The cylinder is placed in a rectangular domain. The farfield boundaries are located 10 diameters away from the cylinder in the upstream, upward and downward directions and 30 diameters away in the downstream direction. A hybrid mesh of 3427 elements is used, with 196 quadrilateral elements near the cylinder and 3231 triangles in the rest of the domain. A close view of the mesh is provided in Fig. 7.12. On the farfield boundary, the pressure, temperature and velocity are settled. At the cylinder surface, a no-slip isothermal wall condition is imposed. On quadrilateral elements, the standard SD method based on the interior FP located at the zeros of the corresponding Legendre polynomials is used. On triangles, the SDRT$^{\mathrm{WSJ}}$ scheme associated with the RK4J temporal scheme is used for $p \in [\![2, 5]\!]$ since a steady problem is considered. The same polynomial degree is used for both triangular and quadrilateral elements. Roe's Riemann solver is used to compute flux at interface flux points and the CFL number is set to 0.06 (the maximum one affordable using $p = 5$ for a SDRT$^{\mathrm{WSJ}}$ scheme associated with the RK4J temporal scheme). The computation convergence is monitored by computing the L_2 norm of the residual on the density between iteration n and $n + 1$ using:

$$||\mathrm{Res}||_2 = \sqrt{\frac{\int_\Omega (\rho^{n+1} - \rho^n)^2 \, d\Omega}{\int_\Omega d\Omega}}. \tag{7.24}$$

Integration is performed using the 175-points symmetric quadrature given by Wandzura and Xiao [192]. The decay of the residual against number of iteration for SD/SDRT$_p$ schemes, $p \in [\![2, 5]\!]$, is shown in Fig. 7.13. As for NACA test cases, computations are very well converged.

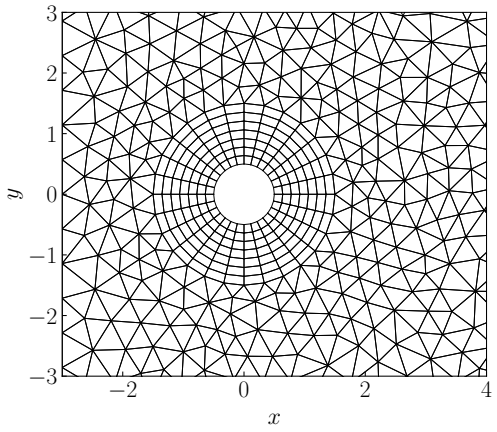


Figure 7.12. – Close view of the hybrid mesh for the viscous flow around a cylinder

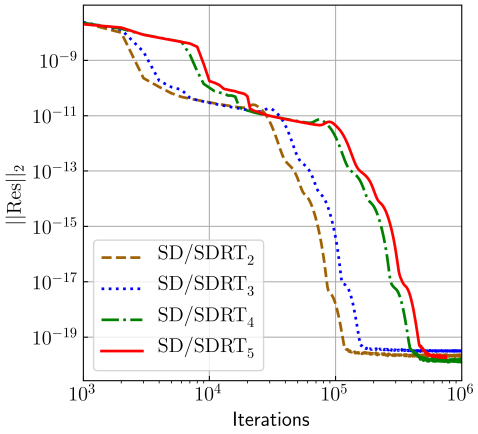
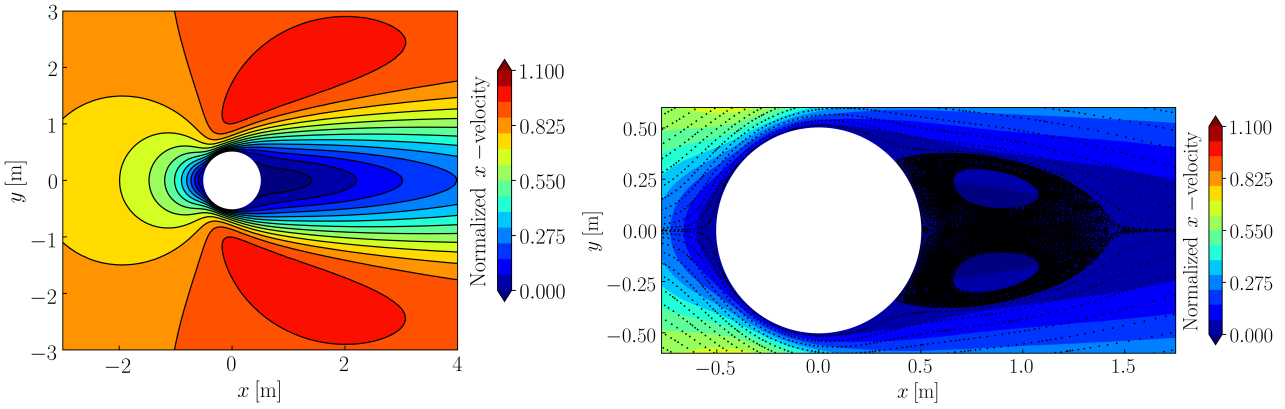


Figure 7.13. – Convergence of the residual for the viscous flow around circular cylinder

Figure 7.14 shows the normalized x-velocity contours and streamlines around the cylinder obtained using a SD/SDRT$_5$ scheme. Streamlines show a recirculation zone within the wake of the cylinder where two vortices are generated.



(a) Normalized x-velocity contours

(b) Close view of streamlines and normalized x-velocity contours

Figure 7.14. – Normalized x-velocity contours and streamlines around the cylinder using a SD/SDRT$_5$ scheme

The pressure coefficient C_p and the dimensionless negative vorticity $\bar{\omega}$ on the cylinder surface are computed using:

$$C_p = \frac{P - P_\infty}{1/2\,\gamma\, M_\infty^2\, P_\infty}, \tag{7.25}$$

$$\bar{\omega} = \frac{\mu_d \tau_\omega}{2\, M_\infty \sqrt{\gamma R_{gas} T}}, \tag{7.26}$$

where τ_ω is the skin shear stress and $R_{gas} = 287.05\ \mathrm{J\,kg^{-1}\,K^{-1}}$ is the specific gas constant. The surface pressure coefficient C_p and the dimensionless negative vorticity $\bar{\omega}$ are respectively plotted in Fig. 7.15 and Fig. 7.16 and compared to results obtained by Dennis and Chang [199]. Since linear quadrilateral elements were used, outputs on the cylinder surface shows high oscillations. Outputs data are thus post-treated using a Savitzky-Golay filter, which is a particular type of low-pass filter well adapted for data smoothing. All SD/SDRT schemes converge to the same solution. The surface pressure coefficient is slightly overestimated for $\theta \in [145°, 180°]$ but presents an excellent agreement for $\theta \in [0°, 145°]$. Likewise, the vorticity is overestimated for $\theta \in [110°, 150°]$. For example, at the peak value around $\theta = 125°$, there is a 5% difference between Dennis and Chang and SD/SDRT results.

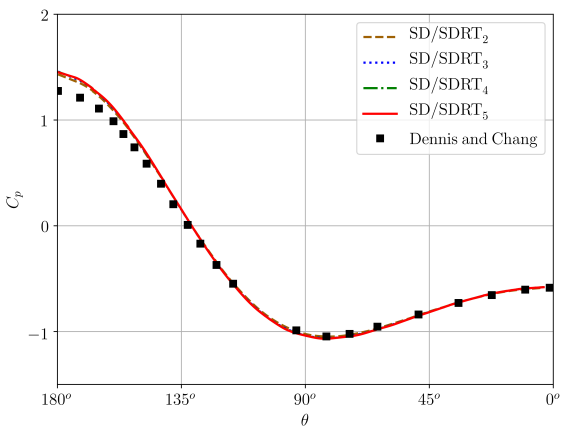


Figure 7.15. – Pressure coefficient C_p distribution on the cylinder surface

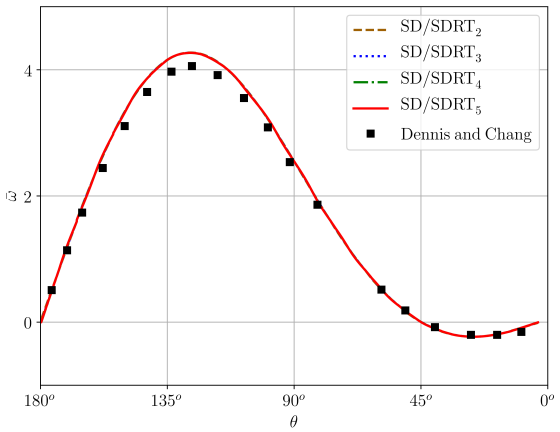


Figure 7.16. – Dimensionless negative vorticity $\bar{\omega}$ distribution on the cylinder surface

To properly quantify the mismatch between results obtained using SD/SDRT schemes and reference data from Dennis and Chang, the pressure drag coefficient $(C_D)_p$, the skin-friction drag coefficient $(C_D)_f$ and the total drag coefficient C_D are computed using:

$$(C_D)_p = \frac{1}{2}\int_0^{\pi} (C_{p_u} - C_{p_l}) \cos\theta d\theta, \tag{7.27}$$

$$(C_D)_f = \int_0^{1} (C_{f_u} + C_{f_l}) dx, \tag{7.28}$$

where

$$C_f = \frac{\tau_\omega}{1/2\,\gamma\, M_\infty^2\, P_\infty}, \tag{7.29}$$

and

$$C_D = (C_D)_p + (C_D)_f\,. \tag{7.30}$$

Values of the drag coefficients as well as the separation angle θ_{sep} and the normalized reattachment length L/d obtained using SD/SDRT schemes are compared with different reference values in Table 7.4. Values presented are computed after applying the Savitzky-Golay filter; however, note that the maximum difference between values on all coefficients obtained with and without the filter is around 0.2%. Reference results [199–201] are based on finite difference approximation and Cartesian grids. The SD/SDRT overestimation compared to Dennis and Chang data observed in Fig. 7.15 and Fig. 7.16 is retrieved on drag coefficients values. This overestimation could be due to the fact that linear quadrilateral elements were used. A mesh composed of quadratic elements could lead to better results. Additionally, using a no-slip adiabatic wall condition at the cylinder surface instead of an isothermal wall condition could be more appropriate. However, compared to other reference data, SD/SDRT schemes lead to a proper estimation of drag coefficients, separation angle and reattachment length. Finally, it can be noted that using the SSP3s3o temporal scheme (associated with a SDRT^{WSJ} implementation for $p \in [\![2,3]\!]$ and a SDRT^{OPT} implementation for $p \in [\![4,5]\!]$) led to the very same results, which is consistent since this is a steady test case.

	C_D	$(C_D)_p$	$(C_D)_f$	θ_{sep}	L/d
SD/SDRT_2	2.135	1.275	0.860	45.5°	0.915
SD/SDRT_3	2.145	1.284	0.860	45.4°	0.915
SD/SDRT_4	2.146	1.286	0.860	45.4°	0.916
SD/SDRT_5	2.147	1.287	0.860	45.3°	0.916
Dennis and Chang [199]	2.045	1.233	0.812	43.7°	0.94
Russell and Z. Jane Wang [200]	2.13	-	-	43.3°	0.94
Calhoun [201]	2.19	-	-	45.5°	0.91

Table 7.4. – Comparison of drag coefficients and separation angle for flow over a cylinder

7.4. 3D Taylor-Green Vortex

To validate the implementation of the SDRT method for the Navier-Stokes equations using tetrahedral grids, the Direct Numerical Simulation of the Taylor-Green Vortex (TGV) at $\mathrm{Re} = \rho_\infty U_\infty L/\mu_{d\infty} = 1600$ is considered. The TGV test case was proposed in the International Workshop on High-Order CFD Methods [193] to test the accuracy and performance of high-order methods. A three-dimensional periodic and transitional flow is considered and defined by:

$$U = U_\infty \sin\left(\frac{x}{L}\right)\cos\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right), \tag{7.31}$$

$$V = -U_\infty \cos\left(\frac{x}{L}\right)\sin\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right), \tag{7.32}$$

$$W = 0, \tag{7.33}$$

$$P = P_\infty + \frac{\rho_\infty U_\infty^2}{16}\left(\cos\left(\frac{2x}{L}\right) + \cos\left(\frac{2y}{L}\right)\right)\left(\cos\left(\frac{2z}{L}\right) + 2\right). \tag{7.34}$$

The flow is governed by the 3D compressible Navier-Stokes equations with constant physical properties and at low Mach number so that the obtained solutions are close to incompressible solutions. The test case conditions are summed up in Table 7.5.

Variable	Notation	Value	Unit
Reynolds number	Re	1600	-
Temperature	T_∞	300	K
Dynamic viscosity	$\mu_{d\infty}$	$1.846 \cdot 10^{-5}$	kg/m/s
Mach number	M_∞	0.1	-
Gas constant	R_{gas}	287.058	J/kg/K
Density	ρ_∞	$8.506 \cdot 10^{-4}$	$\mathrm{kg/m^3}$
Pressure	P_∞	73.254	Pa
Ratio of specific heat	γ	1.4	-
Prandtl number	Pr	0.71	-
Reference length	L	1.0	m

Table 7.5. – Flow conditions for the TGV test case

The computational domain is a cube defined by $\Omega = [-\pi L, \pi L]^3$ and periodic boundary conditions are imposed in the three directions. The SDRT implementation for interior FP is based on the Shunn-Ham quadrature rule. Solutions are time-integrated using the RKo6s temporal scheme and the time step Δt is imposed. Roe's Riemann solver is used to compute flux at cell interfaces. Computations are carried out on 600 processors. Three different regular grids (M_1, M_2 and M_3) are considered. Their number of elements and associated time steps are given in Table 7.6.

Scheme	Mesh	Number of Elements	DOF Number	Δt (sec)
	M_1	663,552	2,654,208	$7.5 \cdot 10^{-5}$
$SDRT_1$	M_2	1,572,864	6,291,456	$5.5 \cdot 10^{-5}$
	M_3	3,072,000	12,288,000	$4.5 \cdot 10^{-5}$
	M_1	663,552	6,635,520	$4.5 \cdot 10^{-5}$
$SDRT_2$	M_2	1,572,864	15,728,640	$3.5 \cdot 10^{-5}$
	M_3	3,072,000	30,720,000	$3.0 \cdot 10^{-5}$

Table 7.6. – Computational conditions for the TGV test case

The physical duration of the computation is based on the characteristic convective time $t_c = L/U_\infty$ and is set to $t_f = 20t_c$. The kinetic energy dissipation rate ε is computed for $t \in [0, t_f]$ and compared to a reference incompressible flow solution obtained using a dealiased pseudo-spectral code (developed at Université Catholique de Louvain, UCL) on a 512^3 mesh and provided by the International Workshop on High-Order CFD Methods [193]. The reference data is denoted 'Spectral-512^3'.

Kinetic Energy Dissipation Rate To compute the kinetic energy dissipation rate, one first needs to compute the kinetic energy E_k, defined by:

$$E_k(t) = \int_\Omega \frac{1}{2}\rho(U^2 + V^2 + W^2)\mathrm{d}\Omega. \tag{7.35}$$

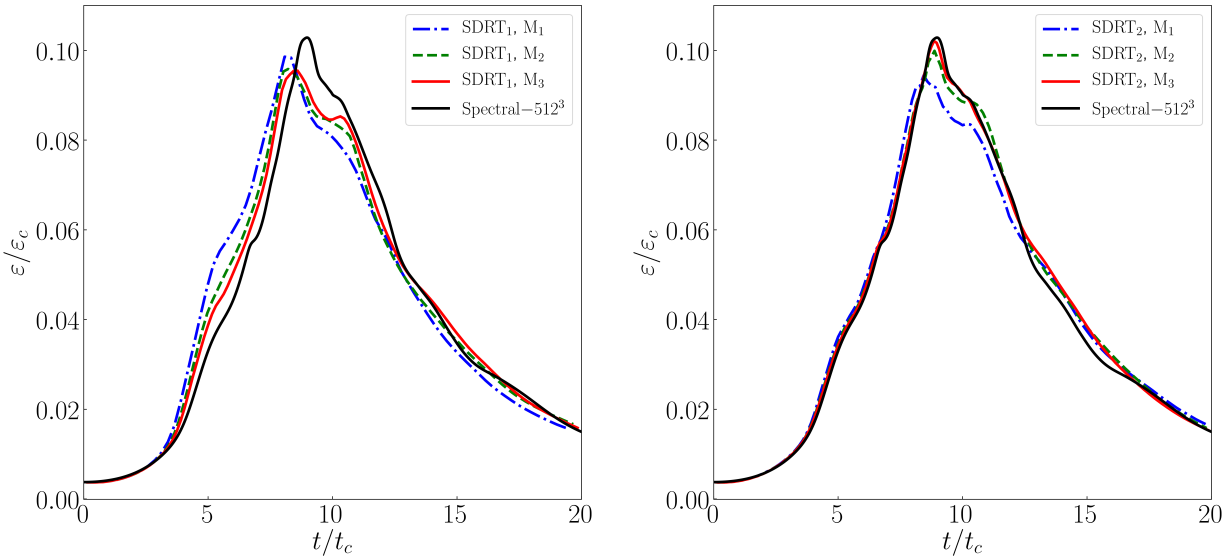
The kinetic energy is computed at each time t as:

$$E_k(t) = \sum_{i=1}^{N}\sum_{j=1}^{N_q} \omega_j |J^{(i,j)}| E_k^{(i)}(\boldsymbol{\xi}_j), \tag{7.36}$$

where $|J^{(i,j)}|$ is the Jacobian determinant at the j-th integration point of the i-th cell and N_q is the number of quadrature points. The quadrature points are located at $\boldsymbol{\xi}_j$ and associated with the weight ω_j. The integration is performed using the 84 points quadrature rule from [174]. The kinetic energy dissipation rate is defined by:

$$\varepsilon(t) = -\frac{\mathrm{d}E_k}{\mathrm{d}t}(t), \tag{7.37}$$

and is computed using a first-order upwind scheme. The kinetic energy dissipation rate is rendered dimensionless by $\varepsilon_c = E_k(t=0)/t_c$. Results obtained with the second and third-order SDRT schemes are compared with reference data in Fig. 7.17. Using the second-order SDRT scheme (Fig. 7.17a), the dimensionless kinetic energy dissipation rate $\varepsilon/\varepsilon_c$ evolution is first quite accurate ($t/t_c \in [0, 3]$) but grows too fast from $t/t_c > 3$ for all grid resolutions. The maximal peak value is underestimated (of 4% for M_1 and 7% for M_2 and M_3) and shifted (of 8% for M_1 and M_2 and 5% for M_3). However, results get closer to the reference data as the number of DOF increases. Using the second-order SDRT scheme leads to better results (Fig. 7.17b). For $t/t_c \in [0, 9]$, results obtained using the M_1 mesh slightly overestimate $\varepsilon/\varepsilon_c$ whereas results on M_2 and M_3 grids show an excellent agreement with the reference data. The improvement of the solution accuracy when the number of DOF increases can be clearly see at $t/t_c = 9$. Compared to the reference data, the peak value is particularly well predicted on the M_3 mesh (underestimation of 0.7%). For $t/t_c \in [12, 17]$, $\varepsilon/\varepsilon_c$ is a little overestimated but the final value at $t/t_c = 20$ matches the reference.



(a) SDRT$_1$ (b) SDRT$_2$

Figure 7.17. – Dimensionless kinetic energy dissipation rate obtained with second and third-order SDRT schemes compared to reference data

Computational Cost To assess the performance of the SDRT implementation on tetrahedral elements, the computational cost is compared to the one obtained with JAGUAR on hexahedral elements and to other solvers based on nodal Discontinuous Galerkin and Finite Difference method on hexahedral grids. All of them are based on explicit time integration. The description and efficiency of each code can be found on the NASA website gathering results of the 3^{rd} International Workshop on Higher-Order CFD Methods [202]. It should be noted that values presented in this section for the solver JAGUAR are preliminary results since they are based on a single computation. To compare the solvers' performance, the computational speed (in μsec) is computed as:

$$\text{Speed} = \frac{\text{Wall Time} \cdot \text{Nbr Cores}}{\text{Nbr It} \cdot \text{DoF}}. \tag{7.38}$$

Computational efficiency are compared in Fig. 7.18. The type of element used by each solver is indicated by markers (a square marker for hexahedral elements and a triangular marker for tetrahedral elements). Using JAGUAR, second-order schemes lead to an equivalent efficiency using either hexahedra or tetrahedra. However, from values obtained using third-order schemes, it becomes obvious that this first implementation for tetrahedra can be improved. The computational efficiency of the third-order SD scheme on hexahedra is around fifteen microseconds whereas the cost of the SDRT implementation on tetrahedra is twice higher. This statement is corroborated by the comparison of JAGUAR to other solvers. All of them lead to better efficiency than the SDRT implementation on tetrahedra, but the standard SD scheme on hexahedra remains quite competitive. This first SDRT implementation should thus be optimized in the future to be more efficient.

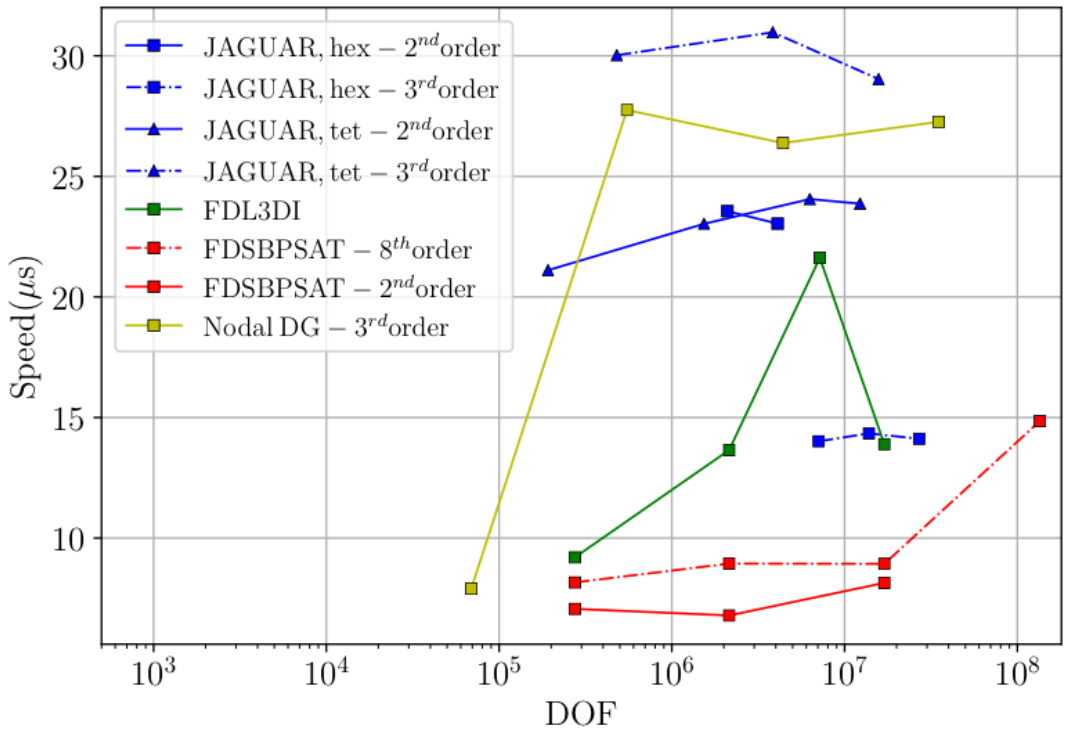


Figure 7.18. – Efficiency of the first SDRT implementation in JAGUAR compared with other solvers

Conclusion et Perspectives

Discussion

The objective of the present thesis was to extend the Spectral Difference method to simplex cells and hybrid grids. For two-dimensional grids, the stability of the SDRT approach first needed to be proven on triangular cells for orders of accuracy higher than four. For three-dimensional cases, the SDRT formulation on tetrahedral and prismatic elements had to be established.

Chapter 1 gave the context of the thesis and underlined the importance of high-order methods for Large Eddy Simulation. A bibliographic review showed the assets and drawbacks of the existing high-order discontinuous methods. The current limitations of the Spectral Difference method using Raviart-Thomas elements on simplex were pointed out. On triangles, the approach was limited to the fourth-order of accuracy due to a lack of stability proof for higher orders whereas the 3D SDRT implementation on tetrahedra was not stated yet.

In Chapter 2, the standard SD approach was recalled since tensor-product cells are involved in hybrid grids. The SDRT method was then detailed for triangles and established for tetrahedral cells. The formulation was extended to prismatic elements using a tensor product between the standard 1D SD and the 2D SDRT schemes. The complexity induced by the treatment of pyramidal elements was briefly addressed and a choice was made to focus on triangles and tetrahedra. This choice was motivated by the fact that those simplex cells offer the possibility to treat complex geometries.

The linear stability of the SDRT scheme was investigated in Chapter 3. To determine stable formulation on triangles for orders of accuracy higher than four, the goal was to study multiple SDRT implementations depending on the choice of the interior FP location. A new stability method was proposed as an efficient way to assess the stability of a scheme but it was demonstrated that results obtained on a domain composed of a fixed number of cells could not be extended in a general framework. This approach was thus put aside in favor of the common Fourier stability analysis. Under a Fourier stability analysis, different interior FP locations were tested using many existing quadrature rules. All of them led to small positive real part values in the spatial operator spectra for $p > 3$. Finally, the linear stability of the SDRT method was established on tetrahedral elements up to the third-order of accuracy.

Chapter 4 explored the possibility of stabilizing the SDRT method based on interior FP located at the Williams-Shunn-Jameson quadrature rule points (SDRT^{WSJ}) on triangles for $p > 3$ using the temporal discretization to dissipate the small positive real part values in the spatial operator spectra. Two options leading to stable coupled time-space discretizations were given. The first one uses the RK4J temporal scheme to dissipate the spatial instability and was validated for all advection angles. However, a high numerical dissipation error was found. The second one uses third-order SSP temporal schemes, which are associated with better dissipation and dispersion properties. However, the stability of the coupled time-space discretization is subject to conditions: for SDRT_4, the CFL number must lie into a given range, whereas for SDRT_5, the stability is not ensured for all advection angles. The strong dependence on the advection angle or on a double condition on the CFL number makes this second possibility impracticable. Additionally, the importance of conducting the Fourier analysis on

the coupled time-space discretization was highlighted since spatially stable SDRT scheme ($p = 2$ and $p = 3$) were shown as leading to an unstable time-space discretization when associated with temporal SSP schemes of order two for a CFL number $\nu > 0.02$. The coupled time-space discretization was finally examined for tetrahedral elements. All temporal schemes studied led to stable results when associated with the SDRT scheme for $p = 1$ and $p = 2$ but the severe stability condition for second-order SSP temporal schemes was also observed.

To overcome the restrictive conditions associated with the solution proposed in Chapter 4, Chapter 5 presented new sets of interior FP leading to spatially stable SDRT formulations (SDRT^{OPT}) for $p = 4$ and $p = 5$. These sets of points were determined through an optimization process based on the Fourier analysis. Proof of the spatial stability of this formulation was given through a study of the spatial operator spectrum and the coupled time-space discretization was then studied to determine CFL stability limits. The optimization process was also conducted for $p = 6$ on triangles and $p = 3$ on tetrahedra but did not lead to concluding results. However, on tetrahedral elements, the optimization problem only considered the interior FP location as an optimization parameter whereas the FP located on faces were fixed at Williams-Shunn-Jameson quadrature points. Since it was then demonstrated that the position of FP located on faces has an important impact on the scheme stability, an optimization problem considering both the interior FP location and the position of FP located on faces as optimization parameters could be able to determine stable SDRT formulations for $p > 2$.

In Chapter 6, the SDRT^{WSJ} and SDRT^{OPT} implementations were first compared in terms of accuracy for first-order Partial Differential Equations. As expected, the SDRT^{WSJ} stabilized by the RK4J temporal scheme led to high dissipation errors, fully driven by the RK4J temporal scheme. On the other hand, the high accuracy of the SDRT^{OPT} scheme was verified using a nonlinear Euler test case for triangular and 2D hybrid grids. A 3D Euler test case was finally used to retrieve the expected orders of accuracy on tetrahedral grids using SDRT schemes for $p = 1$ and $p = 2$.

Eventually, Chapter 7 presented Navier-Stokes applications. The SDRT^{OPT} implementation was used to simulate a steady flow over a NACA0012 airfoil on a quadratic triangular mesh for three different flow conditions. The SDRT^{WSJ} associated with the RK4J temporal scheme was then used to treat a 2D hybrid mesh and simulate the steady flow around a cylinder. Both SDRT implementations led to good agreement with reference data. Finally, the simulation of a Taylor-Green vortex was run using tetrahedral grids. Good agreement with reference data and consistent computational cost validated the 3D SDRT implementation.

Perspectives

In this thesis, the Spectral Difference approach was successfully extended to simplex cells and 2D hybrid grids. A linearly stable SDRT formulation was proposed based on interior FP locations determined using an optimization process up to the sixth-order of accuracy. The implementation of this formulation has allowed us to simulate various test cases using triangular and 2D hybrid grids. The extension of the SDRT method to tetrahedral elements and the proof of its stability up to the third-order has led to the simulation of a 3D turbulent flow.

Determination of Stable SDRT Schemes for Higher Orders

In the future, the determination of linearly stable SDRT schemes for higher orders, especially on tetrahedral elements, seems to be a key point.
To obtain stable SDRT discretizations associated with a fourth-order of accuracy on tetrahedral elements, an approach consisting in considering the location of FP located on faces as optimization parameters was proposed. It is important to highlight that if stable SDRT schemes were obtained by changing the position of FP located on faces, the FP on faces will not be located at the same coordinates as FP located on a prismatic element triangular face. Since the SP location has no impact on the scheme stability for triangular elements, a simple solution to solve that issue is to set the SP location on a triangle at the position of FP located on a tetrahedral face. This way, no mortar techniques will need to be introduced.
The determination of linearly stable SDRT schemes for higher orders on both triangular and tetrahedral elements could also be addressed using a different polynomial space to approximate the flux vector. The Raviart-Thomas space is indeed the smallest possible space to approximate the flux divergence; however, considering a larger space might lead to different conclusions.

Future Implementations

Another perspective would be to use implicit temporal schemes with the SDRT discretization in JAGUAR. In this thesis, only explicit time integration schemes were studied and implemented, leading to a high number of iterations needed to reach convergence on steady problems. Implicit schemes would allow us to use higher CFL numbers and thus reduce the computational time. At this moment, developments are ongoing to implement implicit temporal schemes in JAGUAR for tensor-product elements. This feature could then be extended to simplex cells.
Additionally, the hybrid mesh capability of the JAGUAR solver should be extended from the two-dimensional to the three-dimensional case. This requires implementing the SD/SDRT hybrid approach on prismatic elements and dealing with connectivity between hexahedral, tetrahedral and prismatic cells.

Extension to Complex Flows

Finally, now that the SDRT implementation has been validated in JAGUAR on basic aerodynamic test cases, more complex cases should be considered. For example, JAGUAR's ability to deal with combustion applications using hexahedral elements is currently developed. Extending it to tetrahedral elements would be the next objective, provided that the SDRT implementation is first made compatible with the non-reflecting boundary conditions. Computations of flows with shocks could also be of strong interest. This point is currently addressed in the 1D case using an entropy stable formulation. Finally, the aim is to use JAGUAR for industrial configurations to assess its performance and accuracy compared to other classical solvers.

Appendices

Appendix A

Jacobian Matrices

A.1. General Principle

For an element defined by n nodes, the geometrical transformation from the physical to the reference element takes the form:

$$\mathbf{x}\left(\boldsymbol{\xi}\right) = \sum_{i=1}^{n} M_i(\boldsymbol{\xi})\mathbf{x}_i, \tag{A.1}$$

where:

- $\mathbf{x}$ are the coordinates in the physical domain,
- $\boldsymbol{x_i}$ are the Cartesian coordinates in the physical domain of the i-th node of the cell,
- $\boldsymbol{\xi}$ are the coordinates in the reference domain and
- $M_i(\boldsymbol{\xi})$ are the shape functions.

The Jacobian matrix of the transformation is:

$$J = \left[\frac{\partial\,\mathbf{x}}{\partial\,\boldsymbol{\xi}}\right], \tag{A.2}$$

thus

$$J_{1D} = \left[\frac{\partial x}{\partial \xi}\right], \quad J_{2D} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} \end{bmatrix}, \quad J_{3D} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} & \dfrac{\partial x}{\partial \zeta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} & \dfrac{\partial y}{\partial \zeta} \\ \dfrac{\partial z}{\partial \xi} & \dfrac{\partial z}{\partial \eta} & \dfrac{\partial z}{\partial \zeta} \end{bmatrix}. \tag{A.3}$$

Injecting Eq. (A.1) into Eq. (A.3) leads to:

$$J_{1D} = \left[\frac{\partial x}{\partial \xi}\right] = \sum_{i=1}^{n} \frac{\partial M_i}{\partial \xi} x_i, \tag{A.4}$$

$$\begin{aligned} J_{2D} &= \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} \\ &= \begin{bmatrix} \displaystyle\sum_{i=1}^{n} \frac{\partial M_i}{\partial \xi} x_i & \displaystyle\sum_{i=1}^{n} \frac{\partial M_i}{\partial \eta} x_i \\ \displaystyle\sum_{i=1}^{n} \frac{\partial M_i}{\partial \xi} y_i & \displaystyle\sum_{i=1}^{n} \frac{\partial M_i}{\partial \eta} y_i \end{bmatrix} = \begin{bmatrix} x_1 & \cdots & x_i & \cdots & x_n \\ y_1 & \cdots & y_i & \cdots & y_n \end{bmatrix} \begin{bmatrix} \dfrac{\partial M_1}{\partial \xi} & \dfrac{\partial M_1}{\partial \eta} \\ \vdots & \vdots \\ \dfrac{\partial M_i}{\partial \xi} & \dfrac{\partial M_i}{\partial \eta} \\ \vdots & \vdots \\ \dfrac{\partial M_n}{\partial \xi} & \dfrac{\partial M_n}{\partial \eta} \end{bmatrix}, \end{aligned} \tag{A.5}$$

$$J_{3D} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1}^{n} \frac{\partial M_i}{\partial \xi} x_i & \sum_{i=1}^{n} \frac{\partial M_i}{\partial \eta} x_i & \sum_{i=1}^{n} \frac{\partial M_i}{\partial \zeta} x_i \\ \sum_{i=1}^{n} \frac{\partial M_i}{\partial \xi} y_i & \sum_{i=1}^{n} \frac{\partial M_i}{\partial \eta} y_i & \sum_{i=1}^{n} \frac{\partial M_i}{\partial \zeta} y_i \\ \sum_{i=1}^{n} \frac{\partial M_i}{\partial \xi} z_i & \sum_{i=1}^{n} \frac{\partial M_i}{\partial \eta} z_i & \sum_{i=1}^{n} \frac{\partial M_i}{\partial \zeta} z_i \end{bmatrix} = \begin{bmatrix} x_1 & \cdots & x_i & \cdots & x_n \\ y_1 & \cdots & y_i & \cdots & y_n \\ z_1 & \cdots & z_i & \cdots & z_n \end{bmatrix} \begin{bmatrix} \frac{\partial M_1}{\partial \xi} & \frac{\partial M_1}{\partial \eta} & \frac{\partial M_1}{\partial \zeta} \\ \vdots & \vdots & \vdots \\ \frac{\partial M_i}{\partial \xi} & \frac{\partial M_i}{\partial \eta} & \frac{\partial M_i}{\partial \zeta} \\ \vdots & \vdots & \vdots \\ \frac{\partial M_n}{\partial \xi} & \frac{\partial M_n}{\partial \eta} & \frac{\partial M_n}{\partial \zeta} \end{bmatrix}. \tag{A.6}$$

Given that the transformation is non-singular, the inverse transformation is related to the Jacobian according to:

$$J_{1D}^{-1} = \frac{\partial \xi}{\partial x} = \frac{1}{|J_{1D}|}, \tag{A.7}$$

$$J_{2D}^{-1} = \frac{\partial(\xi, \eta)}{\partial(x, y)} = \frac{1}{|J_{2D}|} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial x}{\partial \eta} \\ -\frac{\partial y}{\partial \xi} & \frac{\partial x}{\partial \xi} \end{bmatrix}, \tag{A.8}$$

$$J_{3D}^{-1} = \frac{\partial(\xi, \eta, \zeta)}{\partial(x, y, z)} = \frac{1}{|J_{3D}|} \begin{bmatrix} \frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \eta} & \frac{\partial x}{\partial \zeta}\frac{\partial z}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial z}{\partial \zeta} & \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial y}{\partial \eta} \\ \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \zeta} & \frac{\partial x}{\partial \xi}\frac{\partial z}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial z}{\partial \xi} & \frac{\partial x}{\partial \zeta}\frac{\partial y}{\partial \xi} - \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \zeta} \\ \frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \eta} - \frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \xi} & \frac{\partial x}{\partial \eta}\frac{\partial z}{\partial \xi} - \frac{\partial x}{\partial \xi}\frac{\partial z}{\partial \eta} & \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} \end{bmatrix}. \tag{A.9}$$

The Jacobian matrix and its inverse are fully expressed through the coordinates of the nodes in the physical domain and the shape functions. In the following sections, the shape functions are given for different elements.

A.2. 1D Elements

A.2.1. Linear Line - n = 2

For a linear line, the geometric transformation is expressed in its polynomial basis $(1, \xi)$ as:

$$x(\xi) = a + b\xi = \begin{bmatrix} 1 & \xi \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}, \tag{A.10}$$

where a, b are unknowns. The reference linear line is chosen as $\mathcal{S} := \{\xi, 0 \leq \xi \leq 1\}$ (Fig. A.1). The transformation is nodal, thus each physical node verifies Eq. (A.10):

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x(0) \\ x(1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}, \tag{A.11}$$

$$\iff \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{A.12}$$

Injecting Eq. (A.12) into Eq. (A.10):

$$\begin{aligned} x(\xi) &= [1 \quad \xi] \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= [1-\xi \quad \xi] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= [M_1 \quad M_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \end{aligned} \tag{A.13}$$

The shape functions and their derivatives are:

$$[M]^\top = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} = \begin{bmatrix} 1-\xi \\ \xi \end{bmatrix}, \tag{A.14}$$

$$\left[\frac{\partial M}{\partial \xi}\right]^\top = \begin{bmatrix} -1 \\ 1 \end{bmatrix}. \tag{A.15}$$

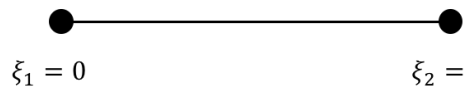


Figure A.1. – Linear line reference element

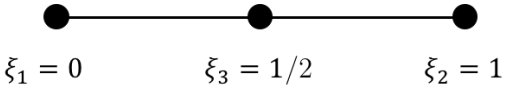


Figure A.2. – Quadratic line reference element

A.2.2. Quadratic Line - $\mathbf{n = 3}$

For a quadratic line, the geometric transformation is expressed in its polynomial basis $(1, \xi, \xi^2)$ as:

$$x(\xi) = a + b\xi + x\xi^2 = \begin{bmatrix} 1 & \xi & \xi^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \tag{A.16}$$

where a, b, c are unknowns. Points are numbered following the CGNS convention (Fig. A.2). The transformation is nodal, thus each physical node verifies Eq. (A.16):

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x(0) \\ x(1) \\ x(1/2) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1/2 & 1/4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \tag{A.17}$$

$$\iff \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1/2 & 1/4 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{A.18}$$

Injecting Eq. (A.18) into Eq. (A.16):

$$\begin{aligned} x(\xi) &= \begin{bmatrix} 1 & \xi & \xi^2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1/2 & 1/4 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\ &= \begin{bmatrix} 1 - 3\xi + 2\xi^2 & 2\xi^2 - \xi & 4(\xi - \xi^2) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\ &= \begin{bmatrix} M_1 & M_2 & M_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \end{aligned} \tag{A.19}$$

The shape functions and their derivatives are:

$$[M]^\top = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 1 - 3\xi + 2\xi^2 \\ 2\xi^2 - \xi \\ 4(\xi - \xi^2) \end{bmatrix}, \tag{A.20}$$

$$\left[\frac{\partial M}{\partial \xi}\right]^\top = \begin{bmatrix} 4\xi - 3 \\ 4\xi - 1 \\ 4 - 8\xi \end{bmatrix}. \tag{A.21}$$

A.3. 2D Elements

A.3.1. Linear Triangle - n = 3

The geometric transformation is expressed in its polynomial basis $(1, \xi, \eta)$ as:

$$x(\xi, \eta) = a + b\xi + c\eta = \begin{bmatrix} 1 & \xi & \eta \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \tag{A.22}$$

where a, b, c are unknowns. The reference triangle is chosen as $\mathcal{T} := \{(\xi, \eta) : 0 \leq \xi, \eta \leq 1, \xi + \eta \leq 1\}$ (Fig. A.3). The transformation is nodal, thus each physical node verifies Eq. (A.22):

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x(0,0) \\ x(1,0) \\ x(0,1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \tag{A.23}$$

$$\iff \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{A.24}$$

Injecting Eq. (A.24) into Eq. (A.22):

$$\begin{aligned} x(\xi, \eta) &= \begin{bmatrix} 1 & \xi & \eta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\ &= \begin{bmatrix} 1 - \xi - \eta & \xi & \eta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\ &= \begin{bmatrix} M_1 & M_2 & M_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \end{aligned} \tag{A.25}$$

The shape functions are:

$$M_1 = 1 - \xi - \eta, \quad M_2 = \xi, \quad M_3 = \eta. \tag{A.26}$$

The shape functions and their derivatives are:

$$[M]^\top = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 1 - \xi - \eta \\ \xi \\ \eta \end{bmatrix}, \tag{A.27}$$

$$\left[\frac{\partial M}{\partial \xi}\right]^\top = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}, \quad \left[\frac{\partial M}{\partial \eta}\right]^\top = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}. \tag{A.28}$$

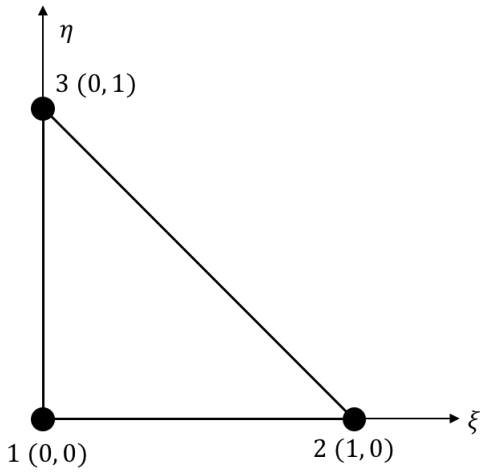


Figure A.3. – Linear triangular reference element

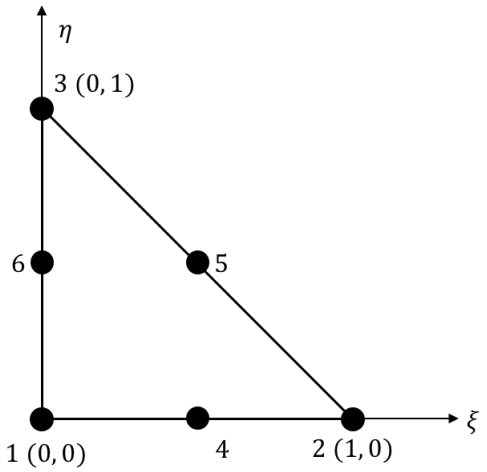


Figure A.4. – Quadratic triangular reference element

A.3.2. Quadratic Triangle - n = 6

The geometric transformation is expressed in its polynomial basis $(1, \xi, \eta, \xi^2, \xi\eta, \eta^2)$ as:

$$x(\xi, \eta) = a + b\xi + c\eta + d\xi^2 + e\xi\eta + f\eta^2 = \begin{bmatrix} 1 & \xi & \eta & \xi^2 & \xi\eta & \eta^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}, \tag{A.29}$$

where a, b, c, d, e, f are unknowns. Points are numbered following the CGNS convention (Fig. A.4). The transformation is nodal, thus each physical node verifies Eq. (A.29):

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} x(0,0) \\ x(1,0) \\ x(0,1) \\ x(1/2,0) \\ x(1/2,1/2) \\ x(0,1/2) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1/2 & 0 & 1/4 & 0 & 0 \\ 1 & 1/2 & 1/2 & 1/4 & 1/4 & 1/4 \\ 1 & 0 & 1/2 & 0 & 0 & 1/4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}, \tag{A.30}$$

$$\iff \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1/2 & 0 & 1/4 & 0 & 0 \\ 1 & 1/2 & 1/2 & 1/4 & 1/4 & 1/4 \\ 1 & 0 & 1/2 & 0 & 0 & 1/4 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}. \tag{A.31}$$

Injecting Eq. (A.31) into Eq. (A.29):

$$\begin{aligned} x(\xi,\eta) &= \begin{bmatrix} 1 & \xi & \eta & \xi^2 & \xi\eta & \eta^2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1/2 & 0 & 1/4 & 0 & 0 \\ 1 & 1/2 & 1/2 & 1/4 & 1/4 & 1/4 \\ 1 & 0 & 1/2 & 0 & 0 & 1/4 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \\ &= \begin{bmatrix} 1-3(\xi+\eta)+2(\xi^2+\eta^2)+4\xi\eta & 2\xi^2-\xi & 2\eta^2-\eta & 4(\xi-\xi\eta-\xi^2) & 4\xi\eta & 4(\eta-\xi\eta-\eta^2) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}. \\ &= \begin{bmatrix} M_1 & M_2 & M_3 & M_4 & M_5 & M_6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \end{aligned} \tag{A.32}$$

The shape functions and their derivatives are:

$$[M]^\top = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \end{bmatrix} = \begin{bmatrix} 1-3(\xi+\eta)+2(\xi^2+\eta^2)+4\xi\eta \\ 2\xi^2-\xi \\ 2\eta^2-\eta \\ 4(\xi-\xi\eta-\xi^2) \\ 4\xi\eta \\ 4(\eta-\xi\eta-\eta^2) \end{bmatrix}, \tag{A.33}$$

$$\left[\frac{\partial M}{\partial \xi}\right]^\top = \begin{bmatrix} -3+4\xi+4\eta \\ 4\xi-1 \\ 0 \\ 4(1-\eta-2\xi) \\ 4\eta \\ -4\eta \end{bmatrix}, \quad \left[\frac{\partial M}{\partial \eta}\right]^\top = \begin{bmatrix} -3+4\eta+4\xi \\ 0 \\ 4\eta-1 \\ -4\xi \\ 4\xi \\ 4(1-\xi-2\eta) \end{bmatrix}. \tag{A.34}$$

A.3.3. Linear Quadrilateral - $\mathbf{n = 4}$

The geometric transformation is based on the polynomial basis $(1, \xi) \times (1, \eta)$. The reference quadrilateral is chosen as $\mathcal{Q} = \{(\xi, \eta), 0 \leq \xi, \eta \leq 1\}$ (Fig. A.5). The shape functions are simply obtained by multiplying the 1D shape function of a linear line with respect to the ξ and η direction:

$$[M]^\top = \begin{bmatrix} M_1(\xi,\eta) \\ M_2(\xi,\eta) \\ M_3(\xi,\eta) \\ M_4(\xi,\eta) \end{bmatrix} = \begin{bmatrix} M_1^{1D}(\xi)\, M_1^{1D}(\eta) \\ M_2^{1D}(\xi)\, M_1^{1D}(\eta) \\ M_2^{1D}(\xi)\, M_2^{1D}(\eta) \\ M_1^{1D}(\xi)\, M_2^{1D}(\eta) \end{bmatrix} = \begin{bmatrix} (1-\xi)(1-\eta) \\ \xi(1-\eta) \\ \xi\eta \\ (1-\xi)\eta \end{bmatrix}. \tag{A.35}$$

Their derivatives are given by:

$$\left[\frac{\partial M}{\partial \xi}\right]^\top = \begin{bmatrix} \dfrac{\partial M_1^{1D}(\xi)}{\partial \xi}\, M_1^{1D}(\eta) \\ \dfrac{\partial M_2^{1D}(\xi)}{\partial \xi}\, M_1^{1D}(\eta) \\ \dfrac{\partial M_2^{1D}(\xi)}{\partial \xi}\, M_2^{1D}(\eta) \\ \dfrac{\partial M_1^{1D}(\xi)}{\partial \xi}\, M_2^{1D}(\eta) \end{bmatrix} = \begin{bmatrix} -(1-\eta) \\ (1-\eta) \\ \eta \\ -\eta \end{bmatrix}, \tag{A.36}$$

$$\left[\frac{\partial M}{\partial \eta}\right]^\top = \begin{bmatrix} M_1^{1D}(\xi)\, \dfrac{\partial M_1^{1D}(\eta)}{\partial \eta} \\ M_2^{1D}(\xi)\, \dfrac{\partial M_1^{1D}(\eta)}{\partial \eta} \\ M_2^{1D}(\xi)\, \dfrac{\partial M_2^{1D}(\eta)}{\partial \eta} \\ M_1^{1D}(\xi)\, \dfrac{\partial M_2^{1D}(\eta)}{\partial \eta} \end{bmatrix} = \begin{bmatrix} -(1-\xi) \\ -\xi \\ \xi \\ (1-\xi) \end{bmatrix}. \tag{A.37}$$

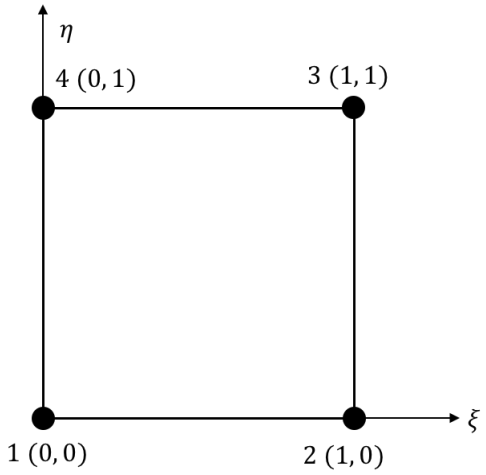


Figure A.5. – Linear quadrilateral reference element

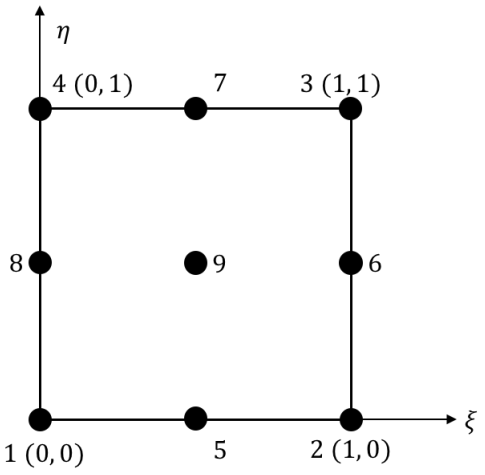


Figure A.6. – Quadratic quadrilateral reference element

A.3.4. Quadratic Quadrilateral - n = 9

The geometric transformation is based on the polynomial basis $(1, \xi, \xi^2) \times (1, \eta, \eta^2)$. Points are numbered following the CGNS convention (Fig. A.6). The shape functions are simply obtained by multiplying the 1D shape function of a quadratic line with respect to the ξ and η direction:

$$[M]^\top = \begin{bmatrix} M_1(\xi,\eta) \\ M_2(\xi,\eta) \\ M_3(\xi,\eta) \\ M_4(\xi,\eta) \\ M_5(\xi,\eta) \\ M_6(\xi,\eta) \\ M_7(\xi,\eta) \\ M_8(\xi,\eta) \\ M_9(\xi,\eta) \end{bmatrix} = \begin{bmatrix} M_1^{1D}(\xi)\, M_1^{1D}(\eta) \\ M_2^{1D}(\xi)\, M_1^{1D}(\eta) \\ M_2^{1D}(\xi)\, M_2^{1D}(\eta) \\ M_1^{1D}(\xi)\, M_2^{1D}(\eta) \\ M_3^{1D}(\xi)\, M_1^{1D}(\eta) \\ M_2^{1D}(\xi)\, M_3^{1D}(\eta) \\ M_3^{1D}(\xi)\, M_2^{1D}(\eta) \\ M_1^{1D}(\xi)\, M_3^{1D}(\eta) \\ M_3^{1D}(\xi)\, M_3^{1D}(\eta) \end{bmatrix} = \begin{bmatrix} (1-3\xi+2\xi^2)(1-3\eta+2\eta^2) \\ (2\xi^2-\xi)(1-3\eta+2\eta^2) \\ (2\xi^2-\xi)(2\eta^2-\eta) \\ (1-3\xi+2\xi^2)(2\eta^2-\eta) \\ 4(\xi-\xi^2)(1-3\eta+2\eta^2) \\ (2\xi^2-\xi)4(\eta-\eta^2) \\ 4(\xi-\xi^2)(2\eta^2-\eta) \\ (1-3\xi+2\xi^2)4(\eta-\eta^2) \\ 4(\xi-\xi^2)4(\eta-\eta^2) \end{bmatrix}. \tag{A.38}$$

Their derivatives are given by:

$$\left[\frac{\partial M}{\partial \xi}\right]^\top = \begin{bmatrix} \dfrac{\partial M_1^{1D}(\xi)}{\partial \xi}\, M_1^{1D}(\eta) \\ \dfrac{\partial M_2^{1D}(\xi)}{\partial \xi}\, M_1^{1D}(\eta) \\ \dfrac{\partial M_2^{1D}(\xi)}{\partial \xi}\, M_2^{1D}(\eta) \\ \dfrac{\partial M_1^{1D}(\xi)}{\partial \xi}\, M_2^{1D}(\eta) \\ \dfrac{\partial M_3^{1D}(\xi)}{\partial \xi}\, M_1^{1D}(\eta) \\ \dfrac{\partial M_2^{1D}(\xi)}{\partial \xi}\, M_3^{1D}(\eta) \\ \dfrac{\partial M_3^{1D}(\xi)}{\partial \xi}\, M_2^{1D}(\eta) \\ \dfrac{\partial M_1^{1D}(\xi)}{\partial \xi}\, M_3^{1D}(\eta) \\ \dfrac{\partial M_3^{1D}(\xi)}{\partial \xi}\, M_3^{1D}(\eta) \end{bmatrix} = \begin{bmatrix} (4\xi-3)(1-3\eta+2\eta^2) \\ (4\xi-1)(1-3\eta+2\eta^2) \\ (4\xi-1)(2\eta^2-\eta) \\ (4\xi-3)(2\eta^2-\eta) \\ (4-8\xi)(1-3\eta+2\eta^2) \\ (4\xi-1)(4(\eta-\eta^2)) \\ (4-8\xi)(2\eta^2-\eta) \\ (4\xi-3)(4(\eta-\eta^2)) \\ (4-8\xi)(4(\eta-\eta^2)) \end{bmatrix}, \tag{A.39}$$

$$\left[\frac{\partial M}{\partial \eta}\right]^\top = \begin{bmatrix} M_1^{1D}(\xi)\,\dfrac{\partial M_1^{1D}(\eta)}{\partial \eta} \\ M_2^{1D}(\xi)\,\dfrac{\partial M_1^{1D}(\eta)}{\partial \eta} \\ M_2^{1D}(\xi)\,\dfrac{\partial M_2^{1D}(\eta)}{\partial \eta} \\ M_1^{1D}(\xi)\,\dfrac{\partial M_2^{1D}(\eta)}{\partial \eta} \\ M_3^{1D}(\xi)\,\dfrac{\partial M_1^{1D}(\eta)}{\partial \eta} \\ M_2^{1D}(\xi)\,\dfrac{\partial M_3^{1D}(\eta)}{\partial \eta} \\ M_3^{1D}(\xi)\,\dfrac{\partial M_2^{1D}(\eta)}{\partial \eta} \\ M_1^{1D}(\xi)\,\dfrac{\partial M_3^{1D}(\eta)}{\partial \eta} \\ M_3^{1D}(\xi)\,\dfrac{\partial M_3^{1D}(\eta)}{\partial \eta} \end{bmatrix} = \begin{bmatrix} (1-3\xi+2\xi^2)(4\eta-3) \\ (2\xi^2-\xi)(4\eta-3) \\ (2\xi^2-\xi)(4\eta-1) \\ (1-3\xi+2\xi^2)(4\eta-1) \\ (4(\xi-\xi^2))(4\eta-3) \\ (2\xi^2-\xi)(4-8\eta) \\ (4(\xi-\xi^2))(4\eta-1) \\ (1-3\xi+2\xi^2)(4-8\eta) \\ (4(\xi-\xi^2))(4-8\eta) \end{bmatrix}. \tag{A.40}$$

A.4. 3D Elements

A.4.1. Linear Tetrahedron - $\mathbf{n = 4}$

The geometric transformation is expressed in its polynomial basis $(1, \xi, \eta, \zeta)$ as:

$$x(\xi,\eta,\zeta) = a + b\xi + c\eta + d\zeta = \begin{bmatrix} 1 & \xi & \eta & \zeta \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}, \tag{A.41}$$

where a, b, c, d are unknowns. The transformation is nodal, thus each physical node verifies Eq. (A.41):

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x(0,0,0) \\ x(1,0,0) \\ x(0,1,0) \\ x(0,0,1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}, \tag{A.42}$$

$$\iff \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}. \tag{A.43}$$

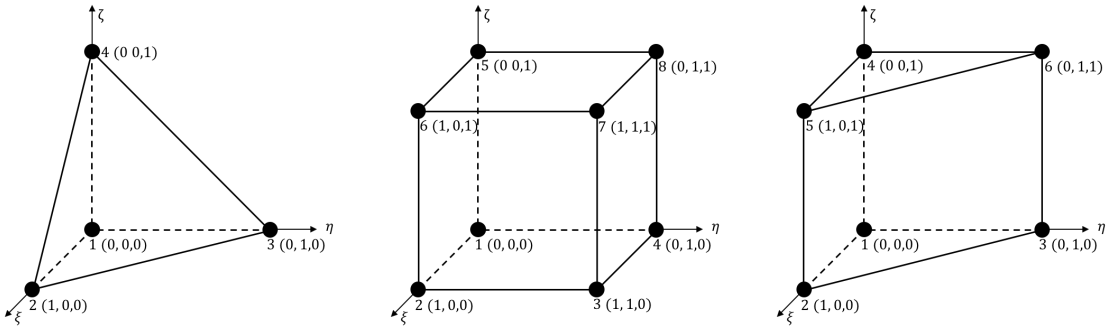
Injecting Eq. (A.43) into Eq. (A.41):

$$\begin{aligned} x(\xi,\eta,\zeta) &= \begin{bmatrix} 1 & \xi & \eta & \zeta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \\ &= \begin{bmatrix} 1-\xi-\eta-\zeta & \xi & \eta & \zeta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \\ &= \begin{bmatrix} M_1 & M_2 & M_3 & M_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}. \end{aligned} \tag{A.44}$$

The shape functions and their derivatives are:

$$[M]^\top = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix} = \begin{bmatrix} 1-\xi-\eta-\zeta \\ \xi \\ \eta \\ \zeta \end{bmatrix}, \tag{A.45}$$

$$\left[\frac{\partial M}{\partial \xi}\right]^\top = \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \left[\frac{\partial M}{\partial \eta}\right]^\top = \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \left[\frac{\partial M}{\partial \zeta}\right]^\top = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \tag{A.46}$$



(a) Tetrahedron (b) Hexahedron (c) Prism

Figure A.7. – 3D linear reference elements

A.4.2. Linear Prism - n = 6

The geometric transformation is based on the polynomial basis $(1, \xi, \eta) \times (1, \zeta)$. The reference prismatic element is chosen as $\mathcal{P} := \{(\xi, \eta) : 0 \leq \xi, \eta \leq 1, \xi + \eta \leq 1\} \times \{\zeta : 0 \leq \zeta \leq 1\}$and points are numbered as shown in Fig. A.7c. The shape functions are obtained by multiplying the 2D shape function of a linear triangle with the 1D shape function of a linear line with respect to the ζ direction:

$$[M]^\top = \begin{bmatrix} M_1(\xi,\eta,\zeta) \\ M_2(\xi,\eta,\zeta) \\ M_3(\xi,\eta,\zeta) \\ M_4(\xi,\eta,\zeta) \\ M_5(\xi,\eta,\zeta) \\ M_6(\xi,\eta,\zeta) \end{bmatrix} = \begin{bmatrix} M_1^{2D}(\xi,\eta)\, M_1^{1D}(\zeta) \\ M_2^{2D}(\xi,\eta)\, M_1^{1D}(\zeta) \\ M_3^{2D}(\xi,\eta)\, M_1^{1D}(\zeta) \\ M_1^{2D}(\xi,\eta)\, M_2^{1D}(\zeta) \\ M_2^{2D}(\xi,\eta)\, M_2^{1D}(\zeta) \\ M_3^{2D}(\xi,\eta)\, M_2^{1D}(\zeta) \end{bmatrix} = \begin{bmatrix} (1-\xi-\eta)\,(1-\zeta) \\ \xi\,(1-\zeta) \\ \eta\,(1-\zeta) \\ (1-\xi-\eta)\,\zeta \\ \xi\,\zeta \\ \eta\,\zeta \end{bmatrix}. \tag{A.47}$$

Their derivatives are given by:

$$\left[\frac{\partial M}{\partial \xi}\right]^\top = \begin{bmatrix} -(1-\zeta) \\ (1-\zeta) \\ 0 \\ -\zeta \\ \zeta \\ 0 \end{bmatrix}, \left[\frac{\partial M}{\partial \eta}\right]^\top = \begin{bmatrix} -(1-\zeta) \\ 0 \\ (1-\zeta) \\ -\zeta \\ 0 \\ \zeta \end{bmatrix}, \left[\frac{\partial M}{\partial \zeta}\right]^\top = \begin{bmatrix} -(1-\xi-\eta) \\ -\xi \\ -\eta \\ (1-\xi-\eta) \\ \xi \\ \eta \end{bmatrix}. \tag{A.48}$$

A.4.3. Linear Hexahedron - n = 8

The geometric transformation is based on the polynomial basis $(1, \xi) \times (1, \eta) \times (1, \zeta)$. The reference hexahedron is chosen as $\mathcal{H} = \{(\xi, \eta, \zeta), 0 \leq \xi, \eta, \zeta \leq 1\}$ and points are numbered as shown in Fig. A.7b. The shape functions are simply obtained by multiplying the 1D shape function of a linear line with respect to the ξ, η and ζ direction:

$$[M]^\top = \begin{bmatrix} M_1(\xi,\eta,\zeta) \\ M_2(\xi,\eta,\zeta) \\ M_3(\xi,\eta,\zeta) \\ M_4(\xi,\eta,\zeta) \\ M_5(\xi,\eta,\zeta) \\ M_6(\xi,\eta,\zeta) \\ M_7(\xi,\eta,\zeta) \\ M_8(\xi,\eta,\zeta) \end{bmatrix} = \begin{bmatrix} M_1^{1D}(\xi)\, M_1^{1D}(\eta)\, M_1^{1D}(\zeta) \\ M_2^{1D}(\xi)\, M_1^{1D}(\eta)\, M_1^{1D}(\zeta) \\ M_2^{1D}(\xi)\, M_2^{1D}(\eta)\, M_1^{1D}(\zeta) \\ M_1^{1D}(\xi)\, M_2^{1D}(\eta)\, M_1^{1D}(\zeta) \\ M_1^{1D}(\xi)\, M_1^{1D}(\eta)\, M_2^{1D}(\zeta) \\ M_2^{1D}(\xi)\, M_1^{1D}(\eta)\, M_2^{1D}(\zeta) \\ M_2^{1D}(\xi)\, M_2^{1D}(\eta)\, M_2^{1D}(\zeta) \\ M_1^{1D}(\xi)\, M_2^{1D}(\eta)\, M_2^{1D}(\zeta) \end{bmatrix} = \begin{bmatrix} (1-\xi)\,(1-\eta)\,(1-\zeta) \\ \xi\,(1-\eta)\,(1-\zeta) \\ \xi\,\eta\,(1-\zeta) \\ (1-\xi)\,\eta\,(1-\zeta) \\ (1-\xi)\,(1-\eta)\,\zeta \\ \xi\,(1-\eta)\,\zeta \\ \xi\,\eta\,\zeta \\ (1-\xi)\,\eta\,\zeta \end{bmatrix}. \tag{A.49}$$

Their derivatives are given by:

$$\left[\frac{\partial M}{\partial \xi}\right]^\top = \begin{bmatrix} -\,(1-\eta)\,(1-\zeta) \\ (1-\eta)\,(1-\zeta) \\ \eta\,(1-\zeta) \\ -\,\eta\,(1-\zeta) \\ -\,(1-\eta)\,\zeta \\ (1-\eta)\,\zeta \\ \eta\,\zeta \\ -\,\eta\,\zeta \end{bmatrix}, \left[\frac{\partial M}{\partial \eta}\right]^\top = \begin{bmatrix} -(1-\xi)\,(1-\zeta) \\ -\xi\,(1-\zeta) \\ \xi\,(1-\zeta) \\ (1-\xi)\,(1-\zeta) \\ -(1-\xi)\,\zeta \\ -\xi\,\zeta \\ \xi\,\zeta \\ (1-\xi)\,\zeta \end{bmatrix}, \left[\frac{\partial M}{\partial \zeta}\right]^\top = \begin{bmatrix} -(1-\xi)\,(1-\eta) \\ -\xi\,(1-\eta) \\ -\xi\,\eta \\ -(1-\xi)\,\eta \\ (1-\xi)\,(1-\eta) \\ \xi\,(1-\eta) \\ \xi\,\eta \\ (1-\xi)\,\eta \end{bmatrix}. \tag{A.50}$$

Appendix B

Demonstration of L^2 Orthogonality

B.1. 2D

To demonstrate the L^2 orthogonality of the PKD basis functions, the following integral must be evaluated:

$$\langle \Phi_{i,j} | \Phi_{k,l} \rangle = \int_{-1}^{1} \int_{-1}^{-y} \Phi_{i,j} \Phi_{k,l} dx dy, \tag{B.1}$$

which can be written in terms of the collapsed coordinate system (ξ, η):

$$\langle \Phi_{i,j} | \Phi_{k,l} \rangle = \int_{-1}^{1} \int_{-1}^{1} P_i^{0,0}(\xi) \left(\frac{1-\eta}{2} \right)^i P_j^{2i+1,0}(\eta) P_k^{0,0}(\xi) \left(\frac{1-\eta}{2} \right)^k P_l^{2k+1,0}(\eta) \, |J| \, d\xi d\eta, \tag{B.2}$$

where the Jacobian determinant is

$$|J| = \left| \frac{\partial(x,y)}{\partial(\xi,\eta)} \right| = \frac{1-\eta}{2}. \tag{B.3}$$

Eq. (B.2) can be written:

$$\begin{aligned} \langle \Phi_{i,j} | \Phi_{k,l} \rangle &= \int_{-1}^{1} \int_{-1}^{1} \left(\frac{1-\eta}{2} \right)^{i+k+1} P_i^{0,0}(\xi) P_j^{2i+1,0}(\eta) P_k^{0,0}(\xi) P_l^{2k+1,0}(\eta) d\xi d\eta \\ &= \frac{1}{2^{i+k+1}} \int_{-1}^{1} P_i^{0,0}(\xi) P_k^{0,0}(\xi) d\xi \int_{-1}^{1} (1-\eta)^{i+k+1} P_j^{2i+1,0}(\eta) P_l^{2k+1,0}(\eta) d\eta. \end{aligned} \tag{B.4}$$

The first integral value follows from the orthogonality of Legendre polynomials, which are the special case $(\alpha, \beta) = (0, 0)$ of Jacobi polynomials:

$$\int_{-1}^{1} P_i^{0,0}(\xi) P_k^{0,0}(\xi) d\xi = \frac{2\delta_{ik}}{2i+1}. \tag{B.5}$$

The first integral is equal to zero if $i \neq k$. When $i = k$, using Eq. (2.62), the second integral becomes:

$$\int_{-1}^{1} (1-\eta)^{i+k+1} P_j^{2i+1,0}(\eta) P_l^{2k+1,0}(\eta) d\eta = \int_{-1}^{1} (1-\eta)^{2i+1} P_j^{2i+1,0}(\eta) P_l^{2i+1,0}(\eta) d\eta = \frac{2^{2i+2} \delta_{jl}}{2j+2i+2}. \tag{B.6}$$

Eq. (B.4) then becomes:

$$\langle \Phi_{i,j} | \Phi_{k,l} \rangle = \frac{\delta_{ik} \delta_{jl}}{(i+1/2)(i+j+1)}. \tag{B.7}$$

From Eq. (B.7), if $i \neq k$ and $j \neq l$, $\langle \Phi_{i,j} | \Phi_{k,l} \rangle$ is necessarily equal to zero, which shows the orthogonality of the PKD basis in $\mathcal{T}^{PKD}$.

B.2. 3D

The integral to evaluate is:

$$\langle \Phi_{i,j,k} | \Phi_{l,m,n} \rangle = \int_{-1}^{1} \int_{-1}^{-z} \int_{-1}^{-1-y-z} \Phi_{i,j,k} \Phi_{l,m,n} dx dy dz, \tag{B.8}$$

which can be written in terms of the collapse coordinate system (ξ, η, ζ):

$$\begin{aligned}\langle \Phi_{i,j,k} | \Phi_{l,m,n} \rangle = \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} P_i^{0,0}(\xi) \left(\frac{1-\eta}{2}\right)^i P_j^{2i+1,0}(\eta) \left(\frac{1-\zeta}{2}\right)^{i+j} P_k^{2(i+j+1),0}(\zeta) \\ P_l^{0,0}(\xi) \left(\frac{1-\eta}{2}\right)^l P_m^{2l+1,0}(\eta) \left(\frac{1-\zeta}{2}\right)^{l+m} P_n^{2(l+m+1),0}(\zeta) \, |J| \, d\xi d\eta d\zeta,\end{aligned} \tag{B.9}$$

where the Jacobian determinant is:

$$|J| = \left| \frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)} \right| = \frac{1-\eta}{2} \left(\frac{1-\zeta}{2}\right)^2. \tag{B.10}$$

Eq. (B.9) can be written

$$\begin{aligned}\langle \Phi_{i,j,k} | \Phi_{l,m,n} \rangle = \int_{-1}^{1} P_i^{0,0}(\xi) P_l^{0,0}(\xi) d\xi \int_{-1}^{1} \left(\frac{1-\eta}{2}\right)^{i+l+1} P_j^{2i+1,0}(\eta) P_m^{2l+1,0}(\eta) d\eta \\ \int_{-1}^{1} \left(\frac{1-\zeta}{2}\right)^{i+j+l+m+2} P_k^{2(i+j+1),0}(\zeta) P_n^{2(l+m+1),0}(\zeta) d\zeta.\end{aligned} \tag{B.11}$$

As for the 2D case, the first integral is:

$$\int_{-1}^{1} P_i^{0,0}(\xi) P_l^{0,0}(\xi) d\xi = \frac{\delta_{il}}{(i+1/2)}. \tag{B.12}$$

The first integral is equal to zero if $i \neq l$. When $i = l$, using Eq. (2.62), the second integral becomes:

$$\int_{-1}^{1} \left(\frac{1-\eta}{2}\right)^{i+l+1} P_j^{2i+1,0}(\eta) P_m^{2l+1,0}(\eta) d\eta = \frac{\delta_{jm}}{(i+j+1)}. \tag{B.13}$$

The second integral is equal to zero if $j \neq m$. When $i = l$ and $j = m$, using Eq. (2.62), the third integral is evaluated as:

$$\begin{aligned}\int_{-1}^{1} \left(\frac{1-\zeta}{2}\right)^{i+j+l+m+2} P_k^{2(i+j+1),0}(\zeta) P_n^{2(l+m+1),0}(\zeta) d\zeta &= \frac{1}{2^{2(i+j+1)}} \\ &\int_{-1}^{1} (1-\zeta)^{2(i+j+1)} P_k^{2(i+j+1),0}(\zeta) P_n^{2(i+j+1),0}(\zeta) d\zeta \\ &= \frac{1}{2^{2(i+j+1)}} \frac{2^{2(i+j+1)+1}}{2k+2(i+j+1)+1} \delta_{kn} \\ &= \frac{\delta_{kn}}{i+j+k+3/2}.\end{aligned} \tag{B.14}$$

Eventually, Eq. (B.8) becomes:

$$\langle \Phi_{i,j,k} | \Phi_{l,m,n} \rangle = \frac{\delta_{il} \delta_{jm} \delta_{kn}}{(i+1/2)(i+j+1)(i+j+k+3/2)}. \tag{B.15}$$

As for the 2D case, if $i \neq l$, $j \neq m$ and $k \neq n$, $\langle \Phi_{i,j,k} | \Phi_{l,m,n} \rangle$ is necessarily equal to zero, which shows the orthogonality of the PKD basis in $\mathcal{T}_e^{PKD}$.

B.3. Normalized PKD Basis

From Eq. (B.7) and Eq. (B.15), it comes:

$$\|\Phi_{i,j}\|^2_{\mathcal{T}^{PKD}} = \langle \Phi_{i,j} | \Phi_{i,j} \rangle = \frac{1}{(i+1/2)(i+j+1)}, \tag{B.16}$$

$$\|\Phi_{i,j,k}\|^2_{\mathcal{T}_e^{PKD}} = \langle \Phi_{i,j,k} | \Phi_{i,j,k} \rangle = \frac{1}{(i+1/2)(i+j+1)(i+j+k+3/2)}, \tag{B.17}$$

which leads to the expression of the normalized 2D and 3D PKD basis:

$$\Phi_{i,j} = \sqrt{(i+1/2)(i+j+1)}\, P_i^{0,0}(\xi) \left(\frac{1-\eta}{2}\right)^i P_j^{2i+1,0}(\eta), \quad i+j \leq p, \tag{B.18}$$

$$\Phi_{i,j,k} = \sqrt{(i+1/2)(i+j+1)(i+j+k+3/2)}\, P_i^{0,0}(\xi) \left(\frac{1-\eta}{2}\right)^i P_j^{2i+1,0}(\eta) \left(\frac{1-\zeta}{2}\right)^{i+j} P_k^{2(i+j+1),0}(\zeta), \quad i+j+k \leq p. \tag{B.19}$$

Appendix C

Stability Analysis on a Fixed Number of Cells - Standard One-Dimensional Spectral Difference Scheme

To further verify the consistency of the FNC stability analysis with the common Fourier stability analysis, the semi-discretized matrix form is integrated in time using the low-storage second-order six-stage Runge-Kutta scheme of Bogey and Bailly (RKo6s) [187]. Stability limits were determined by Vanharen *et al.* [133] for this discretization through the maximum allowable CFL number based on a matrix form of the Fourier analysis by requiring the spectral radius of the matrix $\mathbf{G}$ to be lower than 1.

To investigate the dependency of the approach to the number of cells, $\max\left(\mathrm{Re}\left(\lambda_G\right)\right)$ is plotted as a function of the CFL number for a number of cells $N \in [\![1, 10]\!]$ in Fig. C.1.

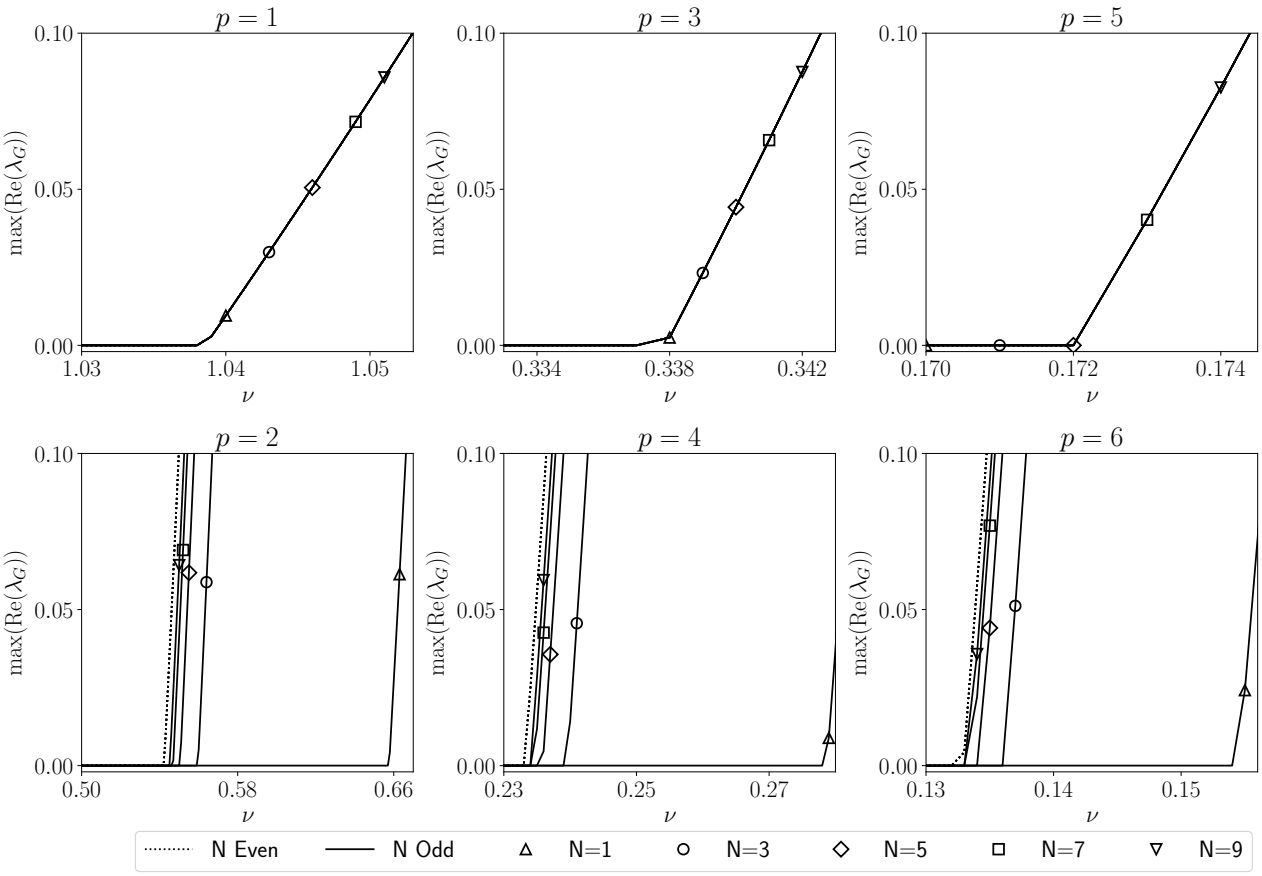


Figure C.1. – CFL stability limits for the 1D SD scheme determined using FNC analysis depending on the number of cells

From Fig. C.1, the existence of a pattern based on the number of degree of freedom can be noted. If p is odd, then the stability limit is always found to be the same and corresponds exactly to the maximum CFL given by the Fourier analysis in [133]. However, if p is even, two cases are possible: if the number of cells N is even, then the stability limit is always the same (and perfectly matches the results presented in [133]); if N is odd, then the stability limit is different, and will converge to the value obtained with a N even when N increases. This results seems to be a consequence of the nature of the total number of DoF value:

- p is odd and N is odd $\rightarrow p+1$ is even $\rightarrow N_{DoF} = (p+1)N$ is even,
- p is odd and N is even $\rightarrow p+1$ is even $\rightarrow N_{DoF} = (p+1)N$ is even,
- p is even and N is even $\rightarrow p+1$ is odd $\rightarrow N_{DoF} = (p+1)N$ is even,
- p is even and N is odd $\rightarrow p+1$ is odd $\rightarrow N_{DoF} = (p+1)N$ is odd.

The only case where N_{DoF} is odd is if p is even and N is odd. The fact that the total number of DoF is odd might influence the periodicity and thus lead to different behavior. When N increases, the stability limit tends to the 'standard' one. However, the assumption that a small number of cells is sufficient to represent the scheme behavior is undermined here. For example, if the analysis is conducted for $p = 4$ and $N = 3$, the stability limit numerically determined will be higher than the one given by the Fourier analysis and thus lead to unstable computations on a larger mesh. Those results allow us to conclude that the approach is consistent with the common Fourier analysis results, but leads to results for a particular domain that can not be considered as general conclusions. For the 1D case, one can suppose that the approach is representative of the scheme behavior on any mesh provided that N_{DoF} is even, but such a hypothesis can not be directly extended to triangular elements.

Appendix D

Matrices Formulation for the Fourier Analysis

D.1. Triangular Elements

The matrices $\mathbf{M}^{0,0}$, $\mathbf{M}^{-1,0}$, $\mathbf{M}^{+1,0}$, $\mathbf{M}^{0,-1}$ and $\mathbf{M}^{0,+1}$ involved in the SDRT spatial discretization for the Fourier analysis on triangular elements (Eq. (3.22)) are detailed in this appendix. Those matrices are given as:

$$\mathbf{M}^{0,0} = \begin{bmatrix} \mathbf{D}_{jk} & O_{N_{SP},N_{FP}} \\ O_{N_{SP},N_{FP}} & \mathbf{D}_{jk} \end{bmatrix} \mathbf{C}^{0,0} \begin{bmatrix} \mathbf{T}_{kj} & O_{N_{FP},N_{SP}} \\ O_{N_{FP},N_{SP}} & \mathbf{T}_{kj} \end{bmatrix}, \tag{D.1}$$

$$\mathbf{M}^{-1,0} = \begin{bmatrix} \mathbf{D}_{jk} & O_{N_{SP},N_{FP}} \\ O_{N_{SP},N_{FP}} & \mathbf{D}_{jk} \end{bmatrix} \mathbf{C}^{-1,0} \begin{bmatrix} \mathbf{T}_{kj} & O_{N_{FP},N_{SP}} \\ O_{N_{FP},N_{SP}} & \mathbf{T}_{kj} \end{bmatrix}, \tag{D.2}$$

$$\mathbf{M}^{+1,0} = \begin{bmatrix} \mathbf{D}_{jk} & O_{N_{SP},N_{FP}} \\ O_{N_{SP},N_{FP}} & \mathbf{D}_{jk} \end{bmatrix} \mathbf{C}^{+1,0} \begin{bmatrix} \mathbf{T}_{kj} & O_{N_{FP},N_{SP}} \\ O_{N_{FP},N_{SP}} & \mathbf{T}_{kj} \end{bmatrix}, \tag{D.3}$$

$$\mathbf{M}^{0,-1} = \begin{bmatrix} \mathbf{D}_{jk} & O_{N_{SP},N_{FP}} \\ O_{N_{SP},N_{FP}} & \mathbf{D}_{jk} \end{bmatrix} \mathbf{C}^{0,-1} \begin{bmatrix} \mathbf{T}_{kj} & O_{N_{FP},N_{SP}} \\ O_{N_{FP},N_{SP}} & \mathbf{T}_{kj} \end{bmatrix}, \tag{D.4}$$

$$\mathbf{M}^{0,+1} = \begin{bmatrix} \mathbf{D}_{jk} & O_{N_{SP},N_{FP}} \\ O_{N_{SP},N_{FP}} & \mathbf{D}_{jk} \end{bmatrix} \mathbf{C}^{0,+1} \begin{bmatrix} \mathbf{T}_{kj} & O_{N_{FP},N_{SP}} \\ O_{N_{FP},N_{SP}} & \mathbf{T}_{kj} \end{bmatrix}, \tag{D.5}$$

with $j \in [\![1, N_{SP}^{tri}]\!], k \in [\![1, N_{FP}^{tri}]\!]$ and $O_{m,n}$ is the zero matrix of size $m \times n$. The transfer matrix is given by Eq. (2.90):

$$\mathbf{T}_{kj} = \sum_{m=1}^{N_{SP}^{tri}} \left(\Phi_m(\boldsymbol{\xi}_j)\right)^{-1} \Phi_m(\boldsymbol{\xi}_k), \tag{D.6}$$

and the differentiation matrix by Eq. (2.101):

$$\mathbf{D}_{jk} = \sum_{n=1}^{N_{FP}^{tri}} \left(\boldsymbol{\phi}_n(\boldsymbol{\xi}_k) \cdot \hat{\mathbf{n}}_k\right)^{-1} \hat{\nabla} \cdot \boldsymbol{\phi}_n(\boldsymbol{\xi}_j). \tag{D.7}$$

The velocity matrices $\mathbf{C}^{0,0}, \mathbf{C}^{-1,0}, \mathbf{C}^{+1,0}, \mathbf{C}^{0,-1}$ and $\mathbf{C}^{0,+1}$ are given by:

$$\mathbf{C}^{0,0} = \begin{bmatrix} \begin{bmatrix} \mathbf{C}^L & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & \mathbf{C}^L & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & \mathbf{C}^L & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & \mathbf{C}^I_{N_i} \end{bmatrix} & \begin{bmatrix} O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & \mathbf{C}^R & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_i} \end{bmatrix} \\ \begin{bmatrix} O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & \mathbf{C}^R & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_i} \end{bmatrix} & \begin{bmatrix} \mathbf{C}^L & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & \mathbf{C}^L & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & \mathbf{C}^L & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & \mathbf{C}^I_{N_i} \end{bmatrix} \end{bmatrix}, \tag{D.8}$$

$$\mathbf{C}^{-1,0} = \begin{bmatrix} \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} & \begin{bmatrix} O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & \mathbf{C}^R & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_i} \end{bmatrix} \\ \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} & \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} \end{bmatrix}, \tag{D.9}$$

$$\mathbf{C}^{0,+1} = \begin{bmatrix} \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} & \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} \\ \begin{bmatrix} O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & \mathbf{C}^R & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_i} \end{bmatrix} & \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} \end{bmatrix}, \tag{D.10}$$

$$\mathbf{C}^{0,-1} = \begin{bmatrix} \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} & \begin{bmatrix} \mathbf{C}^R & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_i} \end{bmatrix} \\ \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} & \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} \end{bmatrix}, \tag{D.11}$$

$$\mathbf{C}^{+1,0} = \begin{bmatrix} \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} & \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} \\ \begin{bmatrix} \mathbf{C}^R & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_i} \end{bmatrix} & \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} \end{bmatrix}, \tag{D.12}$$

where $\mathbf{C}^I$, $\mathbf{C}^L$ and $\mathbf{C}^R$ are defined by:

$$\mathbf{C}^I = \left[\mathrm{diag}(|J|J^{-1}(\mathbf{c}\cdot\hat{\mathbf{n}})) \right]_{N_i,N_i}, \tag{D.13}$$

$$\mathbf{C}^L = (\mathbf{c}\cdot\mathbf{n}) \left[\mathrm{diag}(\frac{1+\mathrm{sign}(\mathbf{c}\cdot\mathbf{n})}{2}) \right]_{N_e,N_e}, \tag{D.14}$$

and

$$\mathbf{C}^R = (\mathbf{c}\cdot\mathbf{n}) \begin{bmatrix} 0 & \cdots & \frac{1-\mathrm{sign}(\mathbf{c}\cdot\mathbf{n})}{2} \\ \vdots & \iddots & \vdots \\ \frac{1-\mathrm{sign}(\mathbf{c}\cdot\mathbf{n})}{2} & \cdots & 0 \end{bmatrix}_{N_e,N_e}. \tag{D.15}$$

D.2. Tetrahedral Elements

The matrices $\mathbf{M}^{0,0,0}$, $\mathbf{M}^{-1,0,0}$, $\mathbf{M}^{+1,0,0}$, $\mathbf{M}^{0,-1,0}$, $\mathbf{M}^{0,+1,0}$, $\mathbf{M}^{0,0,-1}$ and $\mathbf{M}^{0,0,+1}$ involved in the SDRT spatial discretization for the Fourier analysis on tetrahedral elements (Eq. (3.36)) are detailed in this appendix. Those matrices are given as:

$$\mathbf{M}^{0,0,0} = \begin{bmatrix} \mathbf{D}_{jk} & \cdots & O_{N_{SP},N_{FP}} \\ \vdots & \ddots & \vdots \\ O_{N_{SP},N_{FP}} & \cdots & \mathbf{D}_{jk} \end{bmatrix} \mathbf{C}^{0,0,0} \begin{bmatrix} \mathbf{T}_{kj} & \cdots & O_{N_{FP},N_{SP}} \\ \vdots & \ddots & \vdots \\ O_{N_{FP},N_{SP}} & \cdots & \mathbf{T}_{kj} \end{bmatrix}, \tag{D.16}$$

where $O_{m,n}$ is the zero matrix of size $m \times n$. The same goes for $\mathbf{M}^{-1,0,0}$, $\mathbf{M}^{+1,0,0}$, $\mathbf{M}^{0,-1,0}$, $\mathbf{M}^{0,+1,0}$, $\mathbf{M}^{0,0,-1}$ and $\mathbf{M}^{0,0,+1}$, associated respectively to the velocity matrices $\mathbf{C}^{-1,0,0}$, $\mathbf{C}^{+1,0,0}$, $\mathbf{C}^{0,-1,0}$, $\mathbf{C}^{0,+1,0}$, $\mathbf{C}^{0,0,-1}$ and $\mathbf{C}^{0,0,+1}$. The transfer matrix is given by Eq. (2.90):

$$\mathbf{T}_{kj} = \sum_{m=1}^{N_{SP}^{tet}} \left(\Phi_m(\boldsymbol{\xi}_j)\right)^{-1} \; \Phi_m(\boldsymbol{\xi}_k), \tag{D.17}$$

and the differentiation matrix by Eq. (2.101):

$$\mathbf{D}_{jk} = \sum_{n=1}^{N_{FP}^{tet}} \left(\boldsymbol{\phi}_n(\boldsymbol{\xi}_k) \cdot \hat{\mathbf{n}}_k\right)^{-1} \; \hat{\nabla} \cdot \boldsymbol{\phi}_n(\boldsymbol{\xi}_j). \tag{D.18}$$

The velocity matrices are given as:

$$\mathbf{C}^{0,0,0} = \begin{bmatrix} \mathbf{C}^{L} & \mathbf{C}^{T_1,T_2} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & \mathbf{C}^{T_1,T_6} \\ \mathbf{C}^{T_2,T_1} & \mathbf{C}^{L} & \mathbf{C}^{T_2,T_3} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} \\ O_{N_{FP},N_{FP}} & \mathbf{C}^{T_3,T_2} & \mathbf{C}^{L} & \mathbf{C}^{T_3,T_4} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} \\ O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & \mathbf{C}^{T_4,T_3} & \mathbf{C}^{L} & \mathbf{C}^{T_4,T_5} & O_{N_{FP},N_{FP}} \\ O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & \mathbf{C}^{T_5,T_4} & \mathbf{C}^{L} & \mathbf{C}^{T_5,T_6} \\ \mathbf{C}^{T_6,T_1} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & \mathbf{C}^{T_6,T_5} & \mathbf{C}^{L} \end{bmatrix}, \tag{D.19}$$

$$\mathbf{C}^{-1,0,0} = \begin{bmatrix} O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & \mathbf{C}^{T_1,T_5} & O_{N_{FP},N_{FP}} \\ O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & \mathbf{C}^{T_2,T_4} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} \\ & & \begin{bmatrix} O_{4N_{FP},6N_{FP}} \end{bmatrix} & & & \end{bmatrix}, \tag{D.20}$$

$$\mathbf{C}^{+1,0,0} = \begin{bmatrix} & & \begin{bmatrix} O_{3N_{FP},6N_{FP}} \end{bmatrix} & & & \\ O_{N_{FP},N_{FP}} & \mathbf{C}^{T_4,T_2} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} \\ \mathbf{C}^{T_5,T_1} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} \\ & & \begin{bmatrix} O_{N_{FP},6N_{FP}} \end{bmatrix} & & & \end{bmatrix}, \tag{D.21}$$

$$\mathbf{C}^{0,-1,0} = \begin{bmatrix} & & \begin{bmatrix} O_{2N_{FP},6N_{FP}} \end{bmatrix} & & & \\ \mathbf{C}^{T_3,T_1} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} \\ O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & \mathbf{C}^{T_4,T_6} \\ & & \begin{bmatrix} O_{2N_{FP},6N_{FP}} \end{bmatrix} & & & \end{bmatrix}, \tag{D.22}$$

$$\mathbf{C}^{0,+1,0} = \begin{bmatrix} O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & \mathbf{C}^{T_1,T_3} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} \\ & & \begin{bmatrix} O_{4N_{FP},6N_{FP}} \end{bmatrix} & & & \\ O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & \mathbf{C}^{T_6,T_4} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} \end{bmatrix}, \tag{D.23}$$

$$\mathbf{C}^{0,0,-1} = \begin{bmatrix} & & \begin{bmatrix} O_{N_{FP},6N_{FP}} \end{bmatrix} & & & \\ O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & \mathbf{C}^{T_2,T_6} \\ O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & \mathbf{C}^{T_3,T_5} & O_{N_{FP},N_{FP}} \\ & & \begin{bmatrix} O_{3N_{FP},6N_{FP}} \end{bmatrix} & & & \end{bmatrix}, \tag{D.24}$$

$$\mathbf{C}^{0,0,+1} = \begin{bmatrix} & & \begin{bmatrix} O_{4N_{FP},6N_{FP}} \end{bmatrix} & & & \\ O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & \mathbf{C}^{T_5,T_3} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} \\ O_{N_{FP},N_{FP}} & \mathbf{C}^{T_6,T_2} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} & O_{N_{FP},N_{FP}} \end{bmatrix}, \quad \text{(D.25)}$$

where $\mathbf{C}^L$ is defined by:

$$\mathbf{C}^L = \begin{bmatrix} \begin{bmatrix} (\mathbf{c}\cdot\mathbf{n})\dfrac{1+\text{sign}(\mathbf{c}\cdot\mathbf{n})}{2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & (\mathbf{c}\cdot\mathbf{n})\dfrac{1+\text{sign}(\mathbf{c}\cdot\mathbf{n})}{2} \end{bmatrix}_{4N_f,4N_f} & & O_{4N_f,N_i} \\ & O_{N_i,4N_f} & \begin{bmatrix} |J|J^{-1}(\mathbf{c}\cdot\hat{\mathbf{n}}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & |J|J^{-1}(\mathbf{c}\cdot\hat{\mathbf{n}}) \end{bmatrix}_{N_i,N_i} \end{bmatrix}. \quad \text{(D.26)}$$

The matrix $\mathbf{C}^{T_i,T_j}$ links the FP between the triangular faces of T_i and T_j. Its expression will depend on the local connectivity, i.e. the number and the orientation of the two faces in their respective element. The face number gives the local FP numbering whereas the orientation (how the two faces are facing each other) gives the FP order. An example of its determination is provided below for two arbitrary tetrahedral elements.

Determination of the Matrix $\mathbf{C^{T_i,T_j}}$ - Example for $\mathbf{p=1}$ Let us consider two tetrahedron T_1 and T_2 defined by their four nodes:

$$T_1: \quad N_1, N_2, N_3, N_4 = A, B, C, D, \quad \text{(D.27)}$$

$$T_2: \quad N_1, N_2, N_3, N_4 = A, C, E, D. \quad \text{(D.28)}$$

Following the CGNS notations, their faces are defined by:

- T_1, Face 1: A, C, B,
- T_1, Face 2: A, B, D,
- T_1, Face 3: B, C, D,
- T_1, Face 4: C, A, D,
- T_2, Face 1: A, E, C,
- T_2, Face 2: A, C, D,
- T_2, Face 3: C, E, D,
- T_2, Face 4: E, A, D.

They are sharing a face corresponding to Face 4 in T_1 and Face 2 in T_2. In the case of $p = 1$, this indicates that the FP numbers on (T_1, Face 4) are $[10, 12]$ whereas the FP number (Face 2, T_2) are $[4, 6]$. The orientation between the faces then needs to be determined in order to know in which order the FP are facing each other. For two arbitrary faces A and B defined respectively by nodes (A_1, A_2, A_3) and (B_1, B_2, B_3), three cases are possible:

- $A_1 = B_3$, $A_2 = B_2$, $A_3 = B_1$,
- $A_1 = B_2$, $A_2 = B_1$, $A_3 = B_3$,
- $A_1 = B_1$, $A_2 = B_3$, $A_3 = B_2$.

Our example is illustrated in Fig. D.1 and corresponds to the second case.

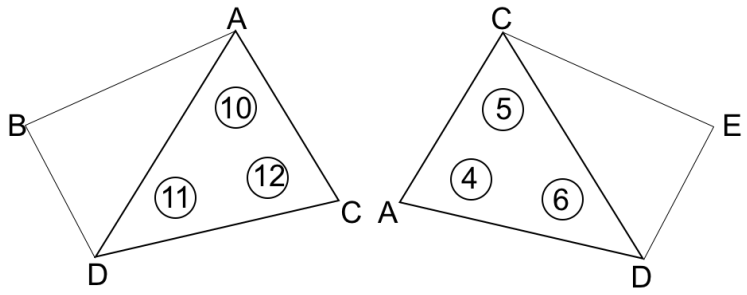


Figure D.1. – Illustration of the orientation determination: T_1 (on the left) and T_2 (on the right)

In this example, the matrix $\mathbf{C}^{T_1,T_2}$ will take the following expression:

$$\mathbf{C}^{T_1,T_2} = (\mathbf{c}\cdot\mathbf{n}) \begin{bmatrix} \begin{bmatrix} O_{N_f,N_f} & \begin{bmatrix} 0 & \frac{1+\mathrm{sign}(\mathbf{c}\cdot\mathbf{n})}{2} & 0 \\ 0 & 0 & \frac{1+\mathrm{sign}(\mathbf{c}\cdot\mathbf{n})}{2} \\ \frac{1+\mathrm{sign}(\mathbf{c}\cdot\mathbf{n})}{2} & 0 & 0 \end{bmatrix} & \begin{matrix} O_{3N_f,N_{FP}} \\ \\ \\ O_{N_i,N_{FP}} \end{matrix} \end{bmatrix} & O_{N_f,2*N_f+N_i} \end{bmatrix}. \tag{D.29}$$

Appendix E

Interior Flux Points Parametrization for the Optimization Algorithm

E.1. Triangular Element

Flux Point	(ξ, η)
$\boldsymbol{\xi}_{16}$	$(\beta_4/2 + \gamma_4, \beta_4/2 - \gamma_4)$
$\boldsymbol{\xi}_{17}$	$(\beta_1/2 + \gamma_1, \beta_1/2 - \gamma_1)$
$\boldsymbol{\xi}_{18}$	$(\beta_1/2 - \gamma_1, \beta_1/2 + \gamma_1)$
$\boldsymbol{\xi}_{19}$	(α_2, α_2)
$\boldsymbol{\xi}_{20}$	$(\beta_3/2 - \gamma_3, \beta_3/2 + \gamma_3)$
$\boldsymbol{\xi}_{21}$	$(\beta_2/2 + \gamma_2, \beta_2/2 - \gamma_2)$
$\boldsymbol{\xi}_{22}$	$(\beta_4/2 - \gamma_4, \beta_4/2 + \gamma_4)$
$\boldsymbol{\xi}_{23}$	$(\beta_3/2 + \gamma_3, \beta_3/2 - \gamma_3)$
$\boldsymbol{\xi}_{24}$	(α_1, α_1)
$\boldsymbol{\xi}_{25}$	$(\beta_2/2 - \gamma_2, \beta_2/2 + \gamma_2)$
$\boldsymbol{\xi}_{26:35}$	$\boldsymbol{\xi}_{16:25}$

Table E.1. – Interior FP coordinate parameters in a triangular element for SDRT_4

Flux Point	(ξ, η)	Flux Point	(ξ, η)
$\boldsymbol{\xi}_{19}$	(α_1, α_1)	$\boldsymbol{\xi}_{27}$	$(\beta_3/2 - \gamma_3, \beta_3/2 + \gamma_3)$
$\boldsymbol{\xi}_{20}$	$(\beta_6/2 - \gamma_6, \beta_6/2 + \gamma_6)$	$\boldsymbol{\xi}_{28}$	$(\beta_2/2 + \gamma_2, \beta_2/2 - \gamma_2)$
$\boldsymbol{\xi}_{21}$	$(\beta_6/2 + \gamma_6, \beta_6/2 - \gamma_6)$	$\boldsymbol{\xi}_{29}$	$(\beta_3/2 + \gamma_3, \beta_3/2 - \gamma_3)$
$\boldsymbol{\xi}_{22}$	(α_3, α_3)	$\boldsymbol{\xi}_{30}$	$(\beta_5/2 + \gamma_5, \beta_5/2 - \gamma_5)$
$\boldsymbol{\xi}_{23}$	(α_2, α_2)	$\boldsymbol{\xi}_{31}$	$(\beta_4/2 + \gamma_4, \beta_4/2 - \gamma_4)$
$\boldsymbol{\xi}_{24}$	$(\beta_2/2 - \gamma_2, \beta_2/2 + \gamma_2)$	$\boldsymbol{\xi}_{32}$	$(\beta_1/2 - \gamma_1, \beta_1/2 + \gamma_1)$
$\boldsymbol{\xi}_{25}$	$(\beta_5/2 - \gamma_5, \beta_5/2 + \gamma_5)$	$\boldsymbol{\xi}_{33}$	$(\beta_4/2 - \gamma_4, \beta_4/2 + \gamma_4)$
$\boldsymbol{\xi}_{26}$	$(\beta_1/2 + \gamma_1, \beta_1/2 - \gamma_1)$	$\boldsymbol{\xi}_{34:48}$	$\boldsymbol{\xi}_{19:33}$

Table E.2. – Interior FP coordinate parameters in a triangular element for SDRT_5

Flux Point	(ξ, η)	Flux Point	(ξ, η)
$\boldsymbol{\xi}_{22}$	(α_1, α_1)	$\boldsymbol{\xi}_{33}$	$(\beta_5/2 + \gamma_5, \beta_5/2 - \gamma_5)$
$\boldsymbol{\xi}_{23}$	(α_2, α_2)	$\boldsymbol{\xi}_{34}$	$(\beta_5/2 - \gamma_5, \beta_5/2 + \gamma_5)$
$\boldsymbol{\xi}_{24}$	(α_3, α_3)	$\boldsymbol{\xi}_{35}$	$(\beta_6/2 + \gamma_6, \beta_6/2 - \gamma_6)$
$\boldsymbol{\xi}_{25}$	$(\beta_1/2 + \gamma_1, \beta_1/2 - \gamma_1)$	$\boldsymbol{\xi}_{36}$	$(\beta_6/2 - \gamma_6, \beta_6/2 + \gamma_6)$
$\boldsymbol{\xi}_{26}$	$(\beta_1/2 - \gamma_1, \beta_1/2 + \gamma_1)$	$\boldsymbol{\xi}_{37}$	$(\beta_7/2 + \gamma_7, \beta_7/2 - \gamma_7)$
$\boldsymbol{\xi}_{27}$	$(\beta_2/2 + \gamma_2, \beta_2/2 - \gamma_2)$	$\boldsymbol{\xi}_{38}$	$(\beta_7/2 - \gamma_7, \beta_7/2 + \gamma_7)$
$\boldsymbol{\xi}_{28}$	$(\beta_2/2 - \gamma_2, \beta_2/2 + \gamma_2)$	$\boldsymbol{\xi}_{39}$	$(\beta_8/2 + \gamma_8, \beta_8/2 - \gamma_8)$
$\boldsymbol{\xi}_{29}$	$(\beta_3/2 + \gamma_3, \beta_3/2 - \gamma_3)$	$\boldsymbol{\xi}_{40}$	$(\beta_8/2 - \gamma_8, \beta_8/2 + \gamma_8)$
$\boldsymbol{\xi}_{30}$	$(\beta_3/2 - \gamma_3, \beta_3/2 + \gamma_3)$	$\boldsymbol{\xi}_{41}$	$(\beta_9/2 + \gamma_9, \beta_9/2 - \gamma_9)$
$\boldsymbol{\xi}_{31}$	$(\beta_4/2 + \gamma_4, \beta_4/2 - \gamma_4)$	$\boldsymbol{\xi}_{42}$	$(\beta_9/2 - \gamma_9, \beta_9/2 + \gamma_9)$
$\boldsymbol{\xi}_{32}$	$(\beta_4/2 - \gamma_4, \beta_4/2 + \gamma_4)$	$\boldsymbol{\xi}_{43:63}$	$\boldsymbol{\xi}_{22:42}$

Table E.3. – Interior FP coordinate parameters in a triangular element for SDRT_6

E.2. Tetrahedral Element

Flux Point	(ξ, η, ζ)
$\boldsymbol{\xi}_{41}$	(α, β, β)
$\boldsymbol{\xi}_{42}$	(β, α, β)
$\boldsymbol{\xi}_{43}$	(β, β, α)
$\boldsymbol{\xi}_{44}$	(β, β, β)
$\boldsymbol{\xi}_{45}$	(γ, γ, δ)
$\boldsymbol{\xi}_{46}$	(γ, δ, γ)
$\boldsymbol{\xi}_{47}$	(γ, δ, δ)
$\boldsymbol{\xi}_{48}$	(δ, γ, γ)
$\boldsymbol{\xi}_{49}$	(δ, γ, δ)
$\boldsymbol{\xi}_{50}$	(δ, δ, γ)
$\boldsymbol{\xi}_{51:60}$	$\boldsymbol{\xi}_{41:50}$
$\boldsymbol{\xi}_{61:70}$	$\boldsymbol{\xi}_{51:60}$

Table E.4. – Interior FP coordinate parameters in a tetrahedral element for SDRT_3

Appendix F

Coefficients for the Runge-Kutta Time Discretization

Scheme	α_{lk}				β_{lk}			
SSP2s2o [186]	1				1			
	1/2	1/2			0	1/2		
SSP3s2o [189]	1				1/2			
	0	1			0	1/2		
	1/3	0	2/3		0	0	1/3	
SSP4s2o [189]	1				1/3			
	0	1			0	1/3		
	0	0	1		0	0	1/3	
	1/4	0	0	3/4	0	0	0	1/4
SSP3s3o [186]	1				1			
	3/4	1/4			0	1/4		
	1/3	0	2/3		0	0	2/3	
SSP4s3o [189]	1				1/2			
	0	1			0	1/2		
	2/3	0	1/3		0	0	1/6	
	0	0	0	1	0	0	0	1/2
RK4J [190]	1				1/4			
	1	0			0	1/2		
	1	0	0		0	0	0.55	
	1	0	0	0	0	0	0	1

Table F.1. – Coefficients (α, β) for the two-, three- and four-stage SSP schemes

		SSP5s3o [189]			
	1				
	0	1			
α_{lk}	0.56656131914033	0	0.43343868085967		
	0.09299483444413	0.00002090369620	0	0.90698426185967	
	0.00736132260920	0.20127980325145	0.00182955389682	0	0.78952932024253
	0.37726891511710				
	0	0.37726891511710			
β_{lk}	0	0	0.16352294089771		
	0.00071997378654	0	0	0.34217696850008	
	0.00277719819460	0.00001567934613	0	0	0.29786487010104
		SSP5s4o [189]			
	1				
	0.44437049406734	0.55562950593266			
α_{lk}	0.62010185138540	0	0.37989814861460		
	0.17807995410773	0	0	0.82192004589227	
	0.00683325884039	0	0.51723167208978	0.12759831133288	0.34833675773694
	0.39175222700392				
	0	0.36841059262959			
β_{lk}	0	0	0.25189177424738		
	0	0	0	0.54497475021237	
	0	0	0	0.08460416338212	0.22600748319395

Table F.2. – Coefficients (α, β) for the five-stage SSP schemes

	RKs4s	RKo5s	RKo6s
γ_1	1	1	1
γ_2	1/2	1/2	1/2
γ_3	1/6	0.165250353664	0.165919771368
γ_4	1/24	0.039372585984	0.040919732041
γ_5		0.007149096448	0.007555704391
γ_6			0.000891421261

Table F.3. – Coefficients γ for the standard four-stage RK scheme and the optimized five- and six-stage RK schemes

List of Figures

1.1. Representation of dissipation and dispersion effects for the linear advection of a planar wave in a periodic domain . . . 3
1.2. Simplified view of the Kolmogorov spectrum and illustration of CFD methods for turbulence . . . 5

2.1. Points distribution in the hexahedral reference element for $p = 1$: SP $\boldsymbol{\xi}_{(j_1,j_2,j_3)}$ (▲), FP in the ξ-direction $\boldsymbol{\xi}_{(k_1,j_2,j_3)}$ (●), FP in the η-direction $\boldsymbol{\xi}_{(j_1,k_2,j_3)}$ (■), FP in the ζ-direction $\boldsymbol{\xi}_{(j_1,j_2,k_3)}$ (◆) . . . 22
2.2. Points distribution and associated DoF in the triangular and tetrahedal reference elements: SP (▲), interior FP (■), edge FP (●) . . . 32
2.3. Points distribution in the prismatic reference element for $p = 1$: SP $\boldsymbol{\xi}_{(j_1,j_2)}$ (▲), FP in the (ξ, η) plane $\boldsymbol{\xi}_{(k_1,j_2)}$ (edge: ●, interior: ■), FP in the ζ-direction $\boldsymbol{\xi}_{(j_1,k_2)}$ (◆) . . . 38

3.1. Computational domain for the 1D FNC linear stability analysis - Example of SP (▲) and FP (■) distributions for $p = 3$, $N = 3$ . . . 47
3.2. Spectrum of the matrix $\mathbf{M}_{\mathrm{n}}$ for the standard 1D SD scheme, $p = 4$: Chebyshev-Gauss-Lobatto FP (+), and Legendre FP(○) . . . 48
3.3. FP numbering in the reference triangular element - Example of FP distribution for $p = 2$ (edge: ●, interior: ■) . . . 49
3.4. Mesh generating pattern used for the 2D Fourier stability analysis on triangles . . . 50
3.5. Fourier footprint of the SDRT_4 and SDRT_5 spatial discretizations on triangles for $\theta = 0$ using different interior FP locations . . . 52
3.6. Computational domain for the FNC linear stability analysis on triangles - Example of FP distribution for $p = 2$, $N = 2$ (edge: ●, interior: ■) . . . 53
3.7. Interior FP set up in the reference triangle for SDRT_3 . . . 55
3.8. Sets of FP determined using FNC analysis in the reference triangle . . . 55
3.9. Spectra of matrix $\mathbf{M}_{\mathrm{n}}$ obtained using the FNC method for SDRT_4 and SDRT_5 schemes 56
3.10. Computational domain for the FNC linear stability analysis on triangles - Example for $N = 5$ . . . 57
3.11. Maximum eigenvalue real part of $\mathbf{M}_{\mathrm{n}}$ determined using the FNC analysis for different domain size . . . 58
3.12. Points distribution in the tetrahedral reference element for $p = 1$: SP $\boldsymbol{\xi}_j$ (▲) and FP $\boldsymbol{\xi}_k$ (face (●) and interior (■)) . . . 60
3.13. Computational domain for the Fourier stability analysis on tetrahedral elements . . . 62
3.14. Spectrum of matrix $\mathbf{M}_{\mathrm{z}}$ for stable SDRT schemes on tetrahedral elements, $(\theta_1, \theta_2) \in (0, \pi/8, \pi/4)^2$ . . . 64
3.15. Spectrum of matrix $\mathbf{M}_{\mathrm{z}}$ for unstable SDRT_2 schemes on tetrahedral elements, $(\theta_1, \theta_2) = (0, 0)$ . . . 64

4.1. Stability behavior of the coupled time-space discretizations for spatially stable SDRT schemes on triangles using SSP3s3o and SSP5s4o temporal schemes . . . 70
4.2. Stability behavior of the coupled time-space discretizations for spatially stable SDRT schemes on triangles using second-order temporal schemes . . . 71
4.3. Solution initialization for the sine advection . . . 71

4.4. Evolution of max$|u|$ using SDRT$_2$ on triangular mesh, $\theta = \pi/8$, $\nu = 0.1$ 71
4.5. Stability behavior of the coupled time-space discretizations for SDRT$_4$ on triangles for $\theta = \pi/8$ 74
4.6. Stability behavior of the coupled time-space discretizations for SDRT$_5$ on triangles for $\theta = \pi/8$ 74
4.7. Evolution of max$|u|$ using SDRT$_5$ on triangles for $\theta = \pi/8$ 76
4.8. Stability behavior of the coupled time-space discretizations for SDRT$_4$ on triangles for $\theta \in [0, 2\pi]$ 76
4.9. Stability behavior of the coupled time-space discretizations for SDRT$_5$ on triangles for $\theta \in [0, 2\pi]$ 77
4.10. Stability study of SDRT$_5$ coupled with SSP3s3o using $\theta = \vartheta = 0$ on triangular elements 78
4.11. Study of SDRT$_5$ on triangles coupled with different temporal schemes using $\theta = \vartheta = 0$, $\nu = 0.05$ 78
4.12. Stability behavior of the coupled time-space discretizations for SDRT$_2$ on tetrahedron 80

5.1. Sets of FP determined using the optimization process on Fourier analysis compared to Williams-Shunn-Jameson sets 87
5.2. Fourier footprint of the SDRT$_4$ (left) and SDRT$_5$ (right) spatial discretizations on triangles 89
5.3. Stability behavior of the coupled time-space discretizations for SDRT$^{\text{OPT}}$ schemes on triangles 90
5.4. Fourier footprint of the SDRT$_6$ spatial discretizations on triangles 92
5.5. Fourier footprint of the SDRT$_3$ spatial discretizations on tetrahedral elements, $(\theta_1, \theta_2, \vartheta_1, \vartheta_2) = (0, 0, 0, 0)$ 95

6.1. Regular triangular grid used for the SDRT implementation comparison 98
6.2. Assessment of the RK4J and SSP3s3o accuracy for the SDRT$^{\text{WSJ}}$ implementation - Cross-section of the solution u at $y = 0$ 99
6.3. Assessment of the SDRT$^{\text{OPT}}$ implementation accuracy - Cross-section of the solution u at $y = 0$ 100
6.4. Initialization of the COVO test case using SDRT$_5$ and $N = 20$ on a triangular grid . . 103
6.5. Cross-section of ρV at $y = 0.05\,\text{m}$ after 50 periods using SDRT$_p$ schemes, $p \in [\![2, 5]\!]$, on triangular grids for $\sim 18,000$ DOF 103
6.6. ρV contours after 50 periods using SDRT$_p$ schemes, $p \in [\![2, 5]\!]$, on triangular grids for $\sim 18,000$ DOF 104
6.7. L_2 error and theoretical order of accuracy slopes for the COVO test case on triangular grids 105
6.8. Initialization of the COVO test case using SD/SDRT$_5$ and $N = 16$ on a hybrid grid . 106
6.9. Cross-section of ρV at $y = 0.05\,\text{m}$ after 5 periods using SD/SDRT$_p$ schemes, $p \in [\![2, 5]\!]$, on hybrid grids for $\sim 10,000$ DOF 106
6.10. ρV contours after 5 periods using SD/SDRT$_p$ schemes, $p \in [\![2, 5]\!]$, on hybrid grids for $\sim 10,000$ DOF 106
6.11. L_2 error and theoretical order of accuracy slopes for the COVO test case on hybrid grids after 5 periods 107
6.12. Initial density condition on a $10 \times 10 \times 10$ regular tetrahedral grid using $p = 2$ 108

7.1. Unstructured mesh around a NACA0012 airfoil - 2407 quadratic triangular elements . 114
7.2. Convergence of the residual for the transonic viscous flow over a NACA0012 airfoil . . 115
7.3. Case A: Mach number contours using SDRT$_p$ schemes, $p \in [\![2, 5]\!]$ 116
7.4. Case A: Surface skin-friction coefficient C_f 117
7.5. Case A: Surface pressure coefficient C_p 117
7.6. Case B: Mach number contours using SDRT$_p$ schemes, $p \in [\![2, 5]\!]$ 118

7.7. Case B: Surface skin-friction coefficient C_f . . . 119
7.8. Case B: Surface pressure coefficient C_p . . . 119
7.9. Case C: Mach number contours using SDRT$_p$ schemes, $p \in [\![2, 5]\!]$ . . . 120
7.10. Case C: Surface skin-friction coefficient C_f . . . 121
7.11. Case C: Surface pressure coefficient C_p . . . 121
7.12. Close view of the hybrid mesh for the viscous flow around a cylinder . . . 122
7.13. Convergence of the residual for the viscous flow around circular cylinder . . . 122
7.14. Normalized x-velocity contours and streamlines around the cylinder using a SD/SDRT$_5$ scheme . . . 123
7.15. Pressure coefficient C_p distribution on the cylinder surface . . . 123
7.16. Dimensionless negative vorticity $\bar{\omega}$ distribution on the cylinder surface . . . 123
7.17. Dimensionless kinetic energy dissipation rate obtained with second and third-order SDRT schemes compared to reference data . . . 127
7.18. Efficiency of the first SDRT implementation in JAGUAR compared with other solvers 128

A.1. Linear line reference element . . . 137
A.2. Quadratic line reference element . . . 137
A.3. Linear triangular reference element . . . 139
A.4. Quadratic triangular reference element . . . 139
A.5. Linear quadrilateral reference element . . . 141
A.6. Quadratic quadrilateral reference element . . . 141
A.7. 3D linear reference elements . . . 144

C.1. CFL stability limits for the 1D SD scheme determined using FNC analysis depending on the number of cells . . . 151

D.1. Illustration of the orientation determination: T_1 (on the left) and T_2 (on the right) . . 157

List of Tables

2.1. Definition of the RT_p space 29

3.1. Values of $\max(\mathrm{Re}(\lambda_{\mathbf{M}_z}))$ for $\theta = 0$ using different quadrature rules as the interior FP locations 52
3.2. Interior FP coordinate parameters for SDRT_3 55
3.3. Number of SP and physical interior FP for SDRT scheme on tetrahedral elements . . . 61

4.1. CFL stability limits for SDRT_2 and SDRT_3 on triangles coupled with different temporal schemes 69
4.2. CFL stability limits for SDRT_4 and SDRT_5 on triangles coupled with different temporal schemes, $\theta = \pi/8$ 75
4.3. CFL stability limits for SDRT_4 and SDRT_5 on triangles coupled with different temporal schemes, $\theta \in [0, 2\pi]$ 77
4.4. CFL stability limits for SDRT_1 and SDRT_2 on tetrahedron coupled with different temporal schemes 81

5.1. Coordinate parameters of interior FP determined using the optimization process on Fourier analysis 87
5.2. CFL stability limits for $\mathrm{SDRT}^{\mathrm{OPT}}$ schemes on triangles coupled with different temporal schemes 91
5.3. Coordinate parameters of interior FP determined using the optimization process on Fourier analysis for $\mathrm{SDRT}_6, \theta = 0$ 92
5.4. Impact of the FP located on faces: values of $\max(\mathrm{Re}(\lambda_{\mathbf{M}_z}))$ for SDRT_3, $(\theta_1, \theta_2, \vartheta_1, \vartheta_2) = (0, 0, 0, 0)$ 96

6.1. Assessment of the RK4J and SSP3s3o accuracy for the $\mathrm{SDRT}^{\mathrm{WSJ}}$ implementation - L_2 error 100
6.2. Assessment of the $\mathrm{SDRT}^{\mathrm{OPT}}$ implementation accuracy - L_2 error 101
6.3. Overall accuracy orders for the COVO test case on triangular grids 105
6.4. Overall accuracy orders for the COVO test case on hybrid grids after 5 periods 107
6.5. L_2 error and order of accuracy values for regular and irregular tetrahedral grids 109

7.1. Case A: Comparison of drag coefficients and separation point location 118
7.2. Case B: Comparison of drag coefficients and separation point location 119
7.3. Case C: Comparison of drag coefficients and separation point location 121
7.4. Comparison of drag coefficients and separation angle for flow over a cylinder 124
7.5. Flow conditions for the TGV test case 125
7.6. Computational conditions for the TGV test case 126

E.1. Interior FP coordinate parameters in a triangular element for SDRT_4 159
E.2. Interior FP coordinate parameters in a triangular element for SDRT_5 159
E.3. Interior FP coordinate parameters in a triangular element for SDRT_6 160
E.4. Interior FP coordinate parameters in a tetrahedral element for SDRT_3 161

F.1. Coefficients (α, β) for the two-, three- and four-stage SSP schemes 163

F.2. Coefficients (α, β) for the five-stage SSP schemes 164
F.3. Coefficients γ for the standard four-stage RK scheme and the optimized five- and six-stage RK schemes 164

List of Algorithms

1. Fourier analysis optimization algorithm on triangles for SDRT_4 86
2. Fourier analysis optimization algorithm on tetrahedral elements for SDRT_3 94

Bibliography

[1] Z.J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, and M. Visbal, "High-order CFD methods: current status and perspective," International Journal for Numerical Methods in Fluids, vol. 72, no. 8, pp. 811–845, 2013.

[2] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, "Uniformly high order accurate essentially non-oscillatory schemes, III," Journal of Computational Physics, vol. 71, no. 2, pp. 231–303, 1987.

[3] R. Abgrall, "On Essentially Non-oscillatory Schemes on Unstructured Meshes: Analysis and Implementation," Journal of Computational Physics, vol. 144, pp. 45–58, 1994.

[4] C. Ollivier-Gooch, "Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction," Journal of Computational Physics, vol. 133, no. 1, p. 6–17, 1997.

[5] X.-D. Liu, S. Osher, and T. Chan, "Weighted Essentially Non-oscillatory Schemes," Journal of Computational Physics, vol. 115, no. 1, pp. 200–212, 1994.

[6] O. Friedrich, "Weighted Essentially Non-Oscillatory schemes for the interpolation of mean values on unstructured grids," Journal of Computational Physics, vol. 144, no. 1, p. 194–212, 1998.

[7] C. Hu and C.-W. Shu, "Weighted Essentially Non-oscillatory Schemes on Triangular Meshes," Journal of Computational Physics, vol. 1, no. 1, pp. 97–127, 1999.

[8] T. Barth and P. Frederickson, "Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction," in 28th Aerospace Sciences Meeting, American Institute of Aeronautics and Astronautics, 1990.

[9] A. Nejat and C. Ollivier-Gooch, "A high-order accurate unstructured Finite Volume Newton-Krylov algorithm for inviscid compressible flows," Journal of Computational Physics, vol. 227, no. 4, p. 2582–2609, 2008.

[10] C. Ollivier-Gooch and M. Van Altena, "A High-Order-Accurate Unstructured Mesh Finite-Volume Scheme for the Advection-Diffusion Equation," Journal of Computational Physics, vol. 181, no. 2, pp. 729–752, 2002.

[11] M. Delanaye and Y. Liu, "Quadratic reconstruction finite volume schemes on 3D arbitrary unstructured polyhedral grids," in 14th Computational Fluid Dynamics Conference, Norfolk, Virginia, USA (AIAA99-3259), 1999.

[12] D. R. Chapman, "Computational Aerodynamics Development and Outlook," AIAA Journal, vol. 17, no. 12, pp. 1293–1313, 1979.

[13] A. N. Kolmogorov, "The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers," Dokl. Akad. Nauk SSSR, vol. 30, no. 299, 1941.

[14] L. Zhou, Modeling of Single-Phase Turbulence, ch. 5, pp. 89–120. 2018.

[15] J. Boussinesq, Théorie de l'ecoulement tourbillonant et tumultueux des liquides dans les lits rectilignes à grande section, ch. I, II. 1897.

[16] B. E. Launder and B. I. Sharma, "Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc," Letters in Heat and Mass Transfer, vol. 1, no. 2, pp. 131–137, 1974.

[17] K.-Y. Chien, "Predictions of channel and boundary-layer flows with a low-Reynolds number turbulence model," AIAA Journal, vol. 20, no. 1, pp. 33–38, 1982.

[18] B. Mohammadi, "Complex turbulent flows computations with a two-layer approach," International Journal for Numerical Methods in Fluids, vol. 15, no. 747–771, 1992.

[19] P. Spalart and S. Allmaras, "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Journal, vol. 439, 1992.

[20] P. Y. Chou, "On velocity correlations and the solutions of the equations of turbulent fluctuation," Quarterly of Applied Mathematics, vol. 3, no. 1, pp. 38–54, 1945.

[21] S. B. Pope, "Turbulent Flows," Journal of Turbulence, vol. 1, no. 771, pp. 1468–5248, 2000.

[22] S. Ghosal, T. Lund, P. Moin, and K. Akselvoll, "A dynamic localization model for large-eddy simulation of turbulent flows," Journal of Fluid Mechanics, vol. 286, pp. 229–255, 1995.

[23] J. Smagorinsky, "General circulation experiments with the primitive equations, I. the basic experiment," Monthly Weather Review, vol. 91, no. 3, p. 99–164, 1963.

[24] D. Lilly, "A proposed modification of the Germano subgrid-scale closure method," Physics of Fluids A: Fluid Dynamics, vol. 4, pp. 633–635, 1992.

[25] O. Metais and M. Lesieur, "Spectral large-eddy simulation of isotropic and stably stratified turbulence," Journal of Fluid Mechanics, vol. 239, pp. 157–194, 1992.

[26] F. Nicoud and F. Ducros, "Subgrid-scale stress modelling based on the square of the velocity gradient tensor," Flow, Turbulence and Combustion, vol. 62, no. 3, pp. 183–200, 1999.

[27] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot, "A dynamic subgrid-scale eddy viscosity model," Physics of Fluids, vol. 3, no. 7, pp. 760–1765, 1991.

[28] H. Pitsch, "Large-Eddy Simulation of Turbulent Combustion," Annual Review of Fluid Mechanics, vol. 38, pp. 453–482, 2006.

[29] G. Boudier, L. Gicquel, T. Poinsot, D. Bissières, and C. Bérat, "Comparison of LES, RANS and experiments in an aeronautical gas turbine combustion chamber," Proceedings of the Combustion Institute, vol. 31, pp. 3075–3082, 2007.

[30] S. Som, P. Sénécal, and E. Pomraning, "Comparison of RANS and LES Turbulence Models against Constant Volume Diesel Experiments," 2012.

[31] Y. Xiaofeng, G. Saurabh, K. Tang-Wei, and G. Venkatesh, "RANS and Large Eddy Simulation of Internal Combustion Engine Flows—A Comparative Study," Journal of Engineering for Gas Turbines and Power, vol. 136, no. 5, 2014.

[32] P. Warzecha and A. Boguslawski, "LES and RANS modeling of pulverized coal combustion in swirl burner for air and oxy-combustion technologies," Energy, vol. 66, pp. 732–743, 2014.

[33] B. Cockburn, G. E. Karniadakis, and C.-W. Shu, eds., Discontinuous Galerkin Methods. Springer Berlin Heidelberg, 2000.

[34] J. S. Hesthaven and T. Warburton, Nodal Discontinuous Galerkin Methods. Springer New York, 2008.

[35] D. Di Pietro and A. Ern, Mathematical Aspects of Discontinuous Galerkin Methods. Springer Berlin Heidelberg, 2012.

[36] X. Feng, O. Karakashian, and Y. Xing, eds., Recent Developments in Discontinuous Galerkin Finite Element Methods for Partial Differential Equations. Springer International Publishing, 2014.

[37] V. Dolejší and M. Feistauer, Discontinuous Galerkin Method. Springer International Publishing, 2015.

[38] M. Uzunca, Adaptive Discontinuous Galerkin Methods for Non-linear Reactive Flows. Springer International Publishing, 2016.

[39] A. Cangiani, Z. Dong, E. H. Georgoulis, and P. Houston, hp-Version Discontinuous Galerkin Methods on Polygonal and Polyhedral Meshes. Springer International Publishing, 2017.

[40] S. Du and F.-J. Sayas, An Invitation to the Theory of the Hybridizable Discontinuous Galerkin Method. Springer International Publishing, 2019.

[41] N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. Ven, and K. Sørensen, eds., ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications, vol. 13 of Notes on Numerical Fluid Mechanics and Multidisciplinary Design. Springer Berlin Heidelberg, 2010.

[42] 36th CFD/ADIGMA course on hp-adaptive and hp-multigrid methods. Von Karman Institute, VKI LS 2010-01, 2010.

[43] N. Kroll, C. Hirsch, F. Bassi, C. Johnston, and K. Hillewaert, eds., IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach, Results of a Collaborative Project Funded by the European Union, vol. 128 of Notes on Numerical Fluid Mechanics and Multidisciplinary Design. Springer, 2015.

[44] V. Brunet, E. Croner, A. Minot, J. de Laborderie, E. Lippinois, S. Richard, J.-F. Boussuge, J. Dombard, F. Duchaine, L. Gicquel, T. Poinsot, G. Puigt, G. Staffelbach, L. Segui, O. Vermorel, N. Villedieu, J.-S. Cagnone, K. Hillewaert, M. Rasquin, G. Lartigue, V. Moureau, T. Roger, V. Couaillier, E. Martin, M. de la Llave Plata, J.-M. Le Gouez, and F. Renac, "Comparison of various CFD codes for LES simulations of turbomachinery: from inviscid vortex convection to multi-stage compressor," in ASME Turbo Expo 2018: Turbomahinery Technical Conference and Exposition, (Oslo, Norway), 2018.

[45] W. H. Reed and T. R. Hill, "Triangular Mesh Methods for the Neutron Transport Equation," Los Alamos Report LA-UR-73-479, 1973.

[46] B. Cockburn and C.-W. Shu, "TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws II: General Framework," Mathematics of Computation, vol. 52, no. 186, pp. 411–435, 1989.

[47] B. Cockburn and S.-Y. Lin, "TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws III: One Dimensional Systems," Journal of Computational Physics, vol. 84, no. 1, pp. 90–113, 1989.

[48] B. Cockburn, S. Hou, and C.-W. Shu, "The Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws IV: The multidimensional case," Mathematics of Computation, vol. 54, pp. 545–581, 1990.

[49] B. Cockburn and C.-W. Shu, "The Runge-Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems," Journal of Computational Physics, vol. 141, no. 2, pp. 199–224, 1998.

[50] B. Cockburn and C. Shu, "Runge–Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems," Journal of Scientific Computing, vol. 16, pp. 173–261, 2001.

[51] F. Bassi, L. Botti, A. Colombo, D. D. Pietro, and P. Tesini, "On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations," Journal of Computational Physics, vol. 231, no. 1, pp. 45–65, 2012.

[52] J.-F. Remacle, J. Flaherty, and M. Shepard, "An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems," SIAM Review, vol. 45, pp. 53–72, 2003.

[53] G. J. Gassner, F. Lörcher, C.-D. Munz, and J. S.Hesthaven, "Polymorphic nodal elements and their application in discontinuous Galerkin methods," Journal of Computational Physics, vol. 228, no. 5, pp. 1573–1590, 2009.

[54] D. Wirasaet, S. Tanaka, E. Kubatko, J. Westerink, and C. Dawson, "A performance comparison of nodal discontinuous Galerkin methods on triangles and quadrilaterals," International Journal for Numerical Methods in Fluids, vol. 64, pp. 1336–1362, 2010.

[55] G. J. Gassner, A. R. Winters, and D. A. Kopriva, "Split form nodal discontinuous Galerkin schemes with summation-by-parts property for the compressible Euler equations," Journal of Computational Physics, vol. 327, pp. 39–66, 2016.

[56] T. Chen and C.-W. Shu, "Entropy stable high order discontinuous Galerkin methods with suitable quadrature rules for hyperbolic conservation laws," Journal of Computational Physics, vol. 345, pp. 427–461, 2017.

[57] D. A. Kopriva and G. Gassner, "On the quadrature and weak form choices in collocation type discontinuous Galerkin spectral element methods," J. Sci. Comput, vol. 44, pp. 136–155, 2010.

[58] X. Zhang and C.-W. Shu, "On positivity-preserving high order discontinuous Galerkin schemes for compressible euler equations on rectangular meshes," Journal of Computational Physics, vol. 229, no. 23, pp. 8918–8934, 2010.

[59] G. Jiang and C.-W. Shu, "On cell entropy inequality for discontinuous Galerkin methods," Mathematics of Computation, vol. 62, no. 206, pp. 531–538, 1994.

[60] W. Li, J. Pan, and Y.-X. Ren, "The discontinuous Galerkin spectral element methods for compressible flows on two-dimensional mixed grids," Journal of Computational Physics, vol. 364, pp. 314–346, 2018.

[61] F. Renac, "Entropy stable DGSEM for nonlinear hyperbolic systems in nonconservative form with application to two-phase flows," Journal of Computational Physics, vol. 382, pp. 1–26, 2019.

[62] Z.J. Wang, "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids: Basic Formulation," Journal of Computational Physics, vol. 178, no. 1, pp. 210–251, 2002.

[63] Z.J. Wang and Y. Liu, "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids," Journal of Computational Physics, vol. 179, no. 2, pp. 665–697, 2002.

[64] Z.J. Wang and Y. Liu, "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids III: One Dimensional Systems and Partition Optimization," Journal of Scientific Computing, vol. 20, no. 1, pp. 137–157, 2004.

[65] Z.J. Wang, L. Zhang, and Y. Liu, "Spectral (finite) Volume Method for Conservation Laws on Unstructured Grids IV: Extension to Two-dimensional Systems," Journal of Computational Physics, vol. 194, no. 2, pp. 716–741, 2004.

[66] Y. Liu, M. Vinokur, and Z.J. Wang, "Spectral (finite) Volume Method for Conservation Laws on Unstructured Grids V: Extension to Three-dimensional Systems," Journal of Computational Physics, vol. 212, no. 2, pp. 454–472, 2006.

[67] Y. Sun, Z.J. Wang, and Y. Liu, "Spectral (finite) Volume Method for Conservation Laws on Unstructured Grids VI: Extension to Viscous Flow," Journal of Computational Physics, vol. 215, no. 1, pp. 41–58, 2006.

[68] Z.J. Wang and Y. Liu, "Extension of the Spectral Volume Method to High-order Boundary Representation," Journal of Computational Physics, vol. 211, no. 1, pp. 154–178, 2006.

[69] C. Breviglieri, J. L. F. Azevedo, E. Basso, and M. A. F. Souza, "Implicit High-Order Spectral Finite Volume Method for Inviscid Compressible Flows," AIAA Journal, vol. 48, no. 10, pp. 2365–2376, 2010.

[70] C. Breviglieri, J. L. F. Azevedo, and E. Basso, "An Unstructured Grid Implementation of High-Order Spectral Finite Volume Schemes," J. Braz. Soc. Mech. Sci. & Eng., vol. 32, 2010.

[71] R. Harris, Z.J. Wang, and Y. Liu, "Efficient quadrature-free high-order spectral volume method on unstructured grids: Theory and 2D implementation," Journal of Computational Physics, vol. 227, no. 3, pp. 1620–1642, 2008.

[72] M. Parsani, K. Van den Abeele, C. Lacor, and E. Turkel, "Implicit LU-SGS algorithm for high-order methods on unstructured grid with p-multigrid strategy for solving the steady Navier-Stokes equations," Journal of Computational Physics, vol. 229, no. 3, pp. 828–850, 2010.

[73] R. Kannan and Z.J. Wang, "LDG2: A Variant of the LDG Flux Formulation for the Spectral Volume Method," Journal of Scientific Computing, vol. 46, no. 2, pp. 314–328, 2011.

[74] R. Raghavendra, "A high order spectral volume method for equations containing third spatial derivatives using an interior penalty formulation," CFD Letters, vol. 3, no. 2, pp. 74–88, 2011.

[75] M. Yang, R. Harris, Z.J. Wang, and Y. Liu, "Efficient Quadrature-Free 3D High-Order Spectral Volume Method on Unstructured Grids," in 18th AIAA Computational Fluid Dynamics Conference, 2007.

[76] K. Van den Abeele and C. Lacor, "An accuracy and stability study of the 2D spectral volume method," Journal of Computational Physics, vol. 226, no. 1, pp. 1007–1026, 2007.

[77] R. Harris and Z.J. Wang, "Partition Design and Optimization for High-Order Spectral Volume Schemes," in 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition, 2009.

[78] R. Kannan and Z. Wang, "Improving the High Order Spectral Volume Formulation Using a Diffusion Regulator," Communications in Computational Physics, vol. 12, no. 1, pp. 247–260, 2012.

[79] D. A. Kopriva and J. H. Kolias, "A Conservative and Staggered-Grid Chebyshev and Multidomain Method and for Compressible and Flows," Journal of Computational Physics, vol. 125, pp. 244–261, 1996.

[80] H. T. Huynh, "A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods," in 18th AIAA Computational Fluid Dynamics Conference, 2007.

[81] Z.J. Wang and H. Gao, "A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids," Journal of Computational Physics, vol. 228, no. 21, pp. 8161–8186, 2009.

[82] Y. Allaneau and A. Jameson, "Connections between the filtered discontinuous Galerkin method and the flux reconstruction approach to high order discretizations," Computer Methods in Applied Mechanics and Engineering, vol. 200, pp. 3626–3636, 2011.

[83] A. Jameson, "A Proof of the Stability of the Spectral Difference Method for All Orders of Accuracy," Journal of Scientific Computing, vol. 45, no. 1-3, pp. 348–358, 2010.

[84] A. Jameson, P. Vincent, and P. Castonguay, "On the Non-linear Stability of Flux Reconstruction Schemes," Journal of Scientific Computing, vol. 50, no. 2, pp. 434–445, 2012.

[85] A. Sheshadri and A. Jameson, "On the Stability of the Flux Reconstruction Schemes on Quadrilateral Elements for the Linear Advection Equation," Journal of Scientific Computing, vol. 67, no. 2, pp. 769–790, 2015.

[86] P. E. Vincent, P. Castonguay, and A. Jameson, "Insights from von Neumann analysis of high-order flux reconstruction schemes," Journal of Computational Physics, vol. 230, no. 22, pp. 8134–8154, 2011.

[87] K. Asthana and A. Jameson, "High-Order Flux Reconstruction Schemes with Minimal Dispersion and Dissipation," Journal of Scientific Computing, vol. 62, no. 3, pp. 913–944, 2015.

[88] P. E. Vincent, P. Castonguay, and A. Jameson, "A New Class of High-Order Energy Stable Flux Reconstruction Schemes," Journal of Scientific Computing, vol. 47, no. 1, pp. 50–72, 2011.

[89] P. Castonguay, P. Vincent, and A. Jameson, "A New Class of High-Order Energy Stable Flux Reconstruction Schemes for Triangular Elements," Journal of Scientific Computing, vol. 51, no. 1, pp. 224–256, 2012.

[90] H. Huynh, Z. Wang, and P. Vincent, "High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids," Computers & Fluids, vol. 98, pp. 209–220, 2014.

[91] K. Ou, P. E. Vincent, and A. Jameson, "High-Order Methods for Diffusion Equation with Energy Stable Flux Reconstruction Scheme," in 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposi, 2011.

[92] D. M. Williams and A. Jameson, "Energy Stable Flux Reconstruction Schemes for Advection-Diffusion Problems on Tetrahedra," Journal of Scientific Computing, vol. 59, no. 3, pp. 721–759, 2014.

[93] K. Asthana, M. R. López-Morales, and A. Jameson, "Non-linear stabilization of high-order Flux Reconstruction schemes via Fourier-spectral filtering," Journal of Computational Physics, vol. 303, pp. 269–294, 2015.

[94] J. Cagnone, B. Vermeire, and S. Nadarajah, "A p-adaptive LCP formulation for the compressible Navier-Stokes equations," Journal of Computational Physics, vol. 233, pp. 324–338, 2013.

[95] L. Shi and Z.J. Wang, "Adjoint-based error estimation and mesh adaptation for the correction procedure via reconstruction method," Journal of Computational Physics, vol. 295, pp. 261–284, 2015.

[96] J. Romero and A. Jameson, "Extension of the Flux Reconstruction Method to Triangular Elements using Collapsed-Edge Quadrilaterals," in 54th AIAA Aerospace Sciences Meeting, 2016.

[97] J. Romero, K. Asthana, and A. Jameson, "A Simplified Formulation of the Flux Reconstruction Method," Journal of Scientific Computing, vol. 67, no. 1, pp. 351–374, 2016.

[98] F. Witherden, B. Vermeire, and P. Vincent, "Heterogeneous computing on mixed unstructured grids with PyFR," Computers & Fluids, vol. 120, pp. 173–186, 2015.

[99] D. Williams, Energy stable high-order methods for simulating unsteady, viscous, compressible flows on unstructured grids. PhD thesis, Stanford University, 2013.

[100] B. Vermeire, S. Nadarajah, and P. Tucker, "Implicit Large Eddy Simulation Using the High-Order Correction Procedure via Reconstruction Scheme," International Journal for Numerical Methods in Fluids, vol. 82, 2015.

[101] T. Haga, S. Tsutsumi, S. Kawai, and R. Takaki, "Large-Eddy Simulation of a Supersonic Jet using High-Order Flux Reconstruction Scheme," in 53rd AIAA Aerospace Sciences Meeting, American Institute of Aeronautics and Astronautics, 2015.

[102] F. Witherden and A. Jameson, "Impact of Number Representation for High-Order Implicit Large-Eddy Simulations," AIAA Journal, vol. 58, pp. 1–14, 2019.

[103] H. Zhu, S. Fu, L. Shi, and Z.J. Wang, "A Hybrid RANS-Implicit LES Approach for the High-Order FR/CPR Method," 54th AIAA Aerospace Sciences Meeting, 2016.

[104] Z.J. Wang and H. T. Huynh, "A review of flux reconstruction or correction procedure via reconstruction method for the Navier-Stokes equations," Mechanical Engineering Reviews, vol. 2, no. 1, 2016.

[105] F. Witherden, P. Vincent, and A. Jameson, "High-Order Flux Reconstruction Schemes," in Handbook of Numerical Analysis, pp. 227–263, Elsevier, 2016.

[106] D. A. Kopriva, "A Conservative Staggered-Grid Chebyshev Multidomain Method for Compressible Flows. II. A Semi-Structured Method," Journal of Computational Physics, vol. 128, no. 2, pp. 475–488, 1996.

[107] Y. Liu, M. Vinokur, and Z.J. Wang, "Spectral Difference Method for Unstructured Grids I: Basic Formulation," Journal of Computational Physics, vol. 216, no. 2, pp. 780–801, 2006.

[108] Z.J. Wang, Y. Liu, G. May, and A. Jameson, "Spectral Difference Method for Unstructured Grids II: Extension to the Euler Equations," Journal of Scientific Computing, vol. 32, no. 1, pp. 45–71, 2007.

[109] G. May and A. Jameson, "A Spectral Difference Method for the Euler and Navier-Stokes Equations on Unstructured Meshes," in 44th AIAA Aerospace Sciences Meeting and Exhibit, 2006.

[110] Y. Sun, Z.J. Wang, and Y. Liu, "High-order multidomain spectral difference method for the Navier-Stokes equations on unstructured hexahedral grids," Communications in Computational Physics, vol. 2, no. 2, pp. 310–333, 2007.

[111] W. Chen, Y. Ju, and C. Zhang, "A collocated-grid spectral difference method for compressible flows," Computers & Fluids, vol. 196, p. 104341, 2020.

[112] C. Liang, R. Kannan, and Z.J. Wang, "A p-multigrid spectral difference method with explicit and implicit smoothers on unstructured triangular grids," Computers & Fluids, vol. 38, no. 2, pp. 254–265, 2009.

[113] Y. Sun, Z.J. Wang, and Y. Liu, "Efficient Implicit Non-linear LU-SGS Approach for Compressible Flow Computation Using High-Order Spectral Difference Method," Communications in Computational Physics, vol. 5, no. 2-4, pp. 760–778, 2009.

[114] B. Zhang and C. Liang, "A Simple, Efficient, High-order Accurate Sliding-Mesh Interface Approach to the Spectral Difference Method on Coupled Rotating and Stationary Domains," Journal of Computational Physics, vol. 295, pp. 147–16, 2015.

[115] A. DeJong and C. Liang, "Parallel spectral difference method for predicting 3D vortex-induced vibrations," Computers & Fluids, vol. 98, pp. 17–26, 2014.

[116] A. H. Mohammad, Z.J. Wang, and C. Liang, "Large Eddy Simulation of Flow over a Cylinder Using High-Order Spectral Difference Method," Advances in Applied Mathematics and Mechanics, vol. 2, no. 4, pp. 451–466, 2010.

[117] M. Parsani, Development of an efficient Navier-Stokes/LES solver on unstructured grids for high-order accurate schemes. PhD thesis, Vrije Universiteit Brussel, 2010.

[118] M. Parsani, G. Ghorbaniasl, C. Lacor, and E. Turkel, "An implicit high-order spectral difference approach for large eddy simulation," Journal of Computational Physics, vol. 229, no. 14, pp. 5373–5393, 2010.

[119] M. Yu, H. Hu, and Z.J. Wang, "A Numerical Study of Vortex-Dominated Flow around an Oscillating Airfoil with High-Order Spectral Difference Method," in 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2010.

[120] Y. Zhou and Z.J. Wang, "Implicit Large Eddy Simulation of Transitional Flow over a SD7003 Wing Using High-order Spectral Difference Method," in 40th Fluid Dynamics Conference and Exhibit, 2010.

[121] Y. Zhou and Z.J. Wang, "Simulation of CAA Banchmark Problems Using High-Order Spectral Difference Method and Perfectly Matched Layers," in 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2010.

[122] S. Premasuthan, C. Liang, and A. Jameson, "Computation Of Flows with Shocks Using Spectral Difference Scheme with Artificial Viscosity," in 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2010.

[123] T. Poinsot and S. Lele, "Boundary Conditions for Direct Simulations of Compressible Viscous Flows," Journal of Computational Physics, vol. 101, pp. 104–129, 1992.

[124] R. Fiévet, H. Deniau, J. Brazier, and E. Piot, "Numerical Study of Hypersonic Boundary-Layer Transition Delay through Second-Mode Absorption," in AIAA Scitech 2020 Forum, AIAA Paper 2020-2061.

[125] R. Fiévet, H. Deniau, and E. Piot, "Strong compact formalism for characteristic boundary conditions with discontinuous spectral methods," Journal of Computational Physics, vol. 408, p. 109276, 2020.

[126] J.-B. Chapelier, G. Lodato, and A. Jameson, "A study on the numerical dissipation of the Spectral Difference method for freely decaying and wall-bounded turbulence," Computers & Fluids, vol. 139, 2016.

[127] J. Wang, C. Liang, and M. S. Miesch, "A Compressible High-Order Unstructured Spectral Difference Code for Stratified Convection in Rotating Spherical Shells," Journal of Computational Physics, vol. 290, pp. 90–111, 2015.

[128] C. Liang, C. Cox, and M. Plesniak, "A comparison of computational efficiencies of spectral difference method and correction procedure via reconstruction," Journal of Computational Physics, vol. 239, pp. 138–146, 2013.

[129] C. Cox, W. Trojak, T. Dzanic, F. Witherden, and A. Jameson, "Accuracy, stability, and performance comparison between the spectral difference and flux reconstruction schemes," 2020.

[130] G. May, "On the Connection Between the Spectral Difference Method and the Discontinuous Galerkin Method," Communications in Computational Physics, vol. 9, no. 4, pp. 1071–1080, 2011.

[131] K. Van den Abeele, C. Lacor, and Z.J. Wang, "On the connection between the spectral volume and the spectral difference method," Journal of Computational Physics, vol. 227, no. 2, pp. 877–885, 2007.

[132] K. Van den Abeele, C. Lacor, and Z.J. Wang, "On the Stability and Accuracy of the Spectral Difference Method," Journal of Scientific Computing, vol. 37, no. 2, pp. 162–188, 2008.

[133] J. Vanharen, G. Puigt, X. Vasseur, J.-F. Boussuge, and P. Sagaut, "Revisiting the spectral analysis for high-order spectral discontinuous methods," Journal of Computational Physics, vol. 337, pp. 379–402, 2017.

[134] Z.J. Wang and Y. Liu, "The Spectral Difference Method for the 2D Euler Equations on Unstructured Grids," in 17th AIAA computational Fluid Dynamics Conference, AIAA Paper 2005-5112, 2005.

[135] Y. Liu, M. Vinokur, and Z.J. Wang, "Discontinuous spectral difference method for conservation laws on unstructured grids," in Computational Fluid Dynamics 2004, Proceedings of the Third International Conference on Computational Fluid Dynamics (ICCFD3), Toronto, 12-16 July 2004, pp. 449–454, 2006.

[136] C. Liang, R. Kannan, and Z.J. Wang, "A p-multigrid spectral difference method with explicit and implicit smoothers on unstructured grids," in 18th AIAA Computational Fluid Dynamics Conference, 25-28 June, Miami, FL, AIAA Paper 2007-4326, 2007.

[137] C. Liang, A. Jameson, and Z.J. Wang, "Spectral difference method for compressible flow on unstructured grids with mixed elements," Journal of Computational Physics, vol. 228, no. 8, pp. 2847–2858, 2009.

[138] A. Meister, S. Ortleb, T. Sonar, and M. Wirz, "A comparison of the Discontinuous-Galerkin and Spectral-Difference Method on triangulations using PKD polynomials," Journal of Computational Physics, vol. 231, pp. 7722–7729, 2012.

[139] M. Pawlak and T. Sonar, "A Spectral Difference Method with Dubiner Polynomials on Triangulations," in Series in Contemporary Applied Mathematics, pp. 610–616, Co-Published with Higher Education Press, dec 2012.

[140] M. G. Blyth and C. Pozrikidis, "A Lobatto interpolation grid over the triangle," IMA Journal of Applied Mathematics, vol. 71, pp. 153–169, feb 2006.

[141] A. Balan, G. May, and J. Schöberl, "A Stable and Spectral Difference and Method for Triangles," in 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 2011.

[142] A. Balan, G. May, and J. Schöberl, "A stable high-order Spectral Difference method for hyperbolic conservation laws on triangular elements," Journal of Computational Physics, vol. 231, no. 5, pp. 2359–2375, 2012.

[143] G. May and J. Schöberl, "Analysis of a Spectral Difference Scheme with Flux Interpolation on Raviart-Thomas Elements," tech. rep., Aachen Institute for Advanced Study in Computational Engineering Science, 2010.

[144] M. Li, Z. Qiu, C. Liang, M. Sprague, M. Xu, and C. A. Garris, "A new high-order spectral difference method for simulating viscous flows on unstructured grids with mixed-element meshes," Computers & Fluids, vol. 184, pp. 187–198, 2019.

[145] Z. Qiu, B. Zhang, C. Liang, and M. Xu, "A high-order solver for simulating vortex-induced vibrations using sliding-mesh spectral difference method and hybrid grids," International Journal for Numerical Methods in Fluids, vol. 90, pp. 171–194, 2019.

[146] A. Cassagne, J.-F. Boussuge, N. Villedieu, G. Puigt, I. D'Ast, and A. Genot, "JAGUAR: a new CFD code dedicated to massively parallel high-order LES computations on complex geometry," in 50th 3AF International Conference on Applied Aerodynamics, 2015.

[147] A. Cassagne, G. Puigt, and J. Boussuge, "High-order Method for a New Generation of Large Eddy Simulation Solver," tech. rep., Partnership for Advanced Computing in Europe (PRACE), 2015. Available online at www.prace-ri.eu.

[148] Z.J. Wang, "High-order methods for the Euler and Navier–Stokes equations on unstructured grids," Progress in Aerospace Sciences, vol. 43, pp. 1–41, 2007.

[149] C. Liang, S. Premasuthan, and A. Jameson, "High-order accurate simulation of low-Mach laminar flow past two side-by-side cylinders using spectral difference method," Computers & Structures, vol. 87, no. 11-12, pp. 812–827, 2009.

[150] G. E. Karniadakis and S. J. Sherwin, Spectral/HP Element Methods for Computational Fluid Dynamics. Oxford University Press, Oxford, 2005.

[151] J. Proriol, "Sur une famille de polynômes à deux variables orthogonaux dans un triangle," Sci. Paris 257, p. 2459–2461, 1957.

[152] T. Koornwinder, "Two-variable analogues of the classical orthogonal polynomials," in Theory and Applications of Special Functions (R. Askey, ed.), (San Diego), 1975.

[153] M. Dubiner, "Spectral methods on triangles and other domains," Journal of Scientific Computing, vol. 6, no. 4, pp. 345–390, 1991.

[154] S. Sherwin and G. Karniadakis, "A new triangular and tetrahedral basis for high-order (hp) finite element methods," International Journal For Numerical Methods In Engineering, vol. 38, pp. 3775–3802, 1995.

[155] G. Pena, Spectral element approximation of the incompressible Navier-Stokes equations in a moving domain and applications. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2009.

[156] P. Raviart and J. Thomas, "A mixed finite element method for 2nd order elliptic problem," in Lecture Notes in Mathematics 606, p. 292–315, 1977.

[157] J. Nedelec, "Mixed finite elements in R3," Numerische Mathematik, vol. 35, pp. 315–341, 1980.

[158] P. Castonguay, High-order energy stable flux reconstruction schemes for fluid flow simulations on unstructured grids. PhD thesis, Stanford University, 2012.

[159] G. Bedrosian, "Shape functions and integration formulas for three-dimensional finite element analysis," International Journal for Numerical Methods in Engineering, vol. 35, pp. 95–108, 1992.

[160] N. Nigam and J. Phillips, "High-order finite elements on pyramids: approximation spaces, unisolvency and exactness," IMA Journal of Numerical Analysis, vol. 32, no. 2, pp. 448–483, 2012.

[161] M. Bergot, G. Cohen, and M. Duruflé, "Higher-order Finite Elements for Hybrid Meshes Using New Nodal Pyramidal Elements," Journal of Scientific Computing, vol. 42, pp. 345–381, 2010.

[162] M. Bergot and M. Duruflé, "Higher-Order Discontinuous Galerkin Method for Pyramidal Elements using Orthogonal Bases," Numerical Methods for Partial Differential Equations, vol. 29, no. 1, pp. 144–169, 2013.

[163] J. Chan and T. Warburton, "Orthogonal Bases for Vertex-Mapped Pyramids," SIAM Journal on Scientific Computing, vol. 38, no. 2, pp. A1146–A1170, 2016.

[164] J. Chan, Z. Wang, A. Modave, J. Remacle, and T. Warburton, "GPU-accelerated discontinuous Galerkin methods on hybrid meshes," Journal of Computational Physics, vol. 318, pp. 142–168, 2016.

[165] J. Chan and T. Warburton, "A Comparison of High Order Interpolation Nodes for the Pyramid," SIAM Journal on Scientific Computing, vol. 37, no. 5, pp. A2151–A2170, 2015.

[166] F. Witherden, A. Farrington, and P. Vincent, "PyFR: An open source framework for solving advection–diffusion type problems on streaming architectures using the flux reconstruction approach," Computer Physics Communications, vol. 185, pp. 3028–3040, 2014.

[167] C. Pereira and B. Vermeire, "Spectral Properties of High-Order Element Types for Implicit Large Eddy Simulation," Journal of Scientific Computing, vol. 85, no. 48, 2020.

[168] P. Hammer, O. Marlowe, and A. Stroud, "Numerical Integration Over Simplexes and Cones," Mathematical Tables and Other Aids to Computation, vol. 10, no. 55, pp. 130–137, 1956.

[169] M. Laursen and M. Gellert, "Some criteria for numerically integrated matrices and quadrature formulas for triangles," International Journal for Numerical Methods in Engineering, vol. 12(1), pp. 67–76, 1978.

[170] D. Dunavant, "High Degree Efficient Symmetrical Gaussian Quadrature Rules for the Triangle," International Journal for Numerical Methods in Engineering, vol. 21, no. 6, pp. 1129–1148, 1985.

[171] L. Zhang, T. Cui, and H. Liu, "A set of symmetric quadrature rules on triangles and tetrahedra," J. Comput. Math., vol. 27, pp. 89–96, 2009.

[172] H. Xiao and Z. Gimbutas, "A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions," Computers and Mathematics with Applications, vol. 59, no. 2, pp. 663–676, 2010.

[173] B. Vioreanu and V. Rokhlin, "Spectra of Multiplication Operators as a Numerical Tool," SIAM Journal on Scientific Computing, vol. 36, pp. 267–288, 2014.

[174] D. Williams, L. Shunn, and A. Jameson, "Symmetric quadrature rules for simplexes based on sphere close packed lattice arrangements," Journal of Computational and Applied Mathematics, vol. 266, pp. 18–38, 2014.

[175] G. Cowper, "Gaussian quadrature formulas for triangles," Numerical Methods in Engineering, vol. 7(3), pp. 405–408, 1973.

[176] J. Lyness and D. Jespersen, "Moderate degree symmetric quadrature rules for the triangle," IMA Journal of Applied Mathematics, vol. 15, no. 1, pp. 19–32, 1975.

[177] F. Witherden and P. Vincent, "On the identification of symmetric quadrature rules for finite element methods," Computers and Mathematics with Applications, vol. 69, pp. 1232–1241, 2015.

[178] S.-A. Papanicolopulos, "Computation of moderate-degree fully symmetric cubature rules on the triangle using symmetric polynomials and algebraic solving," Computer and Mathematics with Applications, vol. 69, pp. 650–666, 2015.

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

[180] P. Silvester, "Symmetric quadrature formulae for simplexes," Mathematics of Computation, vol. 24, pp. 95–100, 1970.

[181] L. Shunn and F. Ham, "Symmetric quadrature rules for tetrahedra based on a cubic close-packed lattice arrangement," Journal of Computational and Applied Mathematics, vol. 236, no. 17, pp. 4348–4364, 2012.

[182] P. Keast, "Moderate degree tetrahedral quadrature formulas," Computer Methods in Applied Mechanics and Engineering, vol. 55, no. 3, pp. 339–348, 1986.

[183] Y. Jinyun, "Symmetyric Gaussian quadrature formulae for tetrahedronal regions," Computer Methods in Applied Mechanics and Engineering, vol. 43, pp. 349–353, 1984.

[184] Y. Liu and M. Vinokur, "Exact integrations of polynomials and symmetric quadrature formulas over arbitrary polyhedral grids," Journal of Computational Physics, vol. 140, pp. 122–147, 1998.

[185] J. Jaśkowiec and N. Sukumar, "High-order cubature rules for tetrahedra," Numerical Methods in Engineering, vol. 121, no. 11, pp. 2418–2436, 2020.

[186] S. Gottlieb and C. Shu, "Total Variation Diminishing Runge-Kutta Schemes," Mathematics of Computation, vol. 67, no. 221, p. 73–85, 1998.

[187] C. Bogey and C. Bailly, "A family of low dispersive and low dissipative explicit schemes for flow and noise computations," Journal of Computational Physics, vol. 194, pp. 194–214, 2004.

[188] S. Gottlieb, C.-W. Shu, and E. Tadmor, "Strong Stability-Preserving High-Order Time Discretization Methods," SIAM Review, vol. 43, no. 1, pp. 89–112, 2001.

[189] R. Spiteri and S. Ruuth, "A New Class of Optimal High-Order Strong-Stability-Preserving Time Discretization Methods," SIAM Journal on Numerical Analysis, vol. 40, no. 2, pp. 469—-491, 2002.

[190] A. Jameson, "Solution of the Euler equations for two dimensional transonic flow by a multigrid method," Applied Mathematics and Computation, vol. 13, no. 3, pp. 327–355, 1983.

[191] N. Chalmers and L. Krivodonova, "A robust cfl condition for the discontinuous galerkin method on triangular meshes," Journal of Computational Physics, vol. 403, 2020.

[192] S. Wandzura and H. Xiao, "Symmetric quadrature rules on a triangle," Computers and Mathematics with Applications, vol. 45, pp. 1829–1840, 2003.

[193] https://how5.cenaero.be/.

[194] P. L. Roe, "Approximate Riemann solvers, parameter vectors, and difference schemes," Journal of Computational Physics, vol. 43, no. 2, pp. 357–372, 1981.

[195] G.-S. Jiang and C.-W. Shu, "Efficient Implementation of Weighted ENO Schemes," Journal of Computational Physics, vol. 126, no. 1, pp. 202–228, 1996.

[196] W. Pazner and P.-O. Persson, "Approximate tensor-product preconditioners for very high order discontinuous Galerkin methods," Journal of Computational Physics, vol. 354, pp. 344–369, 2018.

[197] R. C. Swanson and S. Langer, "Comparison of NACA 0012 Laminar Flow Solutions: Structured and Unstructured Grid Methods," Technical Memorandum NASA-TM-2016-219003, NASA Langley Research Center, 2016.

[198] S. D. Miller, "Lift, drag and moment of a naca 0015 airfoil," tech. rep., Department of Aerospace Engineering, Ohio State University, 2008.

[199] S. C. R. Dennis and G.-Z. Chang, "Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100," Journal of Fluid Mechanics, vol. 42, no. 3, pp. 471–489, 1970.

[200] D. Russell and Z. Jane Wang, "A cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow," Journal of Computational Physics, vol. 191, no. 1, pp. 177–205, 2003.

[201] D. Calhoun, "A Cartesian Grid Method for Solving the Two-Dimensional Streamfunction-Vorticity Equations in Irregular Regions," Journal of Computational Physics, vol. 176, no. 2, pp. 231–275, 2002.

[202] https://www1.grc.nasa.gov/research-and-engineering/hiocfd/.