Innovative numerical schemes for 3D supersonic aerodynamics on unstructured mesh.

Agnès Chan

THÈSE de DOCTORAT

PRÉSENTÉE à

L'UNIVERSITÉ DE BORDEAUX

ECOLE DOCTORALE DE MATHEMATIQUES ET

D'INFORMATIQUE (EDMI)

Par **Agnes CHAN**

Innovative numerical schemes for 3D supersonic aerodynamics on unstructured mesh.

Soutenue le 18 Novembre 2022

ENCADREMENT

Raphael Loubère	DR CNRS, Université de Bordeaux, France	Directeur de thèse
Pierre-Henri Maire	Ingénieur-Chercheur CEA-Cesta, HDR, France	Directeur de thèse

JURY

Rémi Abgrall	Pr. - Université de Zürich, Suisse	Rapporteur
Claus-Dieter Munz	Pr. - Université de Stuttgart, Allemagne	Rapporteur
Héloïse Beaugendre	McF. - Bordeaux INP, France	Examinatrice
Christophe Chalons	Pr. - Université de Versailles, Saint-Quentin-en-Yvelines, France	Examinateur
Angelo Iollo	Pr. - Université de Bordeaux, France	Président
Marie-Hélène Vignal	McF. - Institut de Mathématiques de Toulouse, France	Examinatrice

Résumé

Après une pause de quelques décennies, un certain intérêt refait surface pour la conception d'aéronefs hypersoniques, appuyé par l'accroissement général des capacités de calculs à hautes performances facilitant les simulations numériques d'écoulements aérodynamiques multidimensionnels. Le modèle physique sous-jacent est constitué par les équations de Navier-Stokes décrivant les lois de conservation de la masse, de la quantité de mouvement et de l'énergie d'un fluide compressible visqueux et thermo-conducteur. Les variables physiques de l'écoulement subissent des variations spatiales et temporelles importantes et complexes. Des méthodes numériques robustes et précises sont nécessaires pour capturer l'ensemble des phénomènes présents dans l'écoulement.

Les méthodes numériques historiques reposent sur des approches de type volumes finis. Il s'agit d'une formulation intégrale de loi de conservation dans laquelle la variation temporelle de la valeur moyenne dans la maille est régie par les flux aux interfaces de la maille. Dans le cadre des méthodes volumes finis pour les équations hyperboliques, les flux aux interfaces sont construits à partir de la solution du problème de Riemann (solveur de Riemann). Une caractéristique commune des solveurs de Riemann existant est qu'ils ont été développés dans le formalisme Eulérien. Il est néanmoins possible d'adopter une approche différente. Dans ce travail, nous adoptons le point de vue de Gallice (2003) qui consiste à construire des solveurs de Riemann Eulériens simples à partir de leur homologues Lagrangien à l'aide de la transformation Lagrange-Euler pour les systèmes de lois de conservation de la mécanique des milieux continus.

Une classe de schémas volumes finis centrés aux cellules basée sur la transformation Lagrange-Euler est introduite. La construction du schéma prend sa source dans l'idée d'étendre le schéma centré Lagrangien (Loubère *et al.* (2016)), pour la dynamique des gaz multidimensionnel au cadre Eulerien en tirant parti de l'approche introduite dans Shen *et al.* (2014). Les flux numériques du schéma sont évalués avec un solveur de Riemann aux noeuds. Ce solveur permet de calculer la vitesse nodale nécessaire pour déplacer la grille en description Lagrangienne. Par conséquent, la conservativité et la stabilité d'entropie du schéma résultent d'une équation vectorielle basée aux noeuds et une inéquation scalaire. La transformation Lagrange-Euler par Gallice (2002*a*, 2003) et revisitée dans Chan *et al.* (2021) permet de construire des solveurs de Riemann eulériens positifs et entropiques à partir de solveurs lagrangiens, à condition qu'une condition sur le pas de temps explicite soit remplie.

Dans le premier chapitre, nous avons repris le formalisme de Gallice (2003) pour développer un schéma de type volumes finis unidimensionel positif et entropiquement stable grace à la transformation Lagrange-Euler. Dans les deuxième et troisième chapitres, nous présentons la théorie de la méthode volumes finis appliquée aux sous-faces des éléments de maillage, suivie de son application aux équations de la dynamique des gaz multidimensionnelle. Dans le dernier chapitre, une extension pour construire un schéma aux sous-faces équilibré pour les équations de Saint-Venant est présentée.

Mots clés: lois de conservations hyperboliques, schéma de type Godunov, solveur de Riemann approché simple, stabilité entropique, descriptions Lagrangienne et Eulérienne.

Abstract

After a break of few decades, a certain interest re-emerges for hypersonic aircraft design largely because of the availability of numerical simulations for multi-dimensional aerodynamic flows. The underlying physical model is the Navier-Stokes equations describing the conservation laws of mass, momentum and energy of a viscous and heat-conductive compressible fluid. The physical variables of the flow undergo important and complex spatial and temporal variations. Hence, robust and accurate numerical methods are mandatory to capture such flows.

Historical numerical methods are based on Finite Volume type approaches. Such an approach is based on an integral formulation of the conservation laws in which the temporal variation of the average value in the cell is governed by the fluxes at the interfaces of the cell. In the context of Finite Volume methods for hyperbolic equations, the fluxes at the interfaces are constructed from the solution of the Riemann problem, obtained with an eponymous Riemann solver. A common feature of existing Riemann solvers is that they were developed in the Eulerian formalism. It is nevertheless possible to adopt a different approach: in this work, we adopt the point of view of Gallice (2003) which consists in constructing simple Eulerian Riemann solvers from their Lagrangian counterpart with the help of the Lagrange-to-Euler mapping for systems of conservation laws of continuum mechanics.

A class of cell-centered Finite Volume scheme based upon the Lagrange-to-Euler mapping will be introduced. This mapping establishes a fundamental relation bridging the Lagrangian and Eulerian descriptions. The construction of the scheme originates from the idea of extending the cell-centered Lagrangian scheme (refer to Loubère *et al.* (2016)) for multidimensional gas dynamics on the Eulerian framework at the same time benefiting from the approach introduced by Shen *et al.* (2014). The numerical fluxes of the scheme are evaluated by means of an approximate Riemann solver located at the grid nodes, which provides the nodal velocity required to move the grid in a compatible manner. The conservation condition and entropy stability of the scheme result respectively from a node-based vectorial equation and a scalar inequation. The Lagrange-to-Euler mapping introduced by Gallice (2002*a*, 2003) and revisited in Chan *et al.* (2021) then allows to build positive and entropic Eulerian Riemann solvers from their Lagrangian counterparts, provided that an explicit time step condition is fulfilled.

The first chapter focuses on the one-dimensional case of a positivity-preserving and entropy stable Finite Volume scheme obtained by bridging the Lagrangian and Eulerian description, after the formalism of Gallice (2003). In the second and third chapters, we present the theory of an original subface-based multidimensional Finite Volume method followed by its application to multidimensional gas dynamics equations. In the last chapter, an extension to build a well-balanced subface-based scheme for shallow-water equations is presented before coming to a conclusion.

Keywords: hyperbolic system of conservation laws, Godunov-type scheme, simple approximate Riemann solver, entropy stability, Lagrangian and Eulerian representations.

Acknowledgements

Firstly, I would like to express my sincere gratitude to both my advisors, Raphael Loubère and Pierre-Henri Maire, for their patience, motivation, immense knowledge and plentiful experience. Their dedicated guidance helped me in all the time of research and writing of this thesis. I could not have imagined having better advisors and mentors for my Ph.D study.

I would like to thank Gérard Gallice, the source of inspiration to the study of this thesis. As the saying goes, retirement is not the end of the road but the beginning of the open highway. His commitment was essential to the accomplishment of this thesis.

My sincere thanks also goes to the rest of my thesis committee for their insightful comments and constructive question. I would like to thank Rémi Abgrall and Claus-Dieter Munz for doing the honours as my thesis referees, same goes toHéloïse Beaugendre, Christophe Chalons, Angelo Iollo and Marie-Hélène Vignal who agreed to be examiners.

I am also grateful to colleagues, friends and research team in University of Catania that help make the work on shallow-water possible. A special shout out to Dharshana for the times we spent together, from sharing the PhD experience to wedding organization.

I would like to thank my parents and my sisters for supporting me spiritually throughout this thesis and my life in general.

The best outcome from these past years is my husband, Thibault, who is also my best friend, confidant and mentor. You've seen me at my worst, but choose to help me better. Thank you for everything 💙

Last but not least, Biscuit who kept me company throughout these years working from home, Thank you furry much!

Contents

1 Introduction

Contents

The conservation laws of gas dynamics, magneto-hydrodynamic flows and other branches of classical physics are typically expressed by nonlinear hyperbolic systems of Partial differential equations (PDEs) that prepare mathematically a host of wave interactions occurrences. The solutions of nonlinear hyperbolic systems generate jump discontinuities that propagate on as shock waves for instance. The mathematical theory may be challenging since it is necessary to confront weak solutions. In consequence, numerical simulation is often employed to tackle hyperbolic problems. A Riemann Solver (RS), should it be exact or approximated, is one of the essential tool in Computational Fluid Dynamics (CFD). In all generality, it is a method for computing the evolution of a discontinuity separating two constant physical states, namely a Riemann problem. The Riemann problem introduced by Riemann (1860) entered the field of computational science more than 150 years ago and S.K. Godunov *et al.* (1979); Godunov (1999) proposed the first Finite-Volume (FV) scheme to simulate compressible gas dynamics equations employing a Riemann solver that makes use of the exact solution of a Riemann problem. This first-order accurate method gained popularity and is now known as the Godunov method/scheme in CFD. However, this exact Riemann solver may become prohibitively expensive when the Riemann problem becomes too complex, leading to the subsequent development of alternative approximate and less expensive Riemann Solvers following on Godunov approach - to cite but a few, Roe (1981), Harten *et al.* (1983) Lax and van Leer (HLL), Einfeldt (1988)(HLLE),Munz *et al.* (1991) (HLLEM), Toro *et al.* (1994) (HLLC), Engquist & Osher (1981), Osher (1984) and others and we refer the interested readers to the book of Toro (1999), Godlewski & Raviart (1996) and more recently Qu *et al.* (2022).

1.1 Approximate Riemann solvers

Of all the aforementioned solvers, the most well-known is probably the one due to Roe (1981). It is based on a linearization of the compressible gas dynamics equations. Numerical cases indicate that the Roe solver has a high resolution in capturing non-linear jumps such as shocks discontinuities and linear contact discontinuities. However, by construction, it unfortunately has the drawback to create nonphysical results, *i.e* entropy violating discontinuous waves, or to generate the infamous carbuncle phenomenon. Note that the effect of this drawback can be lessened by implementing an appropriate entropy fix. Over the years, the HLL by Harten *et al.* (1983) and HLLE by Einfeldt (1988) Riemann solvers also gained considerable popularity thanks to their simplicity. However, they are considered an incomplete Riemann solvers due to the structure of the solution which neglects some intermediate characteristic fields. This also has the downfall of causing excessive numerical dissipation. The modified version of HLL, namely the HLLC by Toro *et al.* (1994) Riemann solver, takes into account the presence of those intermediate waves, *i.e* contact discontinuities and shear waves, thus making it a complete Riemann solver. However, in its current form it does not include an exact computation of the waves speeds that are simply estimated using uppermost and lowermost bounds, refer for instance to Batten *et al.* (1997). Furthermore, the HLLC Riemann solver also generates the carbuncle phenomenon.

The aforementioned approximate Riemann solvers are intrinsically one-dimensional in their framework and in spite of this rather good success of approximate schemes, they are presumably less efficient in multidimensional problems due to the directional bias. The standard way of extending to several dimensions with the one-dimensional schemes is through flux splitting algorithms by solving one-dimensional Riemann problems in a direction normal to each cell interface for all faces independently to achieve multidimensional behavior. This notion neglects flow variations that might be propagating in a transversal direction. Therefore, researchers have been on an endeavor to develop multidimensional Riemann solvers to solve Riemann problems in different directions of the interface, hoping to better capture flow characteristics. The first truly multidimensional solver by Deconinck *et al.* (1993) brought compelling contribution to this topic. The key element of this solver is to assume continuous piecewise linear space variation of the variables defined at cell vertices, allowing for a multidimensional generalization without employing 1D Riemann problems or dimensional splitting. Among other attempts in contributing to multidimensional solvers is the generic multidimensional HLLE solver of Balsara & Dumbser (2015). The base idea is to use the one-dimensional Riemann solver in all directions to coupled with multidimensional correction terms to constitute a multidimensional solver for conservative hyperbolic systems.

1.1.1 Properties of approximate Riemann solvers

One common feature of most of the present solvers is that they were developed in the Eulerian framework point of view. Contrastingly, it is possible to adopt a different approach by tackling the resolution of the Riemann problem from the Lagrangian standpoint. The

main feature of Lagrangian numerical methods lies in the fact that the motion of the fluid is intrinsically linked to the geometrical transformation that follows the fluid path, providing a natural framework to track interfaces of multidimensional flows. One of the earliest work on Lagrangian Godunov-type scheme was by Munz (1994) where a Roe linearization was constructed for the gas dynamics equations on the Lagrangian coordinates. The Roe linearization in Lagrangian coordinates produces non-physical states such as negative specific volume. However, the Lagrangian method shows promising results especially when approximating low density regions and also preserving sharp contact discontinuity. The works in this thesis are substantially inspired by the works of Gallice (2003). The simplicity of the Lagrangian formulation not only allows for a direct estimation of waves speeds for the Eulerian Riemann solvers in an ordered manner but it also facilitates the study of particular properties (conservation, positivity-preserving and entropy stable). Recurrently, numerical approximations may generate negative density or pressure which then leads to instability or code crash. This event is more critical in the Lagrangian framework on account of to the moving and deforming grid during computation. The positivity-preserving property and the entropy consistency can be assessed easily in the Lagrangian framework. Once a Lagrangian Riemann solver is built, the Eulerian counter-part that inherits the properties of the Lagrangian one can be obtained, refer to Gallice (2002*a*); Chan *et al.* (2021). Besides Gallice, only few authors have been working adopting the resolution from a Lagrangian standpoint, for instance, Cheng & Shu (2014) and Shu developed positivity-preserving HLLC approximate Riemann solver for the Lagrangian scheme in one and two dimensional for compressible Euler equations with general equations of state. Amongst others, Vilar *et al.* (2016) also worked on positivity-preserving approximate Riemann solvers conducted for both ideal gas and non ideal gas equations of state and extending to high-order accuracy with appropriate limitation.

As for the multidimensional Lagrangian case, the one-dimensional Riemann solver cannot be implemented directly due to the vast number of neighboring cells sharing a node. Hence, in this framework, the numerical fluxes are evaluated by means of an approximate Riemann solver located at the grid nodes that provides the nodal velocity required to move the grid in a compatible manner. Two main approaches are classically used, namely the staggered scheme and cell-centered Finite Volume schemes. This work focuses on the cell-centered approach. Different techniques may be employed to build the numerical fluxes and move the grid through the use of approximate Riemann solvers with respect to the Geometrical Conservation Law. In Després & Mazeran (2005) and Maire (2009), truly multidimensional Lagrangian schemes for arbitrary unstructured mesh are proposed. In this approach, for the two-dimensional case, the Finite Volume scheme is established using subcell forces that are expressed in terms of the difference between the velocity in the cell and the nodal velocity, instigating non-conservativity of the scheme. Consequently, the conservation of the scheme is retrieved via a nodal conservation condition stating that the sum of the subcell forces impacting a node must be equal to zero. This gives birth to a nodal solver, the cornerstone of Lagrangian schemes in order to evaluate the velocity of displacement of the nodes, thus allowing a compatible movement of the grid with the Geometrical Conservation Law. This new approach endorsed the

construction of multidimensional Finite Volume schemes for Lagrangian gas dynamics that is firmly based on mathematical and physical principles.

Lately, Shen *et al.* (2014) were inspired by the aforementioned Finite Volume schemes for Lagrangian gas dynamics and they adapted it by extending to the case of Eulerian gas dynamics. They employed a HLLC-like approximate Eulerian Riemann solver using subface fluxes of which the approximation is based on Maire (2009) nodal solver, making it possible to ensure the conservation of the scheme. However, the construction of their scheme does not guarantee wave speeds ordering and lacks an in-depth study of important properties such as positive density and internal energy and entropy inequality. Nevertheless, the numerical results proved to be promising with this method and encouraged us to carry out an extensive work not only to correct the previous flaws. The scheme established by Shen *et al.* (2014) and the introduction of simple approximate Riemann solvers by Gallice (2003) provided a starting point to achieve a generalization of formulation of multidimensional Finite volume scheme for all types of hyperbolic systems.

1.1.2 The carbuncle phenomenon

By the same token, the dimensionally split approximate Riemann solvers also give rise to the occurrence of various forms of numerical shock instabilities when simulating shock wave propagation, *i.e.*, odd-even decoupling and carbuncle, for instance refer to Quirk (1994). The carbuncle phenomena, analogically to the cluster of boils developed under the skin, is a numerical instability that manifest as a tumor-like growth on a shock wave or sawtooth disturbances at the shock front. When implementing an approximate Riemann solver with low numerical dissipation, such instabilities appear as serrated disturbance of a flow featured as a nonphysical default and causes the capturing of shock waves to be inaccurate if not totally spoiled. The earliest report on numerical instabilities by Quirk (1994) states that low-dissipative schemes causing expansive growth of acoustic waves may be the root of the carbuncle phenomenon and a list of benchmark test cases is then proposed to check the susceptibility of certain solvers to this shock instability. These problems include the odd-even decoupling and its variation, supersonic flow over a blunt body, shock diffraction over a 90° corner and the double Mach reflection problem. Kemm (2018) investigate the origin of the carbuncle instability and a review of theoretical results provided a set of numerical test to sort out the issues for the instability. The numerical viscosity of the scheme was put to cause however, a deeper understanding of the amount of viscosity needed to stabilize the shock is still lacking. Thus, there is a strong need for really multidimensional formulations that resolve all characteristic fields while instituting necessary dissipation. For instance, Rodionov (2017) constructs a multipurpose remedy by adding multidimensional artificial viscosity. Sangeet & Manda (2018) then proposed a modified HLLC scheme that employs a simple differentiable pressure-ratio based multidimensional shock sensor, namely the HLLC-ADC (Anti-Diffusion Control) scheme Fleischmann *et al.* (2020) also proposes a modification of the HLLC Riemann solver with a centralized reformulation of the numerical flux that reduces the acoustic dissipation, offering a shock-stable solver. All these works rely on adding some sort of additional dissipative term to fight the creation of such spurious modes. Also, follow-

ing Rodionov (2017), we might conjecture that a genuinely multidimensional dissipation is certainly helpful to damp the carbuncle instability. In this current work, we will be bringing into play a node-based conservation condition by involving all states surrounding a node. Upon that, transversal information are not omitted and this strategy substantiates the pre-existing assumption that multidimensional dissipation takes effect on shock instabilities problem. **The real theoretical analysis is yet to be done in order to fully understand the phenomena.**

1.1.3 Constructing a entropy-stable and positivity-preserving multidimensional numerical scheme

The purpose of this thesis is to present an entropy-stable and positivity-preserving Godunov-type numerical scheme for multidimensional hyperbolic systems on unstructured grids. In the interest of building Riemann Solvers with good properties, we adopt Gallice (2000, 2002*a*, 2003)'s work on the contact discontinuity preserving Lagrangian Riemann solver that has positivity-preserving property and entropy stability. We impose those properties onto the approximate Lagrangian Riemann solver and derive its Eulerian counter-part using the Lagrange-to-Euler mapping. Figure 1.1 illustrates the workflow of this thesis and the remaining of this document is organized as follows.

Following this introduction, Chapter 2 is dedicated to the one-dimensional Lagrangian and Eulerian Riemann solver for gas dynamics. Relying on the seminal work of Gallice (2003), a simple approximate Lagrangian Riemann solver is derived firstly. This solver preserves contact discontinuities shear waves and an explicit condition on the wave speeds of the approximate Riemann solver is derived in order to provide refined positivity and entropy stability properties. Then, following the general methodology proposed in Gallice (2002*a*, 2003), we deduce the simple approximate Eulerian Riemann solver from the Lagrangian one. It naturally inherits the positivity and entropy stability properties from its Lagrangian counterpart. A high-order extension up to the $5th$ order of accuracy with the a posterior high-order limiting strategy is then achieved.

In Chapter 3, we describe a multidimensional Finite Volume scheme in a general form, associated to an approximate simple Riemann solver resulting from subface fluxes approximation that relies on the notion of simple Eulerian Riemann solver introduced in the seminal work of Gallice (2003). In the interest of building Riemann solvers with good properties, The Eulerian Riemann solver is constructed from its Lagrangian counterpart by means of the Lagrange-to-Euler mapping. This systematic procedure ensures the transfer of good properties such as positivity preservation and entropy stability. In this framework, the conservativity and the entropy stability are no more locally face-based but result respectively from a scalar inequation and a node-based vectorial equation by calling upon the Lagrangian nodal solver. The corresponding multi-dimensional Finite Volume scheme is characterized by an explicit time step condition ensuring positivity preservation and entropy stability.

For a thorough demonstration, in Chapter 4, the numerical scheme is applied on the two-dimensional gas dynamics equation in the first place. An associated Finite Volume

simulation code has been built in multi-dimensions for unstructured meshes. Parallelization has been accomplished using the MPI library embedded in PETSc. After assessing the scheme in a two-dimensional case, the three-dimensional case, which is a direct extension of the two-dimension one is achieved. A large set of numerical experiments shows that the proposed solver is less sensitive to spurious instabilities such as the infamous carbuncle, compared to the classical one. To further improve accuracy, the current scheme has been extended to second-order in time and space. An implicit time discretization is also implemented in order to study steady inviscid flows.

In Chapter 5, we implement the numerical scheme on conservative laws with source terms, namely the shallow-water equations, with the aim of building a well-balanced Godunov-type scheme. Similarly to the previous chapters, a simple approximate Eulerian Riemann solver is built from the Lagrangian counterpart in one-dimensional and a large set of numerical experiments were done in order to assess the positivity-preserving and well-balanced properties of the scheme. We then follow up with the two-dimensional extension by incorporating the two-dimensional shallow-water equations in the subface-based scheme.

Conclusions and perspectives are then drawn in the final section.

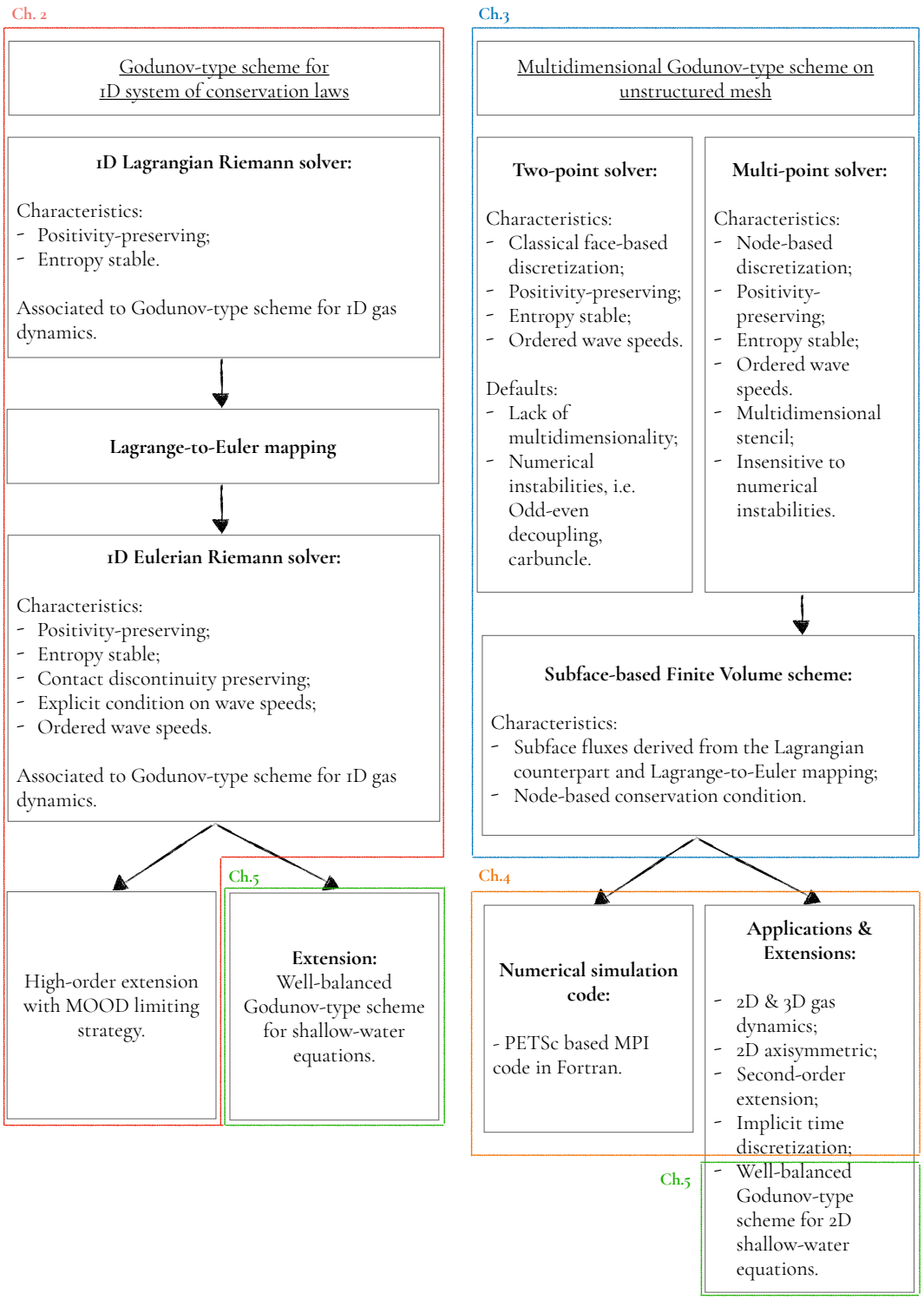


Figure 1.1: Workflow Diagram.

1.2 Communications

This work has lead to the following publications and conferences:

Publications

1. A. Chan, G. Gallice, R. Loubère and P.-H. Maire, *Positivity preserving and entropy consistent approximate Riemann solvers dedicated to the high-order MOOD- based Finite Volume discretization of Lagrangian and Eulerian gas dynamics*, Computers & Fluids, 2021.

2. G. Gallice, A. Chan, R. Loubère and P.-H. Maire, *Entropy stable and positivity preserving Godunov-type schemes for multidimensional hyperbolic systems on unstructured grid*, Journal of Computational Physics, 2022.

3. G. Gallice, A. Chan, R. Loubère and P.-H. Maire, *Entropy stable and positivity preserving Godunov-type schemes for three-dimensional hyperbolic systems on unstructured grid*, To be submitted.

4. A. Chan, M. J. Castro, G. Gallice, R. Loubère and P.-H. Maire, *A well-balanced and positive Finite Volume scheme for Shallow-Water equations in multidimensions for unstructured grids derived from Lagrangian roots*, To be submitted.

Conferences

1. April 2022: European workshop on HONOM (High-Order NOnlinear numerical Methods for evolutionary PDE) (Braga, Portugal).

2. June 2022: EDP Seminar at Strasbourg University, (Strasbourg, France).

3. June 2021: 10-ieme Biennale Francaise des Mathématiques Appliquées et Industrielles (La Grande Motte, France).

4. July 2020: 14th WCCM-Eccomas Congress (Paris, France).

2 One-dimensional Godunov-type Finite Volume scheme

Contents

This chapter is based upon the work presented in Chan *et al.* (2021) where we will be building on Gallice (2003)'s work on the contact discontinuity preserving Lagrangian Riemann solver that is positivity preserving and entropy stable. As presented in Gallice (2003)'s work, the positivity conservation and entropy stability properties are imposed onto the Riemann solver in the Lagrangian framework that will be used to derive its counterpart in the Eulerian framework. The associated numerical scheme in both the Lagrangian and Eulerian framework will be first order accurate in both space and time to begin with.

A high-order accurate extension is then introduced. Polynomial reconstruction using a least-square minimization algorithm is employed to gain high accuracy in space, further completed with Runge-Kutta time integrators. Robustness of the high-order computations and its ability to inherit the expected properties from the first order scheme are achieved thanks to the MOOD paradigm based on *a posteriori* limiting and local order decrementing. In this MOOD paradigm, of paramount importance, is the so-called 'parachute' scheme. This is a robust and dissipative scheme that is provably preserving the physical admissible states. The first-order Eulerian scheme based upon the Lagrangian Riemann solver developed will play this crucial role. Numerical tests are done to illustrate and validate this family of high-order Eulerian numerical schemes.

Therefore, the aim of this section is to:

- Recall the fundamental notions of approximate Riemann solvers and Godunov-type schemes.

- Present the relation between the Eulerian and Lagrangian forms of systems of conservation laws in 1D with interesting properties.
- Develop a first-order Godunov type scheme that will be used as the parachute scheme of the high-order MOOD extension.
- Validate the Lagrangian, Eulerian and high-order Eulerian schemes on demanding test cases.

2.1 Godunov-type schemes

In this section, we aim at recalling the fundamental notions of Godunov-type schemes and approximate Riemann solvers, which are often the cornerstone of any proper Finite Volume discretization dedicated to the numerical approximation of non-linear hyperbolic systems of conservation laws.

2.1.1 Notations and governing equations

Let us consider $\mathbf{U} = \mathbf{U}(x,t)$ a vector valued function which depends on the spatial coordinate $x \in \mathbb{R}$ and on the time $t > 0$. This function satisfies the system of conservation laws

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \mathbf{0}, \tag{2.1.1}$$

where $\mathbf{F}(\mathbf{U}) = \mathbf{F}(\mathbf{U}(x,t))$ denotes the flux vector. Its domain of definition , $\mathcal{D}$, is supposed convex. We assume that (2.1.1) is equipped with an entropy-entropy flux pair(Σ, Q), meaning that $\mathbf{U} \to \Sigma(\mathbf{U})$ is convex and Q satisfies

$$\frac{\partial Q}{\partial \mathbf{U}} = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^t \frac{\partial \Sigma}{\partial \mathbf{U}}.$$

We are looking for entropic solutions of (2.1.1) satisfying the entropy inequality, refer to Harten *et al.* (1983),

$$\frac{\partial \Sigma(\mathbf{U})}{\partial t} + \frac{\partial (\mathbf{U})}{\partial x} \leq \mathbf{0}. \tag{2.1.2}$$

The construction of Finite Volume numerical discretization of (2.1.1) generally relies on the solution of the Riemann problem,

$$\begin{aligned} \frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} &= \mathbf{0}. \\ \mathbf{U}(x,0) &= \begin{cases} \mathbf{U}_l & \text{if} \quad x < 0, \\ \mathbf{U}_r & \text{if} \quad x > 0. \end{cases} \end{aligned} \tag{2.1.3}$$

Let $\mathbf{W}$ be an approximation of the solution of this Riemann problem, namely an approximate Riemann solver. Generally, $\mathbf{W}$ shall depend on $\frac{x}{t}$, $\mathbf{U}_l$ and $\mathbf{U}_r$ and we thus denote

it by $\mathbf{W}\left(\frac{x}{t}, \mathbf{U}_l, \mathbf{U}_r\right)$ or $\mathbf{W}_{l,r}\left(\frac{x}{t}\right)$. The approximate Riemann solver has to fulfill the basic requirements

- $\mathbf{W}\left(\frac{x}{t}, \mathbf{U}_l, \mathbf{U}_r\right) = \mathbf{U}_l$ for $-\frac{x}{t}$ large enough,
- $\mathbf{W}\left(\frac{x}{t}, \mathbf{U}_l, \mathbf{U}_r\right) = \mathbf{U}_r$ for $\frac{x}{t}$ large enough,
- $\mathbf{W}\left(\frac{x}{t}, \mathbf{U}, \mathbf{U}\right) = \mathbf{U}$.

Let $x_{i+\frac{1}{2}}$ for $i = 0, \ldots, I+1$, with $x_{i+\frac{1}{2}} < x_{i+\frac{3}{2}}$ denote a sequence of nodes defining a one-dimensional grid and $\mathbf{U}_i^n$ for $i = 1, \ldots, I+1$ the numerical approximation fo the solution of (2.1.1) at time $t = t^n$ which is constant over the cell $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$.

Without loss of generality, we assume that the grid spacing $\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ is constant. For the sake of simplicity, we adopt the following notation convention where at each interface between two cells, $\mathbf{W}_{i+\frac{1}{2}}\left(\frac{x}{t}\right) = \mathbf{W}\left(\frac{x}{t}, \mathbf{U}_i, \mathbf{U}_{i+1}\right)$. With this in mind and following Harten *et al.* (1983), we shall recall the definition of a Godunov-type scheme.

Definition 2.1.1. [Godunov-type scheme] The approximate Riemann solver $\mathbf{W}$ induces a Godunov-type scheme given by

$$\mathbf{U}_i^{n+1} = \frac{1}{2}(\mathbf{U}_{i-\frac{1}{2}}^{+} + \mathbf{U}_{i+\frac{1}{2}}^{-}), \tag{2.1.4}$$

where

$$\mathbf{U}_{i-\frac{1}{2}}^{+} = \frac{2}{\Delta x}\int_0^{\frac{\Delta x}{2}} \mathbf{W}_{i-\frac{1}{2}}\left(\frac{x}{\Delta t}\right) dx \quad , \quad \mathbf{U}_{i+\frac{1}{2}}^{-} = \frac{2}{\Delta x}\int_{-\frac{\Delta x}{2}}^{0} \mathbf{W}_{i+\frac{1}{2}}\left(\frac{x}{\Delta t}\right) dx.$$

Here, $\Delta t = t^{n+1} - t^n > 0$ is the time step.

We are now in position to introduce the definition of the cell interface numerical fluxes that will be useful to construct Finite-Volume discretization. The numerical fluxes at interfaces $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$ viewed from cell i is deduced by integrating (2.1.1) respectively over $[t^n, t^{n+1}] \times [x_{i-\frac{1}{2}}, x_i]$ and $[t^n, t^{n+1}] \times [x_i, x_{i+\frac{1}{2}}]$ and are defined respectively by

$$\begin{aligned}
\mathbf{F}_{i-\frac{1}{2}}^{+} &= \mathbf{F}^{+}(\mathbf{U}_{i-1}, \mathbf{U}_i^n) = \mathbf{F}(\mathbf{U}_i) - \frac{\Delta x}{2\Delta t}\mathbf{U}_i^n + \frac{1}{\Delta t}\int_0^{\frac{\Delta x}{2}} \mathbf{W}_{i-\frac{1}{2}}\left(\frac{x}{\Delta t}\right) dx, \\
\mathbf{F}_{i+\frac{1}{2}}^{-} &= \mathbf{F}^{-}(\mathbf{U}_i, \mathbf{U}_{i+1}^n) = \mathbf{F}(\mathbf{U}_i) - \frac{\Delta x}{2\Delta t}\mathbf{U}_i^n - \frac{1}{\Delta t}\int_{\frac{\Delta x}{2}}^{0} \mathbf{W}_{i+\frac{1}{2}}\left(\frac{x}{\Delta t}\right) dx.
\end{aligned} \tag{2.1.5}$$

Definition 2.1.1 allows us to write the numerical scheme (2.1.4) under the form

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}(\mathbf{F}^-_{i+\frac{1}{2}} - \mathbf{F}^+_{i+\frac{1}{2}}). \tag{2.1.6}$$

Following Harten *et al.* (1983) and Bouchut (2004), for a small enough time step Δt, the left-sided and right-sided numerical fluxes at a generic interface written under a compact form yields

$$\begin{aligned} \mathbf{F}^+(\mathbf{U}_l, \mathbf{U}_r) &= \mathbf{F}(\mathbf{U}_r) + \int_0^{+\infty} (\mathbf{W}_{l,r}(\xi) - \mathbf{U}_r) d\xi, \\ \mathbf{F}^-(\mathbf{U}_l, \mathbf{U}_r) &= \mathbf{F}(\mathbf{U}_l) - \int_{-\infty}^{0} (\mathbf{W}_{l,r}(\xi) - \mathbf{U}_l) d\xi, \end{aligned} \tag{2.1.7}$$

eliminating the dependency on Δt and Δx.

2.1.2 Conservative Godunov-type schemes

In order to guarantee weak solutions satisfying Rankine-Hugoniot relations at discontinuities, it is important that the Godunov-type scheme be conservative. Let us consider the numerical scheme characterized by the piecewise constant approximation $\mathbf{U}_i^n$ satisfying (2.1.6). This numerical scheme is globally conservative if for all sequences $(\mathbf{U}_i^n)_i$ there holds

$$\sum_i \mathbf{U}_i^{n+1} = \sum_i \mathbf{U}_i^n.$$

Substituting (2.1.6) in the left-hand side of the foregoing equation leads to

$$\sum_i (\mathbf{F}^-_{i+\frac{1}{2}} - \mathbf{F}^+_{i-\frac{1}{2}}) = \mathbf{0} \Longleftrightarrow \sum_i (\mathbf{F}^-_{i+\frac{1}{2}} - \mathbf{F}^+_{i+\frac{1}{2}}) = \mathbf{0}.$$

Since, this must hold true for all sequences $(\mathbf{U}_i^n)_i$, the sufficient condition amounts to write $\mathbf{F}^-_{i+\frac{1}{2}} - \mathbf{F}^+_{i+\frac{1}{2}} = \mathbf{0}$ for all i. Finally, we claim that the scheme is locally conservative if for all left and right states, $\mathbf{U}_l$ and $\mathbf{U}_r$, we have $\mathbf{F}^+(\mathbf{U}_l, \mathbf{U}_r) = \mathbf{F}^-(\mathbf{U}_l, \mathbf{U}_r)$. The scheme is conservative if the left-sided and right-sided interface numerical fluxes coincide, refer to Godlewski & Raviart (1996). Computing the difference between the right-sided and left-sided fluxes leads to

$$\mathbf{F}^+(\mathbf{U}_l, \mathbf{U}_r) - \mathbf{F}^-(\mathbf{U}_l, \mathbf{U}_r) = \mathbf{F}_r + \int_0^{\infty} (\mathbf{W}_{l,r}(\xi) - \mathbf{U}_r) d\xi - \mathbf{F}_l + \int_{-\infty}^{0} (\mathbf{W}_{l,r}(\xi) - \mathbf{U}_l) d\xi,$$

gathering the integral terms yields

$$\mathbf{F}^+(\mathbf{U}_l, \mathbf{U}_r) - \mathbf{F}^-(\mathbf{U}_l, \mathbf{U}_r) = \mathbf{F}_r - \mathbf{F}_l + \int_{-\infty}^{+\infty} (\mathbf{W}_{l,r}(\xi) - \mathbf{U}_0(\xi)) d\xi. \tag{2.1.8}$$

This allows to recall the well-known results from Harten *et al.* (1983) regarding the consistency of the approximate Riemann solver with the integral form of the system of conservation laws.

Proposition 2.1.1 (Consistency with the integral form of the conservation laws)**.** The approximate Riemann solver $\mathbf{W}$ is consistent with the integral form of the conservation law (2.1.1) if and only if

$$\int_{-\infty}^{+\infty} [\mathbf{W}_{l,r}(\xi) - \mathbf{U}_0(\xi)]d\xi + \mathbf{F}_r - \mathbf{F}_l = \mathbf{0}. \tag{2.1.9}$$

In this case, the Finite Volume scheme (2.1.4) induced by the approximate Riemann solver is conservative.

2.1.3 Entropic Godunov-type schemes

The entropy fluxes can be defined similarly to what has been done previously for the left-sided and right-sided fluxes. We define the left-sided and right-sided entropy fluxes as

$$\begin{aligned} Q^+(\mathbf{U}_l, \mathbf{U}_r) &= Q_r - \frac{\Delta x}{2\Delta t}\Sigma_r + \frac{1}{\Delta t}\int_0^{\frac{\Delta x}{2}} \Sigma\left(\mathbf{W}_{l,r}\left(\frac{x}{\Delta t}\right)\right) dx, \\ Q^-(\mathbf{U}_l, \mathbf{U}_r) &= Q_l + \frac{\Delta x}{2\Delta t}\Sigma_l - \frac{1}{\Delta t}\int_{-\frac{\Delta x}{2}}^0 \Sigma\left(\mathbf{W}_{l,r}\left(\frac{x}{\Delta t}\right)\right) dx, \end{aligned} \tag{2.1.10}$$

where $Q_s = Q(\mathbf{U}_s)$ and $\Sigma_s = \Sigma(\mathbf{U}_s)$ for $s = l, r$. Introducing $Q^+_{i-\frac{1}{2}} = Q^+(\mathbf{U}^n_{i-1}, \mathbf{U}^n_i)$ and $Q^-_{i+\frac{1}{2}} = Q^-(\mathbf{U}^n_i, \mathbf{U}^n_{i+1})$ and subtracting $Q^+_{i-\frac{1}{2}}$ from $Q^-_{i+\frac{1}{2}}$, we arrive at

$$Q^-_{i+\frac{1}{2}} - Q^+_{i-\frac{1}{2}} = \frac{\Delta x}{\Delta t}\Sigma^n_i - \frac{1}{\Delta t}\left[\int_{-\frac{\Delta x}{2}}^0 \Sigma\left(\mathbf{W}_{i+\frac{1}{2}}\left(\frac{x}{\Delta t}\right)\right) dx + \int_0^{\frac{\Delta x}{2}} \Sigma\left(\mathbf{W}_{i-\frac{1}{2}}\left(\frac{x}{\Delta t}\right)\right) dx\right],$$

where $\Sigma^n_i = \Sigma(\mathbf{U}^n_i)$. By virtue of the entropy convexity and the use of Jensen (1906)'s inequality, we arrive at

$$\begin{aligned} \Sigma^{n+1}_i &\leq \frac{1}{2}(\Sigma^+_{i-\frac{1}{2}} + \Sigma^-_{i+\frac{1}{2}}) \\ &\leq \frac{1}{\Delta x}\left[\int_0^{\frac{\Delta x}{2}} \Sigma(\mathbf{W}_{i-\frac{1}{2}}\left(\frac{x}{\Delta t}\right))dx + \int_{-\frac{\Delta x}{2}}^0 \Sigma(\mathbf{W}_{i+\frac{1}{2}}\left(\frac{x}{\Delta t}\right))\right], \end{aligned}$$

where $\Sigma^+_{i-\frac{1}{2}} = \Sigma(\mathbf{U}^+_{i-\frac{1}{2}})$ and $\Sigma^-_{i+\frac{1}{2}} = \Sigma(\mathbf{U}^-_{i+\frac{1}{2}})$. Finally, the entropy inequality yields

$$\Sigma_i^{n+1} - \Sigma_i^n + \frac{\Delta t}{\Delta x}(Q^-_{i+\frac{1}{2}} - Q^+_{i-\frac{1}{2}}) \leq 0, \tag{2.1.11}$$

where the left-sided and right-sided entropy fluxes are respectively defined by (2.1.10). This inequality is the counterpart of (2.1.4) for entropy.

Similarly to what has been done for the numerical fluxes, the left-sided and right-sided entropy fluxes Q^- and Q^+ can be re-written in the compact form

$$\begin{aligned} Q^+(\mathbf{U}_l, \mathbf{U}_r) &= Q_r + \int_0^{+\infty} (\Sigma(\mathbf{W}_{l,r}(\xi)) - \mathbf{U}_r) d\xi, \\ Q^-(\mathbf{U}_l, \mathbf{U}_r) &= Q_l - \int_{-\infty}^{0} (\Sigma(\mathbf{W}_{l,r}(\xi)) - \mathbf{U}_l) d\xi, \end{aligned} \tag{2.1.12}$$

allowing to suppress the dependency of the entropy fluxes on Δt and δx.

At a discrete level, we aim at designing entropic numerical scheme characterized by the piecewise constant approximation $\mathbf{U}_i^n$ that satisfies (2.1.4). This numerical scheme is entropic if for all sequences $(\mathbf{U}_i^n)_i$, the total entropy decreases with respect to time

$$\sum_i \Sigma_i^{n+1} \leq \sum_i \Sigma_i^n,$$

where $\Sigma_i^n = \Sigma(\mathbf{U}_i^n)$ and $\Sigma_i^{n+1} = \Sigma(\mathbf{U}_i^{n+1})$. By virtue of (2.1.11), the total entropy is decreasing for all Δt and $(\mathbf{U}_i^n)_i$, provided that

$$\sum_i (Q^-_{i+\frac{1}{2}} - Q^+_{i+\frac{1}{2}}) \geq 0.$$

Following Harten *et al.* (1983) once again, we introduce the consistency of the approximate Riemann solver with the integral form of the entropy inequality.

Definition 2.1.2 (Consistency with the integral form of the entropy inequality)**.** The approximate Riemann solver $\mathbf{W}$ is consistent with the integral form of the entropy inequality (2.1.2) if and only if

$$\int_{-\infty}^{+\infty} [\Sigma(\mathbf{W}_{l,r}(\xi)) - \Sigma_0(\xi)] d\xi + Q_r - Q_l \leq 0. \tag{2.1.13}$$

In this case, the numerical scheme (2.1.4) is entropic.

2.1.4 Simple Riemann solvers

Now that we have introduced the notion of conservative and entropic Godunov-type numerical schemes, we recall the definition of simple approximate Riemann solvers following the pioneering works of Gallice (2003, 2000) and Bouchut (2004).

Definition 2.1.3 (Simple Riemann solver)**.** A Riemann solver $\mathbf{W}$ is simple if and only if it consists of $(m+1)$ constant states $(\mathbf{U}_k)_{k=1}^{m+1}$, separated by discontinuities of slopes $(\Lambda_k)_{k=1}^{m}$ in the (x,t) plane. More precisely,

$$\mathbf{W}_{lr}\left(\frac{x}{t}\right) = \begin{cases} \mathbf{U}_1 = \mathbf{U}_l & \text{if} \quad \dfrac{x}{t} < \Lambda_l, \\ \mathbf{U}_k & \text{if} \quad \Lambda_{k-1} < \dfrac{x}{t} < \Lambda_k, \quad k = 2, \dots, m, \\ \mathbf{U}_{m+1} = \mathbf{U}_r & \text{if} \quad \Lambda_m \leq \dfrac{x}{t}. \end{cases} \tag{2.1.14}$$

The simple approximate Riemann solver is denoted $((\mathbf{U}_k)_k, (\Lambda_k)_k)$.

As suggested by its name, the simple solvers represent the simplest form of approximate solvers. Among others, Roe (1981) solver, HLL solver by Harten *et al.* (1983) and HLLC solver from Toro (1999) are famous examples of simple approximate Riemann solvers. Before wrapping up this section, we recall some useful results that enable their characterization.

Proposition 2.1.2 (Characterization of conservative and entropic simple Riemann solver)**.** A simple Riemann solver induces a conservative Godunov-type scheme if and only if

$$-\sum_{k=1}^{m} \Lambda_k(\mathbf{U}_{k+1} - \mathbf{U}_k) + \mathbf{F}_r - \mathbf{F}_l = \mathbf{0}. \tag{2.1.15}$$

In this case, the numerical flux writes

$$\mathbf{H}(\mathbf{U}_l, \mathbf{U}_r) = \frac{1}{2}\left[\mathbf{F}_l + \mathbf{F}_r - \sum_{k=1}^{m} |\Lambda_k|(\mathbf{U}_{k+1} - \mathbf{U}_k)\right]. \tag{2.1.16}$$

In addition, the simple Riemann solver induces an entropic Godunov-type scheme if and only if

$$-\sum_{k=1}^{m} \Lambda_k(\Sigma_{k+1} - \Sigma_k) + Q_r - Q_l \leq 0. \tag{2.1.17}$$

The expression of the numerical flux (2.1.16) results from the computation of the left-sided flux $\mathbf{F}^-$ and the right-sided flux $\mathbf{F}^+$ at the interface $x = 0$. Substituting the simple approximate Riemann solver into (2.1.7) respectively, we arrive at

$$\begin{aligned} \mathbf{F}^- &= \mathbf{F}_l - \sum_{k=1}^{m} \Lambda_k^-(\mathbf{U}_{k+1} - \mathbf{U}_k), \\ \mathbf{F}^+ &= \mathbf{F}_r - \sum_{k=1}^{m} \Lambda_k^+(\mathbf{U}_{k+1} - \mathbf{U}_k), \end{aligned}$$

where for any real x, $x^+ = \frac{1}{2}(|x|+x)$ and $x^- = \frac{1}{2}(|x|-x)$ denote respectively the positive and the negative parts of x. The numerical scheme is conservative when the left-sided and right-sided fluxes coincide, namely $\mathbf{F}^+ - \mathbf{F}^- = \mathbf{0}$, hence the equation (2.1.15). In this case, the numerical flux at the interface is readily obtained as $\mathbf{H}(\mathbf{U}_l, \mathbf{U}_r) = \frac{1}{2}(\mathbf{F}^+ + \mathbf{F}^-)$. We proceed similarly with the entropy inequality computing the left sided Q^- and right-sided Q^+ entropy fluxes thanks to (2.1.10) respectively.

The following section introduces the conservative laws of gas dynamics in the Eulerian framework, followed by the fundamental relation between the Eulerian and Lagrangian form by recalling the Euler-Lagrangian mapping which allows to develop the gas dynamics system in the Lagrangian framework.

2.2 Positivity-preserving and entropy consistent approximate Riemann solver

To illustrate the theoretical notions previously introduced, we aim at recalling the construction of a simple approximate Riemann solver dedicated to the numerical simulation of one-dimensional gas dynamics. We start by recalling formally the links between the smooth and discontinuous solutions of the Lagrangian and Eulerian representations of one-dimensional gas dynamics equations. In the sequel, these notions shall be useful to derive approximate Riemann solvers in both representations. We also quote Gallice (1997) wherein a general framework relating Eulerian and Lagrangian representations of general systems of one-dimensional conservation laws is introduced.

2.2.1 One-dimensional gas dynamics under Eulerian form

The compressible Euler equations are dedicated to the modeling of inviscid non heat conducting fluids. The one-dimensional conservation laws of gas dynamics equation written under Eulerian representation consist of a set of partial differential equations (PDEs) that describes respectively the conservation of mass, momentum and total energy of a fluid particle at the current spatial position, x. Note that x is referred to as the Eulerian coordinate. Denoting the time, $t > 0$, the gas dynamics equations in the Eulerian framework written under the form of the system of conservation laws yield

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial}{\partial x}\mathbf{F}(\mathbf{U}) = \mathbf{0}. \tag{2.2.1}$$

Here, $\mathbf{U} = \mathbf{U}(x,t)$ and $\mathbf{F}(\mathbf{U}) = \mathbf{F}(\mathbf{U}(x,t))$ are respectively the vector of conservative variables and the flux vector. More precisely

$$\mathbf{U} = (\rho, \rho u, \rho e)^t, \quad \mathbf{F} = (\rho u, \rho u^2 + p, \rho u e + p u)^t,$$

where ρ denotes the mass density, u is the velocity, e is the specific total energy and p the pressure. The specific total energy is the sum of the specific internal energy ε, and the specific kinetic energy, $\frac{1}{2}u^2$, that is $e = \varepsilon + \frac{1}{2}u^2$.

The thermodynamic closure of this system is obtained through the use of an equation of state (EOS) that writes $p = p(\rho, \varepsilon)$. For numerical applications, we use the gamma gas law *i.e.*, $p = \rho(\gamma - 1)\varepsilon$ with γ the polytropic index of the gas.

2.2.2 Lagrange-Euler transformation

Let X denote the initial position of the material particle located at x at time $t > 0$. We represent the motion of this material particle with respect to time introducing the Lagrange-Euler mapping (LEM)

$$\Phi : X \mapsto x = \Phi(X, t). \tag{2.2.2}$$

By definition, this mapping satisfies $\Phi(X, 0) = X$ and its Jacobian, J, reads

$$J(X, t) = \frac{\partial \Phi}{\partial X}. \tag{2.2.3}$$

Noticing that $J(X, 0) = 1$, a continuity argument leads us to assume that $J(X, t) > 0$ for $t > 0$. This ensures that the LEM is a one-to-one mapping. The partial derivative of Φ with respect to t holding X fixed is nothing but the kinematic velocity

$$u(X, t) = \frac{\partial \Phi}{\partial t}|_X.$$

Time differentiating the Jacobian of the LEM and utilizing the definition of the kinematic velocity yields the Geometric Conservation Law (GCL)

$$\frac{\partial J}{\partial t} - \frac{\partial u}{\partial X} = 0. \tag{2.2.4}$$

Let us point out that the kinematic velocity and the GCL have been derived by means of the LEM and independently of the system of conservation laws (2.2.1). Obviously, the kinematic velocity coincides with the fluid velocity initially introduced in (2.2.1). Physical variables can be represented regardless in the Lagrangian form or in the Eulerian one. By an abuse of notation, we shall set

$$\mathbf{U}(X, t) = \mathbf{U}(\Phi(X, t), t) = \mathbf{U}(x, t), \tag{2.2.5}$$

in the sequel of the paper, knowing that $x = \Phi(X, t)$. Time differentiating the foregoing identity, holding X fixed, we arrive at

$$\frac{\partial \mathbf{U}(X, t)}{\partial t}|_X = \frac{\partial \mathbf{U}(x, t)}{\partial t}|_x + u\frac{\partial \mathbf{U}(x, t)}{\partial x}. \tag{2.2.6}$$

This equation expresses nothing but the fact that the Lagrangian time derivative (holding X fixed) coincides with the so-called material derivative. Finally, combining identity (2.2.6) and the GCL, we arrive at

$$J^{-1}\frac{\partial(J\mathbf{U})}{\partial t}|_X = \frac{\partial \mathbf{U}}{\partial t}|_x + \frac{\partial(u\mathbf{U})}{\partial x}. \tag{2.2.7}$$

This last identity will be of great help to transform the Eulerian system of conservation laws (2.2.1) into its Lagrangian counterpart.

Substituting identity (2.2.7) into (2.2.1) leads formally to

$$J^{-1}\frac{\partial(J\mathbf{U})}{\partial t}|_X + \frac{\partial}{\partial x}(\mathbf{F}(\mathbf{U}) - u\mathbf{U}) = \mathbf{0}. \tag{2.2.8}$$

Recalling that $J\,\mathrm{d}X = \mathrm{d}x$, the above equation turns into

$$\frac{\partial(J\mathbf{U})}{\partial t}|_X + \frac{\partial}{\partial X}(\mathbf{F}(\mathbf{U}) - u\mathbf{U}) = \mathbf{0}. \tag{2.2.9}$$

From now on, we omit to specify that the time derivative is taken holding X fixed. Let us point out that the first component of this system, *i.e.*, the mass conservation equation, boils down to the trivial equation $\dfrac{\partial(\rho J)}{\partial t} = 0$, which after time integration yields

$$(\rho J)(X,t) = \rho^0(X), \tag{2.2.10}$$

where $\rho^0(X) > 0$ denotes the initial mass density distribution. Thanks to mass conservation, the Jacobian rewrites $J = \rho^0\tau$, where $\tau = \dfrac{1}{\rho}$ is the specific volume assuming that $\rho > 0$. Substituting this expression of the Jacobian into the GCL leads to

$$\rho^0\frac{\partial \tau}{\partial t} - \frac{\partial u}{\partial X} = 0. \tag{2.2.11}$$

Finally, gathering the foregoing results we are now in position to write the Lagrangian form of the one-dimensional gas dynamics

$$\rho^0\frac{\partial \mathbf{V}}{\partial t} + \frac{\partial \mathbf{G}}{\partial X} = \mathbf{0}, \tag{2.2.12}$$

where $\mathbf{V} = (\tau, u, e)^t$ and $\mathbf{G} = (-u, p, pu)^t$ are respectively the vector of conservative variables and the flux vector in the Lagrangian frame. We observe that the second and third components of the flux and the conservative variables written in Lagrangian and Eulerian forms are related by

$$\mathbf{V} = \tau\mathbf{U}, \quad \mathbf{G} = \mathbf{F} - u\mathbf{U}. \tag{2.2.13}$$

Assuming that the specific internal energy is a strictly convex function with respect to the specific volume, τ, and to the specific entropy, η, the Hessian of $\varepsilon(\tau,\eta)$ is positive definite, giving

$$\frac{\partial^2\varepsilon}{\partial\tau^2}>0\ ,\ \frac{\partial^2\varepsilon}{\partial\eta^2}>0\ ,\ \left(\frac{\partial^2\varepsilon}{\partial\tau^2}\right)\left(\frac{\partial^2\varepsilon}{\partial\eta^2}\right)-\left(\frac{\partial^2\varepsilon}{\partial\tau\partial\eta}\right)^2>0 \tag{2.2.14}$$

This amounts to say that $(\tau,\varepsilon)\mapsto\eta(\tau,\varepsilon)$ is strictly concave. The thermodynamic closure of this system is completed by expressing the pressure, p, and the temperature, θ, in terms of the partial derivatives of the specific internal energy

$$p(\tau,\eta)=-\frac{\partial\varepsilon}{\partial\tau},\qquad \theta(\tau,\eta)=\frac{\partial\varepsilon}{\partial\eta}. \tag{2.2.15}$$

This corresponds to the complete equation of state. We also assume that the temperature is strictly positive, *i.e.*, $\theta>0$. The convexity assumption allows us to define the isentropic sound speed

$$\frac{a}{\tau}=\left(-\frac{\partial p}{\partial\tau}\right)^{\frac{1}{2}}=\left(\frac{\partial^2\varepsilon}{\partial\tau^2}\right)^{\frac{1}{2}}. \tag{2.2.16}$$

With this thermodynamic closure, one can easily demonstrate that the one-dimensional Eulerian gas dynamics equations consist of an hyperbolic system characterized by the real eigenvalues $\left(-\frac{a}{\rho^0\tau},0,\frac{a}{\rho^0\tau}\right)$. The interesting reader might find the detailed demonstration, for instance, in Godlewski & Raviart (1996). Moreover, the complete equation of state implies the fundamental Gibbs relation

$$\theta\,\mathrm{d}\eta=p\,\mathrm{d}\tau+\mathrm{d}\varepsilon. \tag{2.2.17}$$

Observing that $\mathrm{d}\varepsilon=\mathrm{d}e-u\,\mathrm{d}u$, the substitution of the Lagrangian gas dynamics equations into the Gibbs relation leads to the supplementary conservation law

$$\rho^0\frac{\partial\eta}{\partial t}=0. \tag{2.2.18}$$

This latter demonstrates that the entropy is conserved for smooth flows. More precisely, time integrating the above equation yields $\eta(X,t)=\eta(X,0)$. Namely, the entropy of a material particle at time $t>0$ is equal to its initial value. On the other hand, for non smooth flows, the physically admissible piecewise continuous solutions have to satisfy the entropy inequality

$$\rho^0\frac{\partial\eta}{\partial t}\geq 0. \tag{2.2.19}$$

The Eulerian counterpart of the Lagrangian entropy conservation equation (2.2.18) is

readily obtained recalling that $\rho J = \rho^0$ and thanks to identity (2.2.7)

$$\frac{\partial(\rho\eta)}{\partial t}|_x + \frac{\partial(\rho\eta u)}{\partial x} = 0. \tag{2.2.20}$$

This equation expresses the conservation of entropy for smooth flows in the Eulerian frame, or equivalently, that the entropy is conserved along the trajectory of material particles. For non smooth flows, the Eulerian counterpart of the Lagrangian entropy inequality (2.2.19) reads

$$\frac{\partial(\rho\eta)}{\partial t}|_x + \frac{\partial(\rho\eta u)}{\partial x} \geq 0. \tag{2.2.21}$$

2.2.3 Simple approximate Riemann solver for Lagrangian gas dynamics

Here, inspired by the seminal works of Gallice (2003), we start by designing a family of approximate Riemann solvers for gas dynamics equations written under Lagrangian representation, which depends on two parameters. Then, we describe how to monitor these parameters to ensure good theoretical properties such as positivity of specific volume and internal energy and also an entropy inequality. Finally, we arrive at the description of a family of Eulerian approximate Riemann solvers simply deduced from their Lagrangian counterparts by mimicking the Lagrange Euler mapping at the discrete level. In this framework, the good properties of the Lagrangian approximate Riemann solver are directly transferred to the Eulerian one. The Riemann problem related to Lagrangian gas dynamics reads

$$\frac{\partial \mathbf{V}}{\partial t} + \frac{\partial \mathbf{G}(\mathbf{V})}{\partial m} = \mathbf{0}, \qquad \mathbf{V}(m,0) = \begin{cases} \mathbf{V}_l & \text{if } m < 0, \\ \mathbf{V}_r & \text{if } m \geq 0. \end{cases} \tag{2.2.22}$$

Let $\mathbf{W}^{Lagr}\left(\mathbf{V}_l, \mathbf{V}_r, \frac{m}{t}\right)$ be an approximate solution to the Riemann problem (3.3.5) defined by

$$\mathbf{W}^{Lagr}\left(\mathbf{V}_l, \mathbf{V}_r, \frac{m}{t}\right) = \begin{cases} \mathbf{V}_l & \text{if} \quad \frac{m}{t} < \lambda_l, \\ \mathbf{V}_l^\star & \text{if} \quad \lambda_l < \frac{m}{t} < 0, \\ \mathbf{V}_r^\star & \text{if} \quad 0 < \frac{m}{t} < \lambda_r, \\ \mathbf{V}_r & \text{if} \quad \lambda_r < \frac{m}{t}. \end{cases} \tag{2.2.23}$$

Following Gallice (2003), $\mathbf{W}^{Lagr}$ is named simple Riemann solver since it consists of four constant states $(\mathbf{V}_l, \mathbf{V}_l^\star, \mathbf{V}_r^\star, \mathbf{V}_r)$ separated by the three discontinuities characterized by the speeds $(\lambda_l, 0, \lambda_r)$. Here, λ_l and λ_r are given strictly positive parameters. It is worth noticing that the three discontinuity speeds of $\mathbf{W}^{Lagr}$ are prescribed to mimic the three eigenvalues of the continuous problem (2.2.12). The structure of the simple Riemann solver for Lagrangian gas dynamics is displayed in figure 2.1. The intermediate states

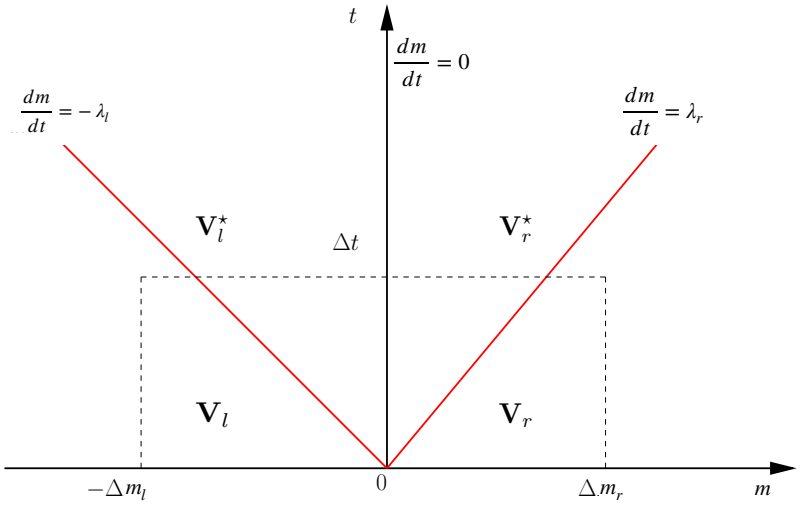


Figure 2.1: Simple Riemann solver for Lagrangian gas dynamics.

write $\mathbf{V}_s^\star = (\tau_s^\star, u_s^\star, e_s^\star)^t$ and $e_s^\star = \varepsilon_s^\star + \frac{1}{2}(u_s^\star)^2$ for $s = l, r$. For now on, for any variable or equation with subscript s we assume that it holds for $s = l, r$.

Characterization of the simple approximate Riemann solver. To characterize the approximate Riemann solver, it is useful to introduce the intermediate fluxes $\overline{\mathbf{G}}_l$ and $\overline{\mathbf{G}}_r$ defined by the jump relations across the left-sided and the right-sided waves

$$\lambda_l(\mathbf{V}_l^\star - \mathbf{V}_l) + \overline{\mathbf{G}}_l - \mathbf{G}_l = \mathbf{0}, \tag{2.2.24a}$$
$$-\lambda_r(\mathbf{V}_r - \mathbf{V}_r^\star) + \mathbf{G}_r - \overline{\mathbf{G}}_r = \mathbf{0}. \tag{2.2.24b}$$

By virtue of (2.1.15) the simple Riemann solver is conservative if and only if

$$\lambda_l(\mathbf{V}_l^\star - \mathbf{V}_l) - \lambda_r(\mathbf{V}_r - \mathbf{V}_r^\star) + \mathbf{G}_r - \mathbf{G}_l = \mathbf{0}. \tag{2.2.25}$$

Summing (2.2.24a) and (2.2.24b), we readily obtain that the Lagrangian simple approximate Riemann is conservative if and only if $\overline{\mathbf{G}}_l = \overline{\mathbf{G}}_r = \overline{\mathbf{G}}$ where $\overline{\mathbf{G}}$ denotes the common value of the intermediate flux

$$\overline{\mathbf{G}} = (-\overline{u}, \overline{p}, \overline{pu})^t.$$

With this notation, the jump relations (2.2.24a) and (2.2.24b) expand into

$$\lambda_l(\tau_l^\star - \tau_l) - (\overline{u} - u_l) = 0, \tag{2.2.26a}$$
$$\lambda_l(u_l^\star - u_l) + \overline{p} - p_l = 0, \tag{2.2.26b}$$
$$\lambda_l(e_l^\star - e_l) + \overline{pu} - p_l u_l = 0, \tag{2.2.26c}$$

and the right ones

$$\lambda_r(\tau_r^\star - \tau_r) + \overline{u} - u_r = 0, \tag{2.2.27a}$$
$$\lambda_r(u_r^\star - u_r) - (\overline{p} - p_r) = 0, \tag{2.2.27b}$$
$$\lambda_r(e_r^\star - e_r) - (\overline{pu} - p_r u_r) = 0. \tag{2.2.27c}$$

Assuming that the wave speeds λ_l and λ_r are known parameters, this results in a system of 6 scalar equations for 9 scalar unknowns $\tau_s^\star, u_s^\star, e_s^\star, \overline{u}, \overline{p}$ and $\overline{pu}$. Following Gallice (2002*a*, 2003), the closure of the foregoing systems shall rely on the natural assumptions:

- The numerical energy flux writes $\overline{pu} = \overline{p}\ \overline{u}$;
- Hypothesis (H_1) is fulfilled where

$$(H_1): \quad \begin{array}{rcl} \lambda_l(\tau_l^\star - \tau_l) - (u_l^\star - u_l) & = & 0, \\ 0(\tau_r^\star - \tau_l^\star) - (u_r^\star - u_l^\star) & = & 0, \\ -\lambda_r(\tau_r - \tau_r^\star) - (u_r - u_r^\star) & = & 0. \end{array} \tag{2.2.28}$$

Now, combining (2.2.28) with the first equations of (2.2.26) and (2.2.27), we arrive at $u_l^\star = u_r^\star = u^\star$, where $u^\star$ denotes the common value of the intermediate velocity and $u^\star = \overline{u}$, allowing to turn the system into a set of 6 scalar equations for 5 unknowns. Next, let us derive expressions of these unknowns in terms of the parameters λ_l and λ_r. These explicit expressions shall be useful to exhibit explicit conditions on the parameters to enforce positivity of specific volumes, specific internal energies and entropy control on the intermediate states of our approximate Riemann solver. We start by summing (2.2.26b) and (2.2.27b) and we readily get the explicit expression of $u^\star$ in terms of the parameters λ_l and λ_r

$$u^\star = \frac{\lambda_l u_l + \lambda_r u_r}{\lambda_l + \lambda_r} - \frac{p_r - p_l}{\lambda_l + \lambda_r}. \tag{2.2.29}$$

Then, substituting this expression into (2.2.26a) and (2.2.27a) yields the explicit expression of $\tau_l^\star$ and $\tau_r^\star$ in terms of parameters λ_l and λ_r as

$$\tau_l^\star = \frac{\lambda_l}{\lambda_l + \lambda_r}\left(\tau_l - \frac{[\![p]\!]}{\lambda_l^2}\right) + \frac{\lambda_r}{\lambda_l + \lambda_r}\left(\tau_l + \frac{[\![u]\!]}{\lambda_l}\right), \tag{2.2.30a}$$
$$\tau_r^\star = \frac{\lambda_l}{\lambda_l + \lambda_r}\left(\tau_r + \frac{[\![u]\!]}{\lambda_r}\right) + \frac{\lambda_r}{\lambda_l + \lambda_r}\left(\tau_r + \frac{[\![p]\!]}{\lambda_r^2}\right), \tag{2.2.30b}$$

where the bracket terms are the jump of the variables $[\![p]\!] = p_r - p_l$ and $[\![u]\!] = u_r - u_l$. It is also interesting to derive the balance of internal energy for the left and right intermediate states. To this end, we compute the balance of kinetic energy dot-multiplying the balance of momentum (2.2.26b) (resp. (2.2.27b)) by the centered velocity $\frac{1}{2}(u^\star + u_l)$ (resp. $\frac{1}{2}(u^\star + u_r)$). Then, subtracting the kinetic energy balance on the left (resp. right) side to the

total energy balance (2.2.26c) (resp. (2.2.27c)) we arrive at

$$\varepsilon_s^\star - \varepsilon_s + \frac{\overline{p} + p_s}{2}(\tau_s^\star - \tau_s) = 0. \tag{2.2.31}$$

Here, we have used the fact that λ_l and λ_r are strictly positive parameters and $e_s^\star = \varepsilon_s^\star + \frac{1}{2}u^\star$.This equation is similar to the well known Hugoniot equation that characterizes the locus of the states resulting from a shock wave. However, this is not exactly the Hugoniot equation since $\overline{p}$ does not correspond to a thermodynamic pressure, *i.e.*, $\overline{p} \neq p(\tau_s^\star, \eta_s^\star)$. In fact, $\overline{p}$ is directly expressed by solving the foregoing systems of equation. For instance, substituting (2.2.26a) (resp. (2.2.27a)) into (2.2.26b) (resp. (2.2.27b)) leads to the expression of $\overline{p} - p_s$ in terms of $\tau_s^\star - \tau_s$ and λ_s

$$\overline{p} - p_s = -\lambda_s^2(\tau_s^\star - \tau_s). \tag{2.2.32}$$

This equation shows that $\overline{p}$ is a decreasing function with respect to $\tau_s^\star$,which is a physical admissible behavior. We are now in position to derive explicit conditions on the parameters λ_l and λ_r to enforce positivity properties.

2.2.4 Positivity properties and inequality entropy of the Lagrangian simple Riemann solver

2.2.4.a Positivity of the specific internal energy

Let us derive an explicit condition on λ_s to ensure the positivity of $\varepsilon_s^\star$. We express $\varepsilon_s^\star$ in terms of ε_s and $\tau_s^\star - \tau_s$ substituting $p^\star$ expression (2.2.32) into the energy balance (2.2.31)

$$\varepsilon_s^\star = \varepsilon_s - p_s(\tau_s^\star - \tau_s) + \frac{\lambda_s}{2}(\tau_s^\star - \tau_s)^2. \tag{2.2.33}$$

Hence, the specific internal energy is a quadratic convex function with respect to the specific volume jump. It is thus always greater than or equal to its minimum value

$$\varepsilon_s^\star \geq \varepsilon_s - \frac{p_s^2}{2\lambda_s^2}.$$

Therefore, the positivity of specific internal energy $\varepsilon_s^\star$ is ensured provided that parameter λ_s satisfies the condition

$$\lambda_s \geq \frac{p_s}{\sqrt{2\varepsilon_s}}. \tag{2.2.34}$$

This condition is not overly restrictive. Indeed, for a perfect gas equation of state, $p = (\gamma - 1)\rho\varepsilon$, where $\gamma > 1$ is the polytropic index, the foregoing condition boils down to $\lambda_s \geq \sqrt{\frac{\gamma - 1}{2\gamma}}\rho_s a_s$. This shows that the parameter λ_s only needs to be greater than or equal to a fraction of the acoustic impedance to ensure the positivity of specific internal

energy. Condition (2.2.34) is relatively well known and has already been given in Vilar *et al.* (2016). Moreover, if the equation of state is such that the isentrope curves are convex, *i.e.*, the function $\tau \mapsto p(\tau, \eta)$ is strictly convex with respect to τ, then the following inequality holds true Menikoff & B.J.Plohr (1989)

$$\frac{a^2}{\tau^2} \geq \frac{p^2}{2\varepsilon}.$$

Combining this result with condition (2.2.34) leads to claim that for a convex equation of state, the approximate Riemann solver preserves the positivity of internal energy provided parameter λ_s fulfills

$$\lambda_s \geq \rho_s a_s. \tag{2.2.35}$$

This last condition simply expresses that the parameters should be greater than or equal to the acoustic impedance. We note in passing that the classical Godunov acoustic solver relying on the particular choice $\lambda_s = \rho_s a_s$ should preserve the positivity of the specific internal energy for any convex equation of state.

2.2.4.b Positivity of the specific volume

In this paragraph, we derive explicit conditions on λ_l, λ_r to ensure the strict positivity of $\tau_l^\star$ and $\tau_r^\star$. First, by virtue of (2.2.30a) and (2.2.30b), it is obvious that $\tau_l^\star$ and $\tau_r^\star$ result in a convex combination of the terms between parentheses. Thus, the positivity of the specific volume holds true provided that these terms between parentheses are positive. This leads us to claim that the positivity of $\tau_l^\star$ and $\tau_r^\star$ is taken for granted if λ_l and λ_r are defined by

$$\lambda_l = \max\left(\rho_l a_l, \sqrt{\frac{[\![p]\!]_+}{\tau_l}}, -\frac{[\![u]\!]}{\tau_l}\right), \quad \text{and} \quad \lambda_r = \max\left(\rho_r a_r, \sqrt{\frac{[\![-p]\!]_+}{\tau_r}}, -\frac{[\![u]\!]}{\tau_r}\right), \tag{2.2.36}$$

where for $x \in \mathbb{R}$, $(x)_+$ denotes the positive part of x, *i.e.*, $(x)_+ = \dfrac{1}{2}(x + |x|)$. Let us point out that the above condition provides us an explicit definition of the parameters λ_l and λ_r in terms of the initial left and right states. In addition, these parameters are always greater than or equal to the left and right acoustic impedances. We observe that this condition has been already proposed in Bouchut & de Luna (2009) where approximate Riemann solvers are constructed by means of relaxation schemes. In the sequel of the paper, this approach to monitor the parameters (λ_l, λ_r) will be referred to as **method 1**.

An alternative approach to enforce the positivity of the specific volumes, $(\tau_l^\star, \tau_r^\star)$, has been proposed in Gallice (2002*a*). It consists in assuming that the parameters, (λ_l, λ_r), are linked by the ratio $r = \dfrac{\lambda_r}{\lambda_l}$. Substituting $\lambda_r = r\lambda_l$ (resp. $\lambda_l = \dfrac{1}{r}\lambda_r$) into (2.2.30a) (resp. (2.2.30b)) leads to the following expressions for $\tau_l^\star$ (resp. $\tau_r^\star$)

$$\tau_l^\star = \frac{(1+r)\tau_l\lambda_l^2 + r[\![u]\!]\lambda_l - [\![p]\!]}{(1+r)\lambda_l^2}, \qquad \tau_r^\star = \frac{(1+r)\tau_r\lambda_r^2 + [\![u]\!]\lambda_r + r[\![p]\!]}{(1+r)\lambda_r^2}.$$

We observe that the positivity of $\tau_l^\star$ (resp. $\tau_r^\star$) is strictly equivalent to the positivity of the quadratic function with respect to λ_l (resp. λ_r) present at the numerator of the above rational expressions. Introducing the discriminants

$$\Delta_l = r^2[\![u]\!]^2 + 4(1+r)\tau_l[\![p]\!], \qquad \Delta_r = [\![u]\!]^2 - 4r(1+r)\tau_r[\![p]\!],$$

we arrive at the following conditions to ensure the positivity of the specific volumes.

- The positivity of $\tau_l^\star$ holds true if

$$\Delta_l \leq 0 \quad \text{or if} \quad \Delta_l > 0 \quad \text{and} \quad \lambda_l > \lambda_l^+ = \frac{1}{2(1+r)\tau_l}(-r[\![u]\!] + \sqrt{\Delta_l}), \quad \lambda_r > r\lambda_l^+. \tag{2.2.37}$$

- The positivity of $\tau_r^\star$ holds true if

$$\Delta_r \leq 0 \quad \text{or if} \quad \Delta_r > 0 \quad \text{and} \quad \lambda_r > \lambda_r^+ = \frac{1}{2(1+r)\tau_r}(-[\![u]\!] + \sqrt{\Delta_r}), \quad \lambda_l > \frac{1}{r}\lambda_r^+. \tag{2.2.38}$$

For practical applications, the parameter r is defined as the ratio of the acoustic impedances, that is, $r = \dfrac{\rho_r a_r}{\rho_l a_l}$. This method of monitoring the parameters (λ_l, λ_r) to enforce the positivity of the specific volumes will be referred to as **method 2** in the sequel of the text. Let us note that this approach might be also employed to enforce the positivity of the specific energies $\varepsilon_l^\star$ and $\varepsilon_r^\star$, the interested reader should refer to Gallice (2003) for more details.

The last approach to enforce the positivity of the specific volumes, $(\tau_l^\star, \tau_r^\star)$, **method 3**, consists in analyzing the signs of the right-hand side of (2.2.30a) and (2.2.30b) as functions of unknowns λ_l and λ_r with parameters $[\![u]\!]$, $[\![p]\!]$, τ_l and τ_r. The specific volumes $\tau_l^\star$ and $\tau_r^\star$ remain positive assuming that τ_l and τ_r are strictly positive, recalling that

$$\tau_l^\star = \tau_l + \frac{1}{\lambda_l(\lambda_l+\lambda_r)}\left(\lambda_r[\![u]\!] - [\![p]\!]\right), \qquad \tau_r^\star = \tau_r + \frac{1}{\lambda_l(\lambda_r+\lambda_r)}\left(\lambda_l[\![u]\!] - [\![p]\!]\right). \tag{2.2.39}$$

These explicit expressions of the specific volumes $\tau_l^\star$ and $\tau_r^\star$ have been obtained substituting the expression of $u^\star$ (2.2.29) into the left and right jump relations, respectively (2.2.26a) and (2.2.27a). To ease the study let us rename the unknowns as $x = \lambda_l > 0$, $y = \lambda_r > 0$ and the parameters are for $s = l, r$

$$a_s = \frac{[\![u]\!]}{\tau_s}, \quad b_s = \frac{[\![p]\!]}{\tau_s} \longrightarrow \operatorname{sgn}(a_l) = \operatorname{sgn}(a_r) \quad \text{and} \quad \operatorname{sgn}(b_l) = \operatorname{sgn}(b_r). \tag{2.2.40}$$

Notice that the signs of a and b are *a priori* unspecified but $\tau_l > 0$ and $\tau_r > 0$, and, $a_l = a_r$, $b_l = b_r$ if and only if $\tau_l = \tau_r$. Then (2.2.39) recasts into

$$(x,y) \in \mathbb{R}^+ \times \mathbb{R}^+ \longmapsto \begin{cases} L(x,y) = x(x+y) + a_l\, y - b_l, \\ R(x,y) = y(x+y) + a_r\, y + b_r. \end{cases} \tag{2.2.41}$$

Hence, depending on the signs of a and b we may have different situations to consider, each L and R functions having different positive regions. Because L and R must be positive simultaneously, the intersection of these regions will be carried on as a last step. Notice that the terms $x(x+y) \geq 0$ and $y(x+y) \geq 0$ because $(x,y) \in \mathbb{R}^+ \times \mathbb{R}^+$, therefore the negative parts of $L(x,y)$ and $R(x,y)$ are necessarily resulting from the terms $a_s y - b_s$. Let us first study $L(x,y)$ as a function of the sign of a_l and b_l.

Case ++: $a_l > 0$, $b_l > 0$ In this case we have only b_l which contributes for the negativity:

$$L(x,y) = \underbrace{x(x+y) + y, a_l}_{\geq 0} \underbrace{-b_l}_{\leq 0}, \tag{2.2.42}$$

The hyperbole L intersects the x-axis at $x = \sqrt{b_l}$ and the y-axis at $y = \frac{b_l}{a_l}$, and to ensure the positivity it is sufficient that (x,y) belongs to the following positivity region

$$\mathcal{C}_L^{++} = \left\{ M = (x,y), \quad \text{s.t.} \quad x > \sqrt{b_l} \text{ and } y > \frac{b_l}{a_l} \right\}. \tag{2.2.43}$$

Case +-: $a_l > 0$, $b_l < 0$ L is always positive in this case, hence $\mathcal{C}_L^{+-} = (\mathbb{R}^2)^+$.

Case -+: $a_l < 0$, $b_l > 0$ In this case L presents a vertical asymptote at location $x_l \in [\alpha_l, \beta_l]$ with $\alpha_l = \min(\sqrt{b_l}, -a_l)$ and $\beta_l = \max(\sqrt{b_l}, -a_l)$. The positivity region is then

$$\mathcal{C}_L^{-+} = \{ M = (x,y), \quad \text{s.t.} \quad x > \beta_l \}. \tag{2.2.44}$$

Case - -: $a_l < 0$, $b_l < 0$ This last case corresponds to a branch of hyperbole intersecting the y-axis and with a vertical asymptote at $x = -a_l$. Hence

$$\mathcal{C}_L^{--} = \left\{ M = (x,y), \quad \text{s.t.} \quad y < \frac{b_l}{a_l} \text{ or } x > -a_l \right\}. \tag{2.2.45}$$

The same study can be made for function R, the situation being equivalent inverting the role of x for y, and, b_l for $-b_r$, therefore the set $\mathcal{C}_R^{++}$, $\mathcal{C}_R^{+-}$, $\mathcal{C}_R^{-+}$ and $\mathcal{C}_R^{--}$ can be derived accordingly. Moreover, the constraints on the acoustic impedances state that $x \geq \rho_l a_l$ and $y \geq \rho_r a_r$.
At last gathering all constraints in the case of compression we can define the regions for $(x,y) = (\lambda_l, \lambda_r)$ ensuring the positivity of the specific volumes

$$\begin{aligned}
\mathcal{C}^{++} &= \left\{ M = (\lambda_l, \lambda_r), \text{ s.t. } \lambda_l \geq \max\left(\sqrt{b_l}, \rho_l a_l\right) \text{ and } \lambda_r \geq \max\left(\tfrac{b_l}{a_l}, \rho_r a_r\right) \right\}, \\
\mathcal{C}^{+-} &= \left\{ M = (\lambda_l, \lambda_r), \text{s.t.} \lambda_r \geq \max\left(\sqrt{-b_r}, \rho_r a_r\right) ; \text{and; } \lambda_l \geq \max\left(-\tfrac{b_l}{a_l}, \rho_l a_l\right) \right\}, \\
\mathcal{C}^{-+} &= \left\{ M = (\lambda_l, \lambda_r), \text{s.t.} \lambda_l \geq \max\left(\beta_l, \rho_l a_l\right) ; \text{and; } \lambda_r \geq \max\left(-a_r, \rho_r a_r\right) \right\}, \\
\mathcal{C}^{--} &= \left\{ M = (\lambda_l, \lambda_r), \text{s.t.} \lambda_r \geq \max\left(\sqrt{b_r}, \rho_r a_r\right) ; \text{and; } \lambda_l \geq \max\left(-a_l, \rho_l a_l\right) \right\},
\end{aligned} \tag{2.2.46}$$

where $\beta_l = \max(\sqrt{b_l}, -a_l)$ and $\beta_r = \max(\sqrt{-b_r}, -a_r)$. As final remarks, let us point that these sufficient constrains could be more precisely derived but at expensive cost. More importantly, recall that these constraints only apply in a compression situation.

2.2.4.c Entropy control

We aim at deriving a condition on λ_s parameters to ensure the entropy inequality, $\eta_s^\star \geq \eta_s$, across the discontinuities of the approximate Riemann solver. First, to obtain an expression of the entropy jump $\eta_s^\star - \eta_s$, we decompose the thermodynamic process $(\tau_s, \eta_s) \mapsto (\tau_s^\star, \eta_s^\star)$ into

- the isentropic process: $(\tau_s, \eta_s) \mapsto (\tau_s^\star, \eta_s)$, followed by
- the isochoric process: $(\tau_s^\star, \eta_s) \mapsto (\tau_s^\star, \eta_s^\star)$.

By virtue of this decomposition, the specific internal energy jump writes

$$\begin{aligned}
\varepsilon(\tau_s^\star, \eta_s^\star) - \varepsilon(\tau_s, \eta_s) &= \overbrace{\varepsilon(\tau_s^\star, \eta_s^\star) - \varepsilon(\tau_s^\star, \eta_s)}^{\text{isochoric process}} + \overbrace{\varepsilon(\tau_s^\star, \eta_s) - \varepsilon(\tau_s, \eta_s)}^{\text{isentropic process}} \\
&= \int_{\eta_s}^{\eta_s^\star} \frac{\partial \varepsilon}{\partial \eta}(\tau_s^\star, \eta)\, \mathrm{d}\eta + \int_{\tau_s}^{\tau_s^\star} \frac{\partial \varepsilon}{\partial \tau}(\tau, \eta_s)\, \mathrm{d}\tau, \\
&= \int_{\eta_s}^{\eta_s^\star} \theta(\tau_s^\star, \eta)\, \mathrm{d}\eta - \int_{\tau_s}^{\tau_s^\star} p(\tau, \eta_s)\, \mathrm{d}\tau.
\end{aligned}$$

The second line at the right-hand side of the above equation results from the definition of the complete equation of state (4.1.2) recalling that the absolute temperature, θ, is strictly positive. Substituting the expression of the internal energy jump (2.2.33) into the foregoing equation yields

$$\int_{\eta_s}^{\eta_s^\star} \theta(\tau_s^\star, \eta)\, \mathrm{d}\eta = \int_{\tau_s}^{\tau_s^\star} p(\tau, \eta_s)\, \mathrm{d}\tau - p_s(\tau_s^\star - \tau_s) + \frac{\lambda_s}{2}(\tau_s^\star - \tau_s)^2,$$

which is rearranged into

$$\int_{\eta_s}^{\eta_s^\star} \theta(\tau_s^\star, \eta)\, \mathrm{d}\eta = \int_{\tau_s}^{\tau_s^\star} (p(\tau, \eta_s) - p(\tau_s, \eta_s))\, \mathrm{d}\tau + \frac{\lambda_s}{2}(\tau_s^\star - \tau_s)^2. \tag{2.2.47}$$

Knowing that $\theta > 0$, the entropy inequality, $\eta_s^\star - \eta_s \geq 0$ holds true if and only if λ_s satisfies the condition

$$\lambda_s^2 \geq -\frac{2}{(\tau_s^\star - \tau_s)^2} \int_{\tau_s}^{\tau_s^\star} (p(\tau, \eta_s) - p(\tau_s, \eta_s))\, \mathrm{d}\tau, \quad \text{for} \quad s = l, r. \tag{2.2.48}$$

The right-hand side of the above inequality is always non-negative since $\dfrac{\partial p}{\partial \tau} < 0$ by virtue of the convexity assumption made on the equation of state and thus the above condition is

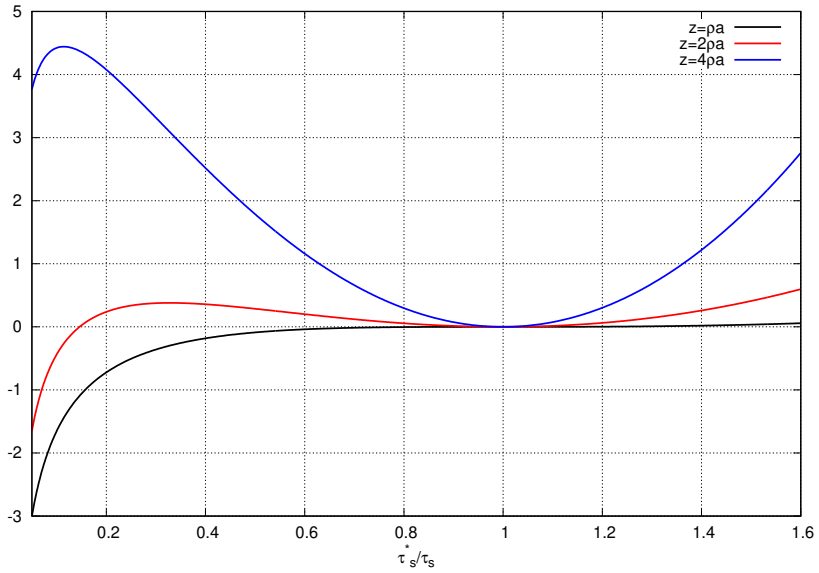


Figure 2.2: Plots of entropy production for a perfect gas equation of state ($\gamma = \dfrac{7}{5}$) for $\lambda_s = \rho_s a_s$, $\lambda_s = 2\rho_s a_s$ and $\lambda_s = 4\rho_s a_s$.

always well-defined. Thanks to the mean value theorem, it is also possible to reformulate condition (2.2.48) into

$$\lambda_s^2 \geq \frac{a^2(\bar{\tau}_s, \eta_s)}{\bar{\tau}_s^2}, \quad \text{where} \quad \bar{\tau}_s \in (\tau_s, \tau_s^\star) \quad \text{for} \quad s = l, r. \tag{2.2.49}$$

Although being simpler, the above condition is not very useful for practical application since it remains implicit by construction. We acknowledge that a similar condition has been derived in Chalons *et al.* (2013) in the framework of relaxation schemes. We conclude this paragraph performing an asymptotic analysis of the entropy production (2.2.47) when the specific volume jump is small, *i.e.*, $\tau_s^\star - \tau_s \ll 1$. In this case, the Taylor expansion of the right-hand side of (2.2.47) up to third-order leads to

$$\int_{\eta_s}^{\eta_s^\star} \theta(\tau_s^\star, \eta)\, \mathrm{d}\eta = \frac{1}{2}\left(\lambda_s^2 - \frac{a_s^2}{\tau_s^2}\right)(\tau_s^\star - \tau_s)^2 + \frac{1}{6}\frac{\partial^2 p}{\partial \tau^2}(\hat{\tau}_s, \eta_s)(\tau_s^\star - \tau_s)^3, \quad \text{where} \quad \hat{\tau}_s \in (\tau_s, \tau_s^\star).$$

This asymptotic analysis shows that the acoustic Godunov solver characterized by $\lambda_s = \dfrac{a_s}{\tau_s}$ has an entropy production of third-order with respect to the specific volume jump. In addition, for a convex equation of state, the second partial derivative of pressure with respect to specific volume is strictly positive, and the entropy production term is negative across compressive discontinuities which is not consistent with the second law of thermodynamics. For a perfect gas equation of state characterized by the polytropic index γ, the isentrope curve writes $p(\tau_s^\star, \eta_s) = p_s \left(\frac{\tau_s^\star}{\tau_s}\right)^\gamma$ for $s = l, r$. Substituting this

function into (2.2.47) yields the expression of the entropy production term

$$\int_{\eta_s}^{\eta_s^\star} \theta(\tau_s^\star, \eta)\,\mathrm{d}\eta = a_s^2 f\left(\frac{\tau_s^\star}{\tau_s}\right),$$
$$\text{where} \quad f(x) = \frac{1}{\gamma(\gamma-1)}(1-x^{1-\gamma}) + \frac{1}{\gamma}(1-x) + \frac{1}{2}\left(\frac{\lambda_s}{\rho_s a_s}\right)^2 (1-x)^2.$$

We have displayed in figure 2.2 the entropy production with respect to the ratio $\frac{\tau_s^\star}{\tau_s}$ for the three following values of the parameter: $\lambda_s = \rho_s a_s$, $\lambda_s = 2\rho_s a_s$ and $\lambda_s = 4\rho_s a_s$. It is clear that the Godunov acoustic solver characterized by $\lambda_s = \rho_s a_s$ exhibits a negative entropy production in the compressive zone, *i.e.*, for $\tau_s^\star \leq \tau_s$, whereas for λ_s sufficiently greater than the acoustic impedance, $\rho_s a_s$, the entropy production remains positive almost everywhere in the compressive zone.

This concludes the design of the Riemann solver in the Lagrangian framework. Next, we present the Godunov-type scheme associated to the Lagrangian Riemann solver.

2.2.5 Godunov-type scheme for Lagrangian gas dynamics

The aim of this section is to provide a first-order space and time Finite Volume discretization of the one-dimensional gas dynamics equations written under Lagrangian formalism. The resulting numerical method is a Godunov-type scheme since the updated cell-averaged value are computed by combining the cell-interface approximate Riemann solvers that have been constructed in section 2.2.3.

2.2.5.a Governing equations and notation

The one-dimensional Lagrangian gas dynamics equations read

$$\rho^0 \frac{\partial \mathbf{V}}{\partial t} + \frac{\partial \mathbf{G}}{\partial m} = \mathbf{0},$$

where $\mathbf{V} = (\tau, u, e)^t$, $\mathbf{G}(\mathbf{V}) = (-u, p, pu)^t$, $e = \varepsilon + \frac{1}{2}u^2$ and $\rho^0(X) > 0$ is the initial mass density. The set of physically admissible states for this system of conservation laws writes

$$\mathcal{A} = \left\{\mathbf{V} = (\tau, u, e)^t,\ \tau > 0,\ \varepsilon > 0\right\}. \tag{2.2.50}$$

The computational domain $\Omega = [X_{\min}, X_{\max}]$ is partitioned into N_c non overlapping cells $[X_{i-\frac{1}{2}}, X_{i+\frac{1}{2}}]$, where $X_{i+\frac{1}{2}}$ denotes the position of a generic node. The cell mass is constant and defined by $m_i = \displaystyle\int_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}} \rho^0(X)\,\mathrm{d}X$. At time t^n, we assume that the solution of the foregoing system of conservation laws is piecewise constant over each cell and defined by

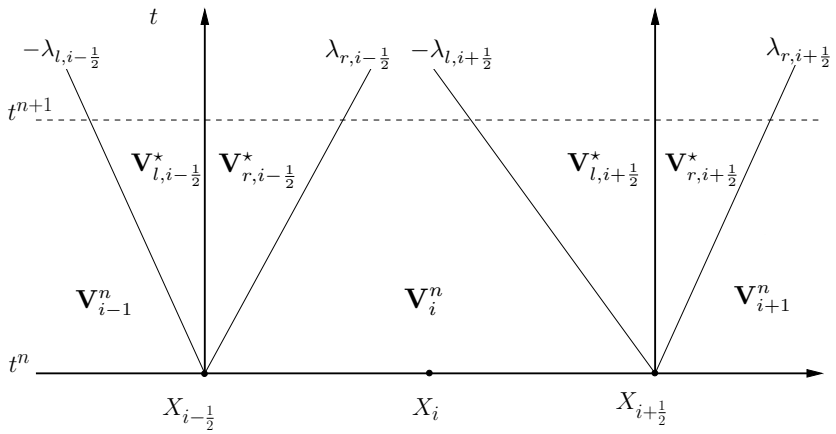


Figure 2.3: Lagrangian approximate Riemann solvers at cell interfaces.

the mass-averaged value

$$\mathbf{V}_i^n = \frac{1}{m_i}\int_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}} \rho^0(X)\mathbf{V}(X,t^n)\,\mathrm{d}X. \tag{2.2.51}$$

Let us compute the discrete solution of the system of conservation laws at time $t^{n+1} = t^n + \Delta t$ in terms of the piecewise solution at time t^n, where $\Delta t > 0$ denotes the time step. This will be done in combining the approximate Riemann solvers located at $X_{i-\frac{1}{2}}$ and $X_{i+\frac{1}{2}}$, refer to figure 2.3.

The time step, Δt, has been chosen sufficiently small to ensure that the rightgoing wave emanating from $X_{i-\frac{1}{2}}$ does not interact with the leftgoing wave emanating from $X_{i+\frac{1}{2}}$. The updated mass-averaged value writes

$$\begin{aligned} m_i\mathbf{V}_i^{n+1} = & \int_{X_{i-\frac{1}{2}}}^{X_i} \rho^0\mathbf{W}^{Lagr}\left(\mathbf{V}_{i-1}^n, \mathbf{V}_i^n, \frac{X-X_{i-\frac{1}{2}}}{\Delta t}\right)\mathrm{d}X \\ + & \int_{X_i}^{X_{i+\frac{1}{2}}} \rho^0\mathbf{W}^{Lagr}\left(\mathbf{V}_i^n, \mathbf{V}_{i+1}^n, \frac{X-X_{i+\frac{1}{2}}}{\Delta t}\right)\mathrm{d}X, \end{aligned} \tag{2.2.52}$$

where $\mathbf{W}^{Lagr}$ is the approximate Riemann solver defined by (2.2.23) and $X_i = \frac{1}{2}(X_{i-\frac{1}{2}} + X_{i+\frac{1}{2}})$. Performing the computation of the integrals, we arrive at the Godunov-type Finite Volume scheme for Lagrangian gas dynamics

$$m_i\mathbf{V}_i^{n+1} = \lambda_{r,i-\frac{1}{2}}\Delta t\mathbf{V}^{\star}_{r,i-\frac{1}{2}} + [m_i - (\lambda_{r,i-\frac{1}{2}} + \lambda_{l,i+\frac{1}{2}})\Delta t]\mathbf{V}_i^n + \lambda_{l,i+\frac{1}{2}}\Delta t\mathbf{V}^{\star}_{l,i+\frac{1}{2}}. \tag{2.2.53}$$

It is worth noting that from the above equation one can recover the classical Finite Volume written in flux form. To this end, let us rearrange (2.2.53) as follows

$$m_i\mathbf{V}_i^{n+1} = m_i\mathbf{V}_i^n - \lambda_{r,i-\frac{1}{2}}\Delta t\left(\mathbf{V}_i^n - \mathbf{V}^{\star}_{r,i-\frac{1}{2}}\right) + \lambda_{l,i+\frac{1}{2}}\Delta t\left(\mathbf{V}^{\star}_{l,i+\frac{1}{2}} - \mathbf{V}_i^n\right). \tag{2.2.54}$$

Now, applying the jump relations (2.2.24b) and (2.2.24a) respectively at $X_{i-\frac{1}{2}}$ and $X_{i+\frac{1}{2}}$ leads to

$$-\lambda_{r,i-\frac{1}{2}}\left(\mathbf{V}_i^n-\mathbf{V}_{r,i-\frac{1}{2}}^\star\right)=\mathbf{G}_{i-\frac{1}{2}}^\star-\mathbf{G}_i^n,\quad \lambda_{l,i+\frac{1}{2}}\left(\mathbf{V}_{l,i+\frac{1}{2}}^\star-\mathbf{V}_i^n\right)=-\mathbf{G}_{i+\frac{1}{2}}^\star+\mathbf{G}_i^n,$$

which allows us to introduce the numerical fluxes at cell interfaces $\mathbf{G}_{i-\frac{1}{2}}^\star$ and $\mathbf{G}_{i+\frac{1}{2}}^\star$. Finally gathering the foregoing results we arrive at the flux form Finite Volume scheme

$$\mathbf{V}_i^{n+1}=\mathbf{V}_i^n-\frac{\Delta t}{m_i}\left(\mathbf{G}_{i+\frac{1}{2}}^\star-\mathbf{G}_{i-\frac{1}{2}}^\star\right). \tag{2.2.55}$$

The cell interface flux is uniquely defined by

$$\mathbf{G}_{i+\frac{1}{2}}^\star=\begin{pmatrix}-u_{i+\frac{1}{2}}^\star\\ p_{i+\frac{1}{2}}^\star\\ p_{i+\frac{1}{2}}^\star u_{i+\frac{1}{2}}^\star\end{pmatrix}. \tag{2.2.56}$$

Here, $u_{i+\frac{1}{2}}^\star$ and $p_{i+\frac{1}{2}}^\star$ are obtained by solving (2.2.26) and (2.2.27) in terms of the parameters (λ_l,λ_r) at $X_{i+\frac{1}{2}}$.

2.2.5.b Time step monitoring

Dividing (2.2.53) by m_i yields

$$\mathbf{V}_i^{n+1}=\frac{\lambda_{r,i-\frac{1}{2}}\Delta t}{m_i}\mathbf{V}_{r,i-\frac{1}{2}}^\star+\left[1-\frac{(\lambda_{r,i-\frac{1}{2}}+\lambda_{l,i+\frac{1}{2}})\Delta t}{m_i}\right]\mathbf{V}_i^n+\frac{\lambda_{l,i+\frac{1}{2}}\Delta t}{m_i}\mathbf{V}_{l,i+\frac{1}{2}}^\star. \tag{2.2.57}$$

It is clear that $\mathbf{V}_i^{n+1}$ is a convex combination of $(\mathbf{V}_{r,i-\frac{1}{2}}^\star,\mathbf{V}_i^n,\mathbf{V}_{l,i+\frac{1}{2}}^\star)$ provided that the coefficients of the combination are non-negative. This implies the following time step monitoring to ensure the convex combination property

$$\Delta t\leq\frac{m_i}{\lambda_{r,i-\frac{1}{2}}+\lambda_{l,i+\frac{1}{2}}}\equiv\Delta t_i. \tag{2.2.58}$$

In the case of the Godunov acoustic solver the parameters (λ_l,λ_r) are equal to the cell acoustic impedance, *i.e.*, $\lambda_{r,i-\frac{1}{2}}=\lambda_{l,i+\frac{1}{2}}=\rho_i^n a_i^n$. Using mass conservation the cell mass writes $m_i=\rho_i^n\Delta x_i^n$, where $\Delta x_i^n=x_{i+\frac{1}{2}}^n-X_{i-\frac{1}{2}}^n$ denotes the Eulerian cell width, and the above time step condition boils down to the well known CFL-like condition

$$\Delta t\leq\frac{\Delta x_i^n}{2a_i^n}. \tag{2.2.59}$$

Here, the Lagrangian position of the cell interfaces are determined through the use of the discrete trajectory equation, *i.e.*, $x_{i+\frac{1}{2}}^{n+1}=x_{i+\frac{1}{2}}^n+\Delta t u_{i+\frac{1}{2}}^\star$.

In practice we compute the time-step as the minimum over all cells of the values Δt_i supplemented with a security factor $0 < CFL \leq 1$:

$$\Delta t = CFL \min_i \left(\frac{m_i}{\lambda_{r,i-\frac{1}{2}} + \lambda_{l,i+\frac{1}{2}}} \right). \tag{2.2.60}$$

2.2.5.c Positivity of specific volume and internal energy

If the time step condition (2.2.58) is taken for granted then through the Finite Volume scheme (2.2.57) τ_i^{n+1}, u_i^{n+1} and e_i^{n+1} are respectively convex combinations of $(\tau^\star_{r,i-\frac{1}{2}}, \tau_i^n, \tau^\star_{l,i+\frac{1}{2}})$, $(u^\star_{r,i-\frac{1}{2}}, u_i^n, u^\star_{l,i+\frac{1}{2}})$ and $(e^\star_{r,i-\frac{1}{2}}, e_i^n, e^\star_{l,i+\frac{1}{2}})$.

First, let us consider the case of the specific volume, assuming that $\tau_i^n > 0$. This means that the positivity of τ_i^{n+1} holds true provided that $\tau^\star_{r,i-\frac{1}{2}} > 0$ and $\tau^\star_{l,i+\frac{1}{2}} > 0$. These latter conditions are satisfied when the parameters (λ_l, λ_r) are determined according to the explicit formulas derived in section 2.2.4.b.

Now, let us investigate the positivity of $\varepsilon_i^{n+1} = e_i^{n+1} - \frac{1}{2}(u_i^{n+1})^2$. Scheme (2.2.57) provides

$$\begin{aligned} u_i^{n+1} =& \alpha_{r,i-\frac{1}{2}} u^\star_{i-\frac{1}{2}} + \alpha_i u_i^n + \alpha_{l,i+\frac{1}{2}} u^\star_{i+\frac{1}{2}}, \\ e_i^{n+1} =& \alpha_{r,i-\frac{1}{2}} e^\star_{r,i-\frac{1}{2}} + \alpha_i e_i^n + \alpha_{l,i+\frac{1}{2}} e^\star_{l,i+\frac{1}{2}}, \end{aligned}$$

where $\alpha_{r,i-\frac{1}{2}} = \frac{\lambda_{r,i-\frac{1}{2}} \Delta t}{m_i}$, $\alpha_{l,i+\frac{1}{2}} = \frac{\lambda_{l,i+\frac{1}{2}} \Delta t}{m_i}$ and $\alpha_i = 1 - (\alpha_{r,i-\frac{1}{2}} + \alpha_{l,i+\frac{1}{2}})$. Convexity of function $x \longmapsto x^2$ implies

$$\frac{1}{2}\left(u_i^{n+1}\right)^2 \leq \frac{1}{2}\alpha_{r,i-\frac{1}{2}} \left(u^\star_{i-\frac{1}{2}}\right)^2 + \frac{1}{2}\alpha_i \left(u_i^n\right)^2 + \frac{1}{2}\alpha_{l,i+\frac{1}{2}} \left(u^\star_{i+\frac{1}{2}}\right)^2.$$

This shows the loss of kinetic energy induced by the averaging process of the cell velocity. This is probably the main source of numerical dissipation inherent to the Godunov scheme. Subtracting the kinetic energy to the total energy at time t^{n+1} leads to the updated internal energy which therefore satisfies

$$\varepsilon_i^{n+1} \geq \alpha_{r,i-\frac{1}{2}} \varepsilon^\star_{r,i-\frac{1}{2}} + \alpha_i \varepsilon_i^n + \alpha_{l,i+\frac{1}{2}} \varepsilon^\star_{l,i+\frac{1}{2}}.$$

This latter equation shows that the kinetic energy is converted into internal energy via the averaging procedure of the Godunov scheme. Finally, if the time step condition (2.2.58) holds true and if we assume that $\varepsilon_i^n > 0$ then $\varepsilon_i^{n+1} > 0$ provided that $\varepsilon^\star_{r,i-\frac{1}{2}} > 0$ and $\varepsilon^\star_{l,i+\frac{1}{2}} > 0$. These latter conditions are satisfied when the parameters (λ_l, λ_r) are determined according to the explicit formulas derived in section 2.2.4.a.

2.2.5.d Entropy inequality

The cell-averaged entropy over $[X_{i-\frac{1}{2}}, X_{i+\frac{1}{2}}]$ at time t is given by

$$\eta_i(t) = \frac{1}{m_i}\int_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}} \rho^0(X)\eta\left(\mathbf{V}(X,t)\right)\,\mathrm{d}X. \tag{2.2.61}$$

Knowing that $\mathbf{V}(X,t)$ is piecewise constant with respect to X, leads to write $\eta_i^n = \eta(\mathbf{V}_i^n)$ and $\eta_i^{n+1} = \eta\left(\mathbf{V}_i^{n+1}\right)$, where $\mathbf{V}_i^{n+1}$ is computed from the Godunov-type scheme (2.2.57). Then, under the time step condition (2.2.58), the updated cell-averaged entropy satisfies

$$\begin{aligned}
\eta_i^{n+1} &= \eta\left(\alpha_{r,i-\frac{1}{2}}\mathbf{V}^\star_{r,i-\frac{1}{2}} + \alpha_i\mathbf{V}_i^n + \alpha_{l,i+\frac{1}{2}}\mathbf{V}^\star_{l,i+\frac{1}{2}}\right),\\
&\geq \alpha_{r,i-\frac{1}{2}}\eta\left(\mathbf{V}^\star_{r,i-\frac{1}{2}}\right) + \alpha_i\eta\left(\mathbf{V}_i^n\right) + \alpha_{l,i+\frac{1}{2}}\eta\left(\mathbf{V}^\star_{l,i+\frac{1}{2}}\right), \qquad \longleftarrow \text{thanks to } \eta \text{ concavity}\\
&\geq \alpha_{r,i-\frac{1}{2}}\eta^\star_{r,i-\frac{1}{2}} + \alpha_i\eta_i^n + \alpha_{l,i+\frac{1}{2}}\eta^\star_{l,i+\frac{1}{2}}.
\end{aligned}$$

Now, observing that $\alpha_{r,i-\frac{1}{2}} + \alpha_i + \alpha_{l,i+\frac{1}{2}} = 1$ we arrive at the inequality

$$\eta_i^{n+1} - \eta_i^n \geq \alpha_{r,i-\frac{1}{2}}(\eta^\star_{r,i-\frac{1}{2}} - \eta_i^n) + \alpha_{l,i+\frac{1}{2}}(\eta^\star_{l,i+\frac{1}{2}} - \eta_i^n). \tag{2.2.62}$$

This shows that the cell entropy increase with respect to time is controlled by the entropy increase across the right-sided and left-sided waves. Finally, the Godunov-type scheme satisfies the entropy inequality

$$\eta_i^{n+1} - \eta_i^n \geq 0, \tag{2.2.63}$$

provided that the Riemann approximate solver is entropic, that is $\eta^\star_{r,i-\frac{1}{2}} \geq \eta_i^n$ and $\eta^\star_{l,i+\frac{1}{2}} \geq \eta_i^n$. These latter conditions are taken for granted provided that the parameters (λ_l, λ_r) satisfy the constraint (2.2.49).

2.2.5.e Summary and algorithm

The algorithm starts from the data of $\mathbf{V}_i^n$ at initial time index $n = 0$ for all cells index i and from the data for the boundary conditions. We then execute successively the steps until the final time $t_{\text{final}} = t^{n+1}$ is reached:

– **In.** Data $\mathbf{V}_i^n$ for all cells.

1. **Riemann problem:** Compute $\mathbf{G}^*$ for any interface.

   a. *State evaluation.* Evaluate left and right states $\mathbf{V}_l$ and $\mathbf{V}_r$ of each interface.

   b. *Evaluate λ_l and λ_r.* using **Method 1**, **Method 2** or **Method 3** (refer to section 2.2.4) to ensure positivity and stable entropy.

   c. *Evaluate the intermediate wave speed u^* with (2.2.29) and intermediate pressure $\overline{p}$ with (2.2.32).*

2. **Compute numerical fluxes $\mathbf{G}^*$** with (2.2.56).

3. **Evaluate Δt** using (5.2.32).

4. **Update $\mathbf{V}_i^{n+1}$** using (2.2.55).

– **Out.** Data $\mathbf{V}_i^{n+1}$ for all cells.

2.2.6 One-dimensional Lagrangian numerical validation

Here, we present several classical test cases to assess the properties of the 1D Lagrangian scheme based on the approximate Riemann solver developed in section 2.2.3. The main purpose is to observe that the numerical method is stable and preserves the positivity of the specific volume and energy. For all test cases, the CFL is set to 0.9 and $\gamma = 7/5$, unless otherwise noticed. We start by performing a comparison between the three available methods for the determination of (λ_l, λ_r) on the Sod shock tube test case with simple waves and the Woodward-Collela blast wave test with interacting waves. After that, we run the Leblanc shock tube test case to verify the robustness of the scheme and the extreme double rarefaction test case to verify the positivity-preserving property of the scheme.

Sod shock tube We run the planar Sod shock tube problem to assess the ability of the methods to capture one-dimensional simple waves. The exact solution for this one-dimensional Riemann problem can be derived using for instance Toro (1999). The domain of the test case is $\Omega = [0, 1]$ with a discontinuity at $x_0 = 0.5$, and the initial condition of the test case are :

$$(\rho, u, p)_l = (1, 0, 1) \quad , \quad (\rho, u, p)_r = (0.125, 0, 0.1) \quad , \quad t_{final} = 0.2.$$

On the left of figure 2.4 we present the results for the density variable on a mesh refinement situation ($N = 100$ to 400 cells) and on the right the comparison of numerical density with different algorithms to determine the impedances are employed. The numerical solutions are compared to the exact solution (black line). Extremely few differences are observed for all three methods. This very same behavior is almost systematically observed for all test cases.

Collela-Woodward blastwave The Woodward-Collela blastwave test Woodward & Colella (1984) is a double shock tube case that simulates the interaction of simple waves. Two shock waves and two contact discontinuities develop and propagate towards the wall boundary conditions and reflect from them. These initial simple waves further interact creating a more complex flow pattern. The reference solution is obtained by a Lagrangian numerical scheme for 5000 cells (black line). The domain of the test case is $\Omega = [0, 1]$ with two discontinuities at $x_0 = 0.1$, and $x_1 = 0.9$ and the initial condition of the test

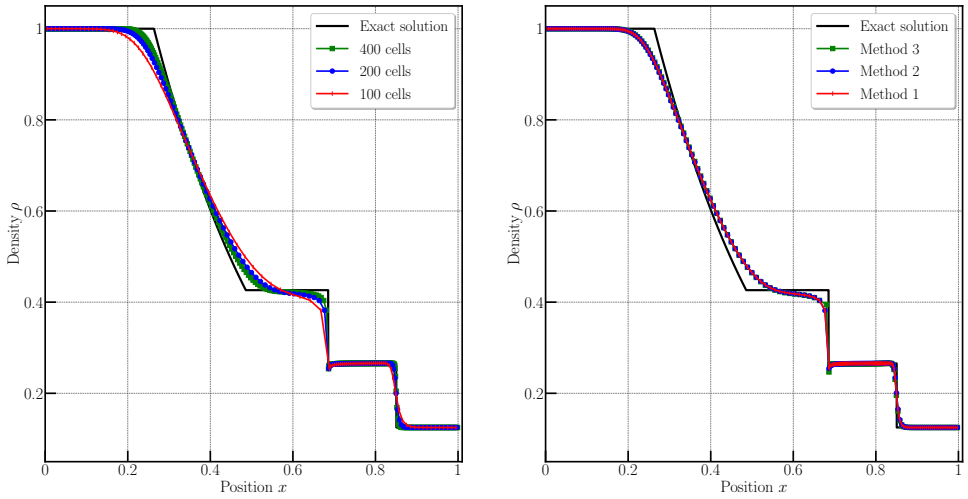


Figure 2.4: Sod shock tube problem — Numerical density of the 1st order Lagrangian scheme — Left : Grid convergence from $N = 100$ to 400 cells of the density variable over the full domain — Right : Comparison of different methods to determine the impedances λ_l, λ_r: Method 1 (magenta diamond), Method 2 (green square) and Method 3 (blue circle).

case are :

$$(\rho, u, p)_l = (1, 0, 1000) \quad , \quad (\rho, u, p)_m = (1, 0, 0.01) \quad , \quad (\rho, u, p)_r = (1, 0, 100),$$

and $t_{final} = 0.038$. In figure 2.5 we present the results for the density variable on a mesh refinement situation ($N = 100$ to 400 cells) using the first method to compute λ_l and λ_r. We then present the comparison of the numerical density obtained by different algorithms to determine the impedances in figure 2.6 on $N = 200$ cells. Figure 2.7 (a) to (c) shows a grayscale map of the normalized right impedance $\lambda_r/\rho_r a_r$ for method 1, 2 and 3 respectively, illustrating the evolution of the right impedance of each cell at each time step. We observe that for all methods, the right impedance appears to match the trajectory of the right moving shock and its subsequent interactions. Method 3 in particular show more patterns and this is due to the construction of this method that is more sensitive. Because the differences in the numerical solution remain small, we select Method 1 for the rest of this work.

Le Blanc shock tube This test is an extreme version of a shock tube (density jump is 10^3, pressure jump is 10^9) generating violent waves, which, however are still simple waves that can be exactly computed Toro (1999). The domain of the test case is $\Omega = [0, 9]$ with a discontinuity at $x_0 = 3$, and the initial condition of the test case are :

$$(\rho, u, p)_l = (1, 0, 2/3 \times 10^{-1}) \quad , \quad (\rho, u, p)_r = (10^{-3}, 0, 2/3 \times 10^{-10}) \quad , \quad t_{final} = 6.0,$$

and here $\gamma = 5/3$. The left and middle images in figure 2.8 present the results for the density and internal energy variables respectively, on a mesh refinement situation ($N = 900$ to 3600 cells) versus the exact solution (black line). The mesh refined solution

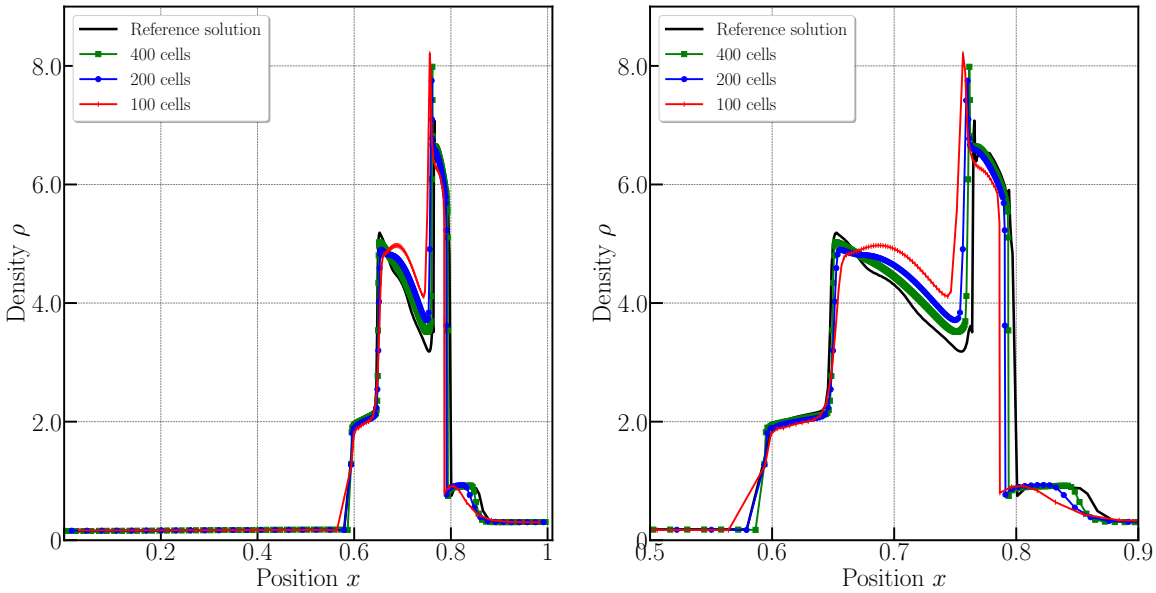


Figure 2.5: Woodward-Collela blastwave problem — 1st order Lagrangian scheme using the first method to determine the impedances — Left : Grid convergence from $N = 100$ to 400 cells of the numerical density over the full domain – Right : zoom on the central area

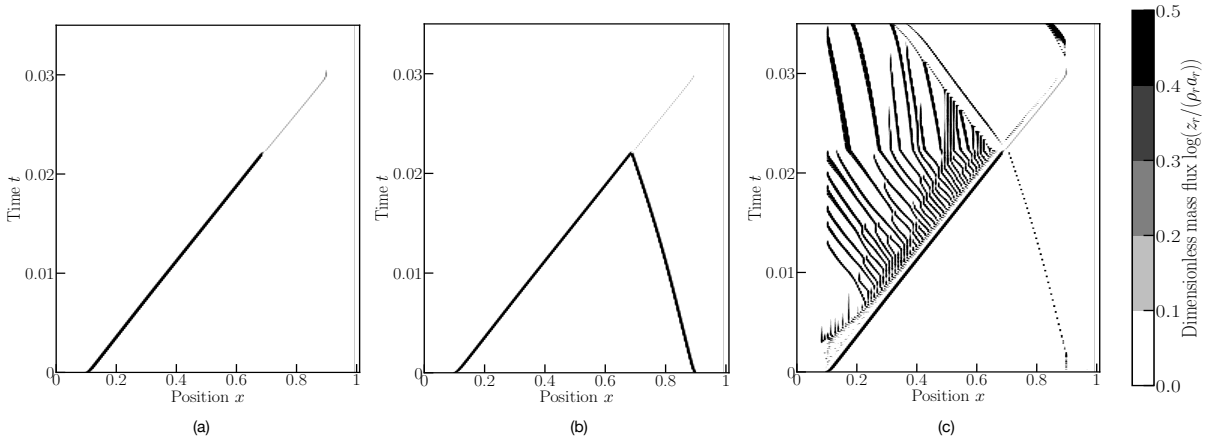


Figure 2.6: Woodward-Collela blastwave problem — 1st order Lagrangian scheme — Mesh $N = 200$ — Left : Comparison of different methods to determine the impedances λ over the full domain such as Method 1 represented in magenta diamond, Method 2 represented in green square and Method 3 represented in blue circle — Right : zoom on the central area.

seems to converge towards the exact solution with classical low convergence speed as is expected for this several Riemann problem. On the right is a grayscale map of the normalized right impedance $\lambda_r/(\rho_r a_r)$, illustrating the evolution of the right impedance for each cell at each time step. Once again we observe that the right impedance matches the trajectory of the right moving shock as expected for this problem involving only simple waves and only one shock wave.

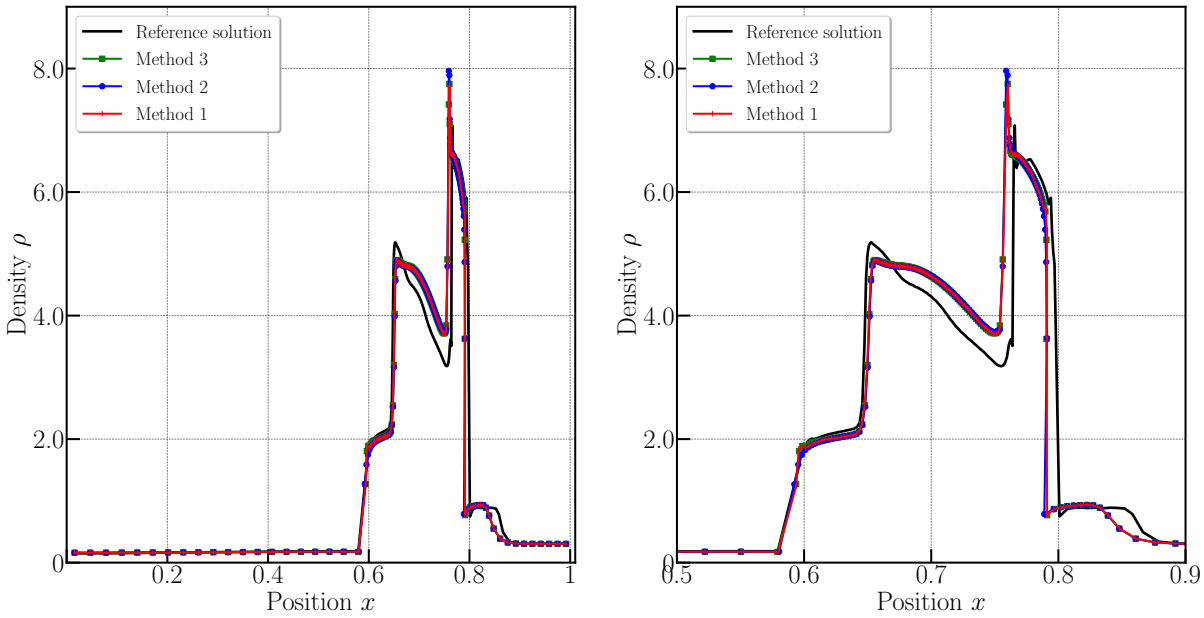


Figure 2.7: Woodward-Collela blastwave problem — 1st order Lagrangian scheme — Evolution of the normalized right impedance λ_r with each time step: (a) Method 1, (b) Method 2 and (c) Method 3.

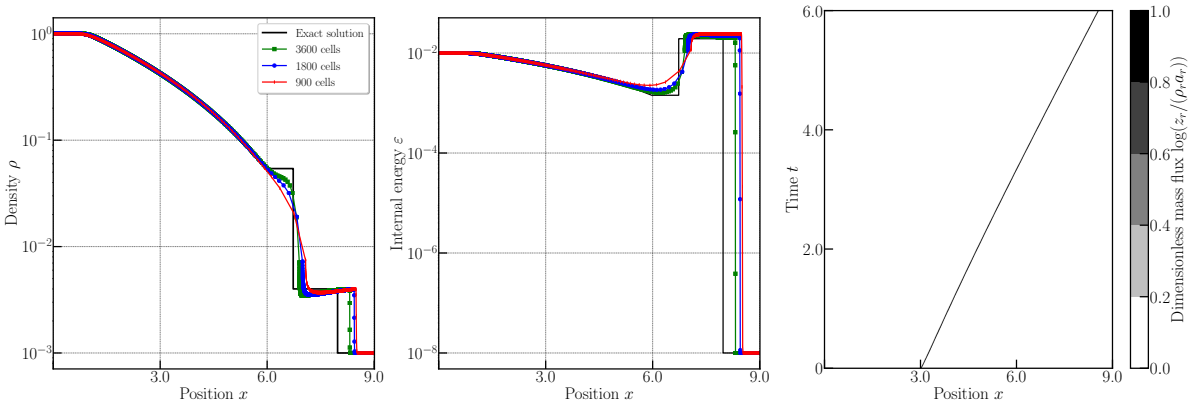


Figure 2.8: Le Blanc shock tube problem — 1st order Lagrangian numerical scheme with Method 1 (to determine the impedance λ) for $N = 900$, 1800 and 3600 cells — Left: Grid convergence of numerical density — Middle: Grid convergence for the numerical specific internal energy — Right: Evolution of the normalized right impedances λ_r for each time step.

Extreme double rarefaction This problem is inspired from the 123 problem from Toro (1999)'s book. The latter is one of benchmark tests presenting near vacuum state as it involves two rarefaction fans moving in opposite directions therefore emptying the central zone where a trivial steady contact discontinuity remains. $\Omega = [0, 2]$ with a discontinuity at $x_0 = 3$, and the initial condition of the test case are :

$$(\rho, u, p)_l = (1, -3.5, 0.4) \quad , \quad (\rho, u, p)_r = (1, 3.5, 0.4) \quad , \quad t_{final} = 0.15.$$

Outflow boundary conditions are considered. In this extreme configuration with $\gamma = 7/5$, the resolution of the Riemann leads to values of density $\rho^* \simeq 1.124 \times 10^{-6}$ and pressure $p^* \simeq 1.875 \times 10^{-9}$ and $\varepsilon^* \simeq 1.192 \times 10^{-3}$, see figure 2.9 at final time. In figure 2.9 we plot the results for successively refined grids. The numerical solutions are compared to the exact solution (black line). The spurious peak in the internal energy profile is a classical flaw of any numerical schemes due to entropy dissipation. Nevertheless we do not observe any lack of positivity for the numerical scheme, although the smallest numerical densities reach 8.65×10^{-3} for 800 cells. Moreover the numerical solution seems to converge towards the exact one.

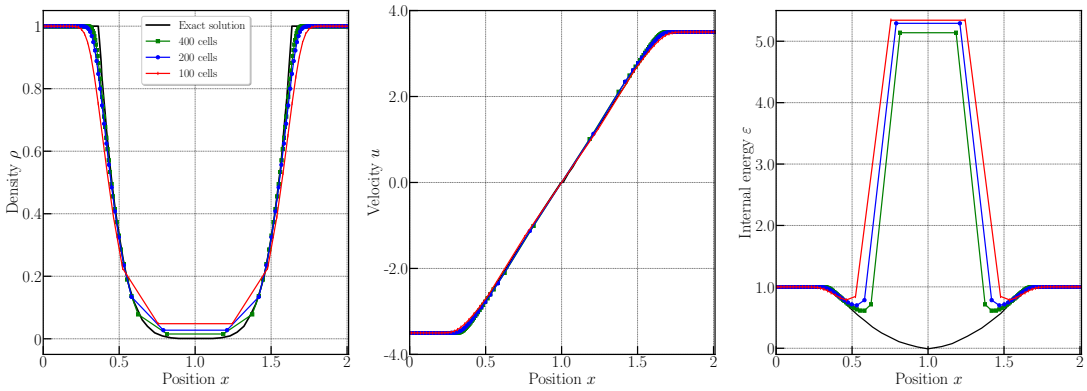


Figure 2.9: Extreme double rarefaction problem — 1st order Lagrangian scheme — Density (left), velocity (middle) and specific internal energy (right) — Grid convergence from $N = 100$ to 400 cells.

Now that the Lagrangian Godunov-type scheme is established and validated, in the next section, we present its Eulerian counter-part by using the Lagrangian solver as a building block. Before doing so, we recall the fundamental relation between the discontinuity speeds in the Lagrangian and Eulerian frameworks.

2.2.7 Fundamental relation between Lagrangian and Eulerian discontinuity speeds

The aim of this paragraph is to recall briefly the (RH) condition related to the system of conservation laws under consideration for both Lagrangian and Eulerian representations. The RH condition expresses the admissibility of piecewise discontinuous solutions with respect to the system of conservation laws written in weak (integral) form, refer for instance to Godlewski & Raviart (1996). In addition, we are going to exhibit the relationship between the Lagrangian and the Eulerian discontinuity speeds. This relationship will be of great interest, in what follows, for deriving the wave speeds of the Eulerian approximate Riemann solver in terms of its Lagrangian counterpart.

Let us consider the discontinuity curve defined in the (m, t) plane by the equation $m = \Xi(t)$ and the related piecewise discontinuous function

$$\mathbf{V}(X,t) = \begin{cases} \mathbf{V}_l & \text{if} \quad m < \Xi(t), \\ \mathbf{V}_r & \text{if} \quad m > \Xi(t). \end{cases} \tag{2.2.64}$$

This is a weak solution of the Lagrangian system of conservation laws (2.2.12) if and only if the Lagrangian RH condition holds true $-\dfrac{d\Xi}{dt}[\![\rho^0\mathbf{V}]\!]+[\![\mathbf{G}]\!]=\mathbf{0}$. Here, $[\![\mathbf{V}]\!]=\mathbf{V}_r-\mathbf{V}_l$ is the jump of $\mathbf{V}$ across the discontinuity. Thanks to (2.2.28), Lagrangian mass conservation RH condition reads $-\dfrac{d\Xi}{dt}[\![\rho^0]\!]=0$. Furthermore, recalling the identity $[\![ab]\!]=\langle a\rangle[\![b]\!]+\langle b\rangle[\![a]\!]$, where $\langle a\rangle=\dfrac{1}{2}(a_l+a_r)$, leads to rewrite the Lagrangian RH condition under the form

$$-\rho^0\frac{d\Xi}{dt}[\![\mathbf{V}]\!]+[\![\mathbf{G}]\!]=\mathbf{0}. \tag{2.2.65}$$

Let $\lambda=\rho^0\dfrac{d\Xi}{dt}$ be the mass flux swept by the Lagrangian discontinuity. For $\lambda=0$ it is a contact discontinuity, whereas for $\lambda>0$ (resp. $\lambda<0$), it is a right-going (resp. left-going) shock wave. With this notation, the Lagrangian RH condition reads

$$-\lambda[\![\mathbf{V}]\!]+[\![\mathbf{G}]\!]=\mathbf{0},\ \text{where } \mathbf{V}=(\tau,u,e)^t \text{ and } \mathbf{G}=(-u,p,pu)^t. \tag{2.2.66}$$

It is well known that the weak solutions satisfying the RH condition are not unique. Nevertheless the physically admissible solutions are selected supplementing the RH condition (2.2.66) by the RH entropy inequality

$$-\lambda[\![\eta]\!]\geq 0. \tag{2.2.67}$$

This RH entropy inequality simply expresses that the entropy of material particles is increasing through the shock wave consistently with the second law of thermodynamics since a shock wave is an irreversible thermodynamic process.

Now, we consider the Eulerian representation of the discontinuity curve defined in the (x,t) plane by the equation $x=\xi(t)$ and the piecewise discontinuous function

$$\mathbf{U}(x,t)=\begin{cases}\mathbf{U}_l & \text{if} \quad x<\xi(t),\\ \mathbf{U}_r & \text{if} \quad x>\xi(t).\end{cases} \tag{2.2.68}$$

This piecewise discontinuous function is a weak solution of the Eulerian system of conservation laws (2.2.1) if and only if the Eulerian RH condition holds true $-\dfrac{d\xi}{dt}[\![\mathbf{U}]\!]+[\![\mathbf{F}]\!]=\mathbf{0}$. Let $\Lambda=\dfrac{d\xi}{dt}$ denotes the Eulerian discontinuity speed, then the Eulerian RH condition turns into

$$-\Lambda[\![\mathbf{U}]\!]+[\![\mathbf{F}]\!]=\mathbf{0},\ \text{where } \mathbf{U}=(\rho,\rho u,\rho e)^t \text{ and } \mathbf{F}=(\rho u,\rho u^2+p,\rho ue+pu)^t. \tag{2.2.69}$$

The selection of the physically admissible weak solutions is enforced through the use of the Eulerian RH entropy inequality

$$-\Lambda[\![\rho\eta]\!]+[\![\rho\eta u]\!]\geq 0. \tag{2.2.70}$$

A weak solution of the gas dynamics equations is characterized by the RH condition, which can be written under Lagrangian representation (2.2.66) as well as under Eulerian representation (2.2.69). We relate the discontinuity speeds λ and Λ by comparing the first component of (2.2.66) and (2.2.69)

$$-\lambda[\![\tau]\!] - [\![u]\!] = 0, \tag{2.2.71a}$$
$$-\Lambda[\![\rho]\!] + [\![\rho u]\!] = 0. \tag{2.2.71b}$$

Before proceeding any further, we introduce some useful notations. Let α and ϕ be real numbers. We denote respectively (α_l, ϕ_l) and (α_r, ϕ_r) the values taken by (α, ϕ) on the left and the right side of the discontinuity. Assuming that the α-weights sum to one, *i.e.*, $\alpha_l + \alpha_r = 1$, the weighted averages of ϕ write

$$\overline{\phi} = \alpha_l\phi_l + \alpha_r\phi_r, \quad \underline{\phi} = \alpha_r\phi_l + \alpha_l\phi_r.$$

This allows us to express the jump of a product thanks to the identity

$$[\![\phi\psi]\!] = \overline{\phi}[\![\psi]\!] + \underline{\psi}[\![\phi]\!]. \tag{2.2.72}$$

Applying this identity to (τ, ρ) and observing that $\tau\rho = 1$ yields $\overline{\rho}[\![\tau]\!] = -\underline{\tau}[\![\rho]\!]$. Substituting this result into (2.2.71a) we get

$$\overline{\rho}[\![u]\!] = \lambda\underline{\tau}[\![\rho]\!]. \tag{2.2.73}$$

On the other hand, applying identity (2.2.72) to (2.2.71b) leads to

$$-\Lambda[\![\rho]\!] + \overline{\rho}[\![u]\!] + \underline{u}[\![\rho]\!] = 0.$$

Substituting the expression of the velocity jump (2.2.73) into the foregoing equation we finally arrive at

$$(-\Lambda + \underline{u} + \underline{\tau}\lambda)[\![\rho]\!] = 0.$$

This equation holds true regardless the mass density jump. Therefore, the term between parentheses must be equal to zero, which provides us the fundamental relation between the Eulerian and Lagrangian discontinuity speeds

$$\Lambda = \underline{\tau}\lambda + \underline{u}. \tag{2.2.74}$$

Utilizing the definition of the underline average and thanks to (2.2.71a) one can easily show that λ satisfies also

$$\Lambda = \tau_l\lambda + u_l = \tau_r\lambda + u_r. \tag{2.2.75}$$

Multiplying respectively the first equality by ρ_l and the second one by ρ_r leads to express the mass flux swept by the Lagrangian discontinuity, λ, in terms of its Eulerian velocity

$$\lambda = -\rho_l(u_l - \Lambda) = -\rho_r(u_r - \Lambda). \tag{2.2.76}$$

Let us point out that the mass flux swept by the Lagrangian discontinuity has exactly the opposite sign to the mass flux crossing the Eulerian discontinuity.

2.2.8 Simple approximate Riemann solver for Eulerian gas dynamics

The Riemann problem for gas dynamics written under Eulerian form reads

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \mathbf{0}, \qquad \mathbf{U}(x,0) = \begin{cases} \mathbf{U}_l & \text{if} \quad x < 0, \\ \mathbf{U}_r & \text{if} \quad x \geq 0. \end{cases} \tag{2.2.77}$$

For the sake of completeness, let us recall that $\mathbf{U} = (\rho, \rho u, \rho e)^t$ is the vector of conservative variables and $\mathbf{F}(\mathbf{U}) = (\rho u, \rho u^2 + p, \rho u e + p u)^t$ is the flux vector. The aim of this section is to derive a simple Riemann solver, $\mathbf{U}^{Eul}\left(\mathbf{U}_l, \mathbf{U}_r, \frac{x}{t}\right)$, approximating the solution of the Riemann problem (2.2.77). We acknowledge that there exists many approximate Riemann solvers for Eulerian gas dynamics, refer for instance to Toro (1999). Here, we present an Eulerian approximate Riemann solver which is deduced from its Lagrangian counterpart, $\mathbf{W}^{Lagr}$. This Riemann solver is of HLLC-type and the difference between the HLLC solver lies in the fact that the approximation of the Eulerian wave speeds are determined intrinsically without having to resort to more or less complicated approximations. This Eulerian approximate Riemann solver is constructed following the general methodology introduced in Gallice (2002*a*, 2003), which allows not only to derive Eulerian approximate Riemann solvers from their Lagrangian counterparts but also to transfer the positivity and entropy stability properties. Therefore, the present Eulerian approximate solver coincides with the one introduced in Gallice (2002*a*, 2003) in a more general and theoretical framework. Let us point out that its has been derived employing a different and more intuitive approach. This Eulerian simple solver, $\mathbf{W}^{Eul}$, consists of 4 constant states $(\mathbf{U}_l, \mathbf{U}_l^\star, \mathbf{U}_r^\star, \mathbf{U}_r)$ separated by the 3 discontinuities characterized by the speeds $(\Lambda_l, \Lambda_0, \Lambda_r)$

$$\mathbf{W}^{Eul}\left(\mathbf{U}_l, \mathbf{U}_r, \frac{x}{t}\right) = \begin{cases} \mathbf{U}_l & \text{if} \quad \dfrac{x}{t} < \Lambda_l, \\ \mathbf{U}_l^\star & \text{if} \quad \Lambda_l < \dfrac{x}{t} < \Lambda_0, \\ \mathbf{U}_r^\star & \text{if} \quad \Lambda_0 < \dfrac{x}{t} < \Lambda_r, \\ \mathbf{U}_r & \text{if} \quad \Lambda_r < \dfrac{x}{t}. \end{cases} \tag{2.2.78}$$

The structure of the simple Riemann solver, $\mathbf{W}^{Eul}$, is displayed in figure 2.10. Our starting point is the Lagrangian approximate Riemann solver, $\mathbf{W}^{Lagr}$, composed of the four states $(\mathbf{V}_l, \mathbf{V}_l^\star, \mathbf{V}_r^\star, \mathbf{V}_r)$, separated by the three discontinuities of speeds $(-\lambda_l, 0, \lambda_r)$. Here, we assume that the parameters λ_l, λ_r are such that the positivity of specific volumes, internal energies and entropy inequalities are taken for granted.

The construction of $\mathbf{W}^{Eul}$ is deduced directly from $\mathbf{W}^{Lagr}$. We have seen in sec-

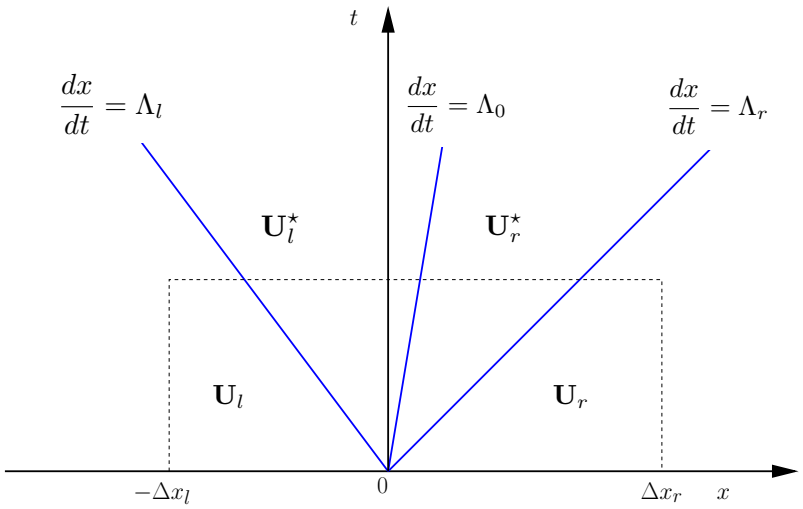


Figure 2.10: Simple Riemann solver for Eulerian gas dynamics.

tion 2.2.7 that the Lagrangian (λ) and the Eulerian (Λ) speeds of an admissible discontinuity are related by (2.2.28)

$$\Lambda = \tau_l \lambda + u_l = \tau_r \lambda + u_r,$$

where the superscripts l and r refer to the left and right sides of the discontinuity under consideration. Mimicking the foregoing formula, we shall express the speeds of the Eulerian discontinuities, $(\Lambda_l, \Lambda_0, \Lambda_r)$, in terms of the speeds of the Lagrangian discontinuities $(-\lambda_l, 0, \lambda_r)$. Replacing the mass flux swept by the Lagrangian discontinuity, λ, respectively by $-\lambda_l$, 0 and λ_r we obtain the expression of the Eulerian speeds

$$\Lambda_l = -\lambda_l \tau_l + u_l = -\lambda_l \tau_l^\star + u^\star, \tag{2.2.79a}$$
$$\Lambda_0 = u^\star, \tag{2.2.79b}$$
$$\Lambda_r = \lambda_r \tau_r^\star + u^\star = \lambda_r \tau_r + u_r. \tag{2.2.79c}$$

We observe that the wave speeds Λ_l, Λ_0 and Λ_r are fully determined by means of the knowledge of the intermediate states of the Lagrangian simple solver since by construction $\lambda_l(\tau_l^\star - \tau_l) - (u^\star - u_l) = 0$ and $\lambda_r(\tau_r^\star - \tau_r) + u^\star - u_r = 0$ holds true, refer to (2.2.26a) and (2.2.27a). Finally, comparing (2.2.79c), (2.2.79b) and (2.2.79c), it is clear that the the wave speeds ordering

$$\Lambda_l \leq \Lambda_0 \leq \Lambda_r, \tag{2.2.80}$$

holds true provided that the parameters λ_l, λ_r and the specific volumes $\tau_l^\star$, $\tau_r^\star$ are strictly positive. This is achieved when λ_l and λ_r are determined for instance by means of condition (2.2.36). Assuming that this condition holds true, we define the intermediate mass density $\rho_s^\star = \frac{1}{\tau_s^\star}$. Substituting it into (2.2.79a), (2.2.79b) and (2.2.79c) allows us to express the Lagrangian mass fluxes $(\lambda_l, 0, \lambda_r)$ in terms of the Eulerian wave speeds as

follows

$$\lambda_l = \rho_l(u_l - \Lambda_l) = \rho_l^\star(u^\star - \Lambda_l), \tag{2.2.81a}$$
$$0 = \rho_l^\star(u^\star - \Lambda_0) = \rho_r^\star(u^\star - \Lambda_0), \tag{2.2.81b}$$
$$\lambda_r = -\rho_r^\star(u^\star - \Lambda_r) = -\rho_r(u_r - \Lambda_r). \tag{2.2.81c}$$

We conclude that the mass flux swept by the Lagrangian discontinuities has exactly the opposite sign to the mass flux crossing the Eulerian discontinuities. This result has been already observed in section 2.2.7 wherein we have derived the Eulerian Rankine-Hugoniot conditions from the Lagrangian ones, refer to (2.2.76).

We are now in position to derive the intermediate states and fluxes of the Eulerian simple solver from the knowledge of the Lagrangian simple solver utilizing the expressions of the Lagrangian mass fluxes in terms of the Eulerian wave speeds. First, let us recall that the Lagrangian solver intermediate states and fluxes are determined by means of the jump relations (2.2.24)

$$\begin{aligned}\lambda_l(\mathbf{V}_l^\star - \mathbf{V}_l) + \mathbf{G}_l^\star - \mathbf{G}_l &= \mathbf{0},\\ 0(\mathbf{V}_r^\star - \mathbf{V}_l^\star) + \mathbf{G}_r^\star - \mathbf{G}_l^\star &= \mathbf{0},\\ -\lambda_r(\mathbf{V}_r - \mathbf{V}_r^\star) + \mathbf{G}_r - \mathbf{G}_r^\star &= \mathbf{0}.\end{aligned}$$

Here, $\mathbf{V}_s = (\tau_s, u_s, e_s)^t$, $\mathbf{V}_s^\star = (\tau_s^\star, u^\star, e_s^\star)^t$, $\mathbf{G}_s = (-u_s, p_s, p_s u_s)^t$ and $\mathbf{G}_s^\star = (-u^\star, p^\star, p^\star u^\star)^t$. The star quantities are fully determined by the systems of equations (2.2.26) and (2.2.27) once the parameters (λ_l, λ_r) are prescribed. Now, substituting the expressions of λ_l, 0 and λ_r given respectively by (2.2.81a), (2.2.81b) and (2.2.81c) into the foregoing equations leads to

$$-\Lambda_l(\rho_l^\star \mathbf{V}_l^\star - \rho_l \mathbf{V}_l) + \rho_l^\star u^\star \mathbf{V}_l^\star + \mathbf{G}_l^\star - (\rho_l u_l \mathbf{V}_l + \mathbf{G}_l) = \mathbf{0}, \tag{2.2.82a}$$
$$-\Lambda_0(\rho_r^\star \mathbf{V}_r^\star - \rho_l^\star \mathbf{V}_l^\star) + \rho_r^\star u^\star \mathbf{V}_r^\star + \mathbf{G}_r^\star - (\rho_l^\star u^\star \mathbf{V}_l^\star + \mathbf{G}_l^\star) = \mathbf{0}, \tag{2.2.82b}$$
$$-\Lambda_r(\rho_r \mathbf{V}_r - \rho_r^\star \mathbf{V}_r^\star) + \rho_r u_r \mathbf{V}_r + \mathbf{G}_r - (\rho_r^\star u^\star \mathbf{V}_r^\star + \mathbf{G}_r^\star) = \mathbf{0}. \tag{2.2.82c}$$

These vectorial equations might be interpreted as jump equations across the Eulerian discontinuities of speeds $(\Lambda_l, \Lambda_0, \Lambda_r)$. However, one readily observes that the first component of the 3 foregoing equations is trivial. Thus, this first component cannot contribute to the definition of the intermediate states and fluxes of the approximate Riemann solver. On the other hand, rearranging (2.2.81a), (2.2.81b) and (2.2.81c) we arrive at

$$\begin{aligned}-\Lambda_l(\rho_l^\star - \rho_l) + \rho_l^\star u^\star - \rho_l u_l &= 0,\\ -\Lambda_0(\rho_r^\star - \rho_l^\star) + \rho_r^\star u^\star - \rho_l^\star u^\star &= 0,\\ -\Lambda_r(\rho_r - \rho_r^\star) + \rho_r u_r - \rho_r^\star u^\star &= 0.\end{aligned}$$

This is nothing but the jump relations across the discontinuities of speeds $(\Lambda_l, \Lambda_0, \Lambda_r)$ related to the Eulerian mass conservation equation. Finally, gathering the second and

the third components of system (2.2.82) with the foregoing system yields

$$-\Lambda_l(\mathbf{U}_l^\star - \mathbf{U}_l) + \mathbf{F}_l^\star - \mathbf{F}_l = \mathbf{0}, \tag{2.2.83a}$$
$$-\Lambda_0(\mathbf{U}_r^\star - \mathbf{U}_l^\star) + \mathbf{F}_r^\star - \mathbf{F}_l^\star = \mathbf{0}, \tag{2.2.83b}$$
$$-\Lambda_r(\mathbf{U}_r - \mathbf{U}_r^\star) + \mathbf{F}_r - \mathbf{F}_r^\star = \mathbf{0}, \tag{2.2.83c}$$

where on the one hand

$$\mathbf{U}_s = (\rho_s, \rho_s u_s, \rho_s e_s)^t, \qquad \mathbf{F}_s = (\rho_s u_s, \rho_s u_s^2 + p_s, \rho_s u_s e_s + p_s u_s)^t,$$

and on the other hand

$$\mathbf{U}_s^\star = (\rho_s^\star, \rho_s^\star u^\star, \rho_s^\star e_s^\star)^t, \qquad \mathbf{F}_s^\star = (\rho_s^\star u^\star, \rho_s^\star (u^\star)^2 + p^\star, \rho_s^\star u^\star e_s^\star + p^\star u^\star)^t, \qquad \text{for} \quad s = l, r. \tag{2.2.84}$$

The Eulerian intermediate states and fluxes are fully determined from the Lagrangian Riemann solver. Moreover, the second and third components of the Eulerian intermediate states and fluxes might be formally related to their Lagrangian counterparts by $\mathbf{U}_s^\star = \rho_s^\star \mathbf{V}_s^\star$ and $\mathbf{F}_s^\star = \rho_s^\star u^\star \mathbf{V}_s^\star + \mathbf{G}_s^\star$ for $s = l, r$. Summing equations (2.2.83a), (2.2.83b) (2.2.83c) leads to

$$\mathbf{F}_r - \mathbf{F}_l = \Lambda_l(\mathbf{U}_l^\star - \mathbf{U}_l) + \Lambda_0(\mathbf{U}_r^\star - \mathbf{U}_l^\star) + \Lambda_r(\mathbf{U}_r - \mathbf{U}_r^\star), \tag{2.2.85}$$

which shows the consistency of the approximate Riemann solver $\mathbf{W}^{Eul}$ with the integral form of the Riemann problem (2.2.77). Indeed, equation (2.2.85) is nothing but the integration of (2.2.77) over the space-time interval $[-\Delta x_l, \Delta x_r] \times [0, \Delta t]$. The interface flux, $\mathbf{F}^\star$, at $\frac{x}{t} = 0$ is expressed in terms of the wave speeds as follows

$$\mathbf{F}^\star = \begin{cases} \mathbf{F}_l & \text{if} \quad 0 < \Lambda_l < \Lambda_0 < \Lambda_r, \\ \mathbf{F}_l^\star & \text{if} \quad \Lambda_l < 0 < \Lambda_0 < \Lambda_r, \\ \mathbf{F}_r^\star & \text{if} \quad \Lambda_l < \Lambda_0 < 0 < \Lambda_r, \\ \mathbf{F}_r & \text{if} \quad \Lambda_l < \Lambda_0 < \Lambda_r < 0. \end{cases} \tag{2.2.86}$$

This expression can be rewritten under the compact form

$$\mathbf{F}^\star = \frac{1}{2}(\mathbf{F}_l + \mathbf{F}_r) - \frac{|\,\Lambda_l\,|}{2}(\mathbf{U}_l^\star - \mathbf{U}_l) - \frac{|\,\Lambda_0\,|}{2}(\mathbf{U}_r^\star - \mathbf{U}_l^\star) - \frac{|\,\Lambda_r\,|}{2}(\mathbf{U}_r - \mathbf{U}_r^\star). \tag{2.2.87}$$

The numerical dissipation is governed by the wave velocities Λ_l, Λ_0 and Λ_r.

Remark 2.2.1 (Similarity with HLLC approximate Riemann solver)**.** *The present Eulerian approximate Riemann solver is very similar to the famous HLLC approximate Riemann solver introduced by Toro et al. (1994) and his co-authors in the seminal work and also presented in details in Toro (1999)'s book. First, the structure of the approximate Riemann solver composed of 4 states separated by 3 waves is exactly the same. Second, the jump relations (2.2.83) satisfied by the intermediate states and fluxes across each wave are also identical to those satisfied by the states and fluxes of the HLLC approximate Rie-*

mann solver. Therefore, the structure of the resulting interface flux (2.2.87) *is the same as the one of the HLLC solver. On the other hand, our Eulerian approximate Riemann solver has been entirely deduced from its Lagrangian counterpart utilizing the relationship* (2.2.79) *between Lagrangian and Eulerian wave speeds. This approach provides us a self-consistent computation of the Eulerian wave speeds, which are ordered by construction. This also allows the transfer of the properties (positivity preserving and entropy stability) of the Lagrangian approximate Riemann solver to its Eulerian counterpart. Let us point out that the wave speeds estimate in the context of the HLLC approximate Riemann solver has been and still remains an active topic of research, refer for instance to Batten* et al. *(1997); Toro* et al. *(2020).*

It remains to assess the consistency of the Eulerian approximate Riemann solver, $\mathbf{W}^{Eul}$, with the integral form of the Eulerian entropy inequality

$$\frac{\partial(\rho\eta)}{\partial t} + \frac{\partial(\rho\eta u)}{\partial x} \geq 0.$$

First, we assume that the Lagrangian approximate Riemann solver, $\mathbf{W}^{Lagr}$, is consistent with the integral form of the Lagrangian entropy inequality. This might be obtained prescribing parameters λ_l, λ_r to satisfy (2.2.49). Let us compute the entropy production across each wave by transforming its Lagrangian expression into its Eulerian counterpart simply by substituting the expressions of λ_l and λ_r in terms of Λ_l and Λ_r given by (2.2.81a) and (2.2.81c). For the left-going wave we arrive at

$$\lambda_l(\eta_l^\star - \eta_l) = \rho_l^\star(u^\star - \Lambda_l)\eta_l^\star - \rho_l(u_l - \Lambda_l)\eta_l = -\Lambda_l(\rho_l^\star\eta_l^\star - \rho_l\eta_l) + \rho_l^\star\eta_l^\star u^\star - \rho_l\eta_l u_l, \quad (2.2.88)$$

whereas for the right-going wave we obtain

$$\lambda_r(\eta_r^\star - \eta_r) = -\rho_r^\star(u^\star - \Lambda_r)\eta_r^\star + \rho_r(u_r - \Lambda_r)\eta_r = -\Lambda_r(\rho_r\eta_r - \rho_r^\star\eta_r^\star) + \rho_r\eta_r u_r - \rho_r^\star\eta_r^\star u^\star. \quad (2.2.89)$$

Summing these two equations leads to

$$\lambda_l(\eta_l^\star - \eta_l) + \lambda_r(\eta_r^\star - \eta_r) = -\Lambda_l(\rho_l^\star\eta_l^\star - \rho_l\eta_l) - \Lambda_0(\rho_r^\star\eta_r^\star - \rho_l^\star\eta_l^\star) - \Lambda_r(\rho_r\eta_r - \rho_r^\star\eta_r^\star) + \rho_r\eta_r u_r - \rho_l\eta_l u_l, \quad (2.2.90)$$

since $u^\star = \Lambda_0$ thanks to (2.2.81b). Finally, we claim that the Eulerian approximate Riemann solver is consistent with the integral form of the Eulerian entropy inequality if and only if the Lagrangian approximate Riemann solver is consistent with the integral form of the Lagrangian entropy inequality, and, there holds

$$-\Lambda_l(\rho_l^\star\eta_l^\star - \rho_l\eta_l) - \Lambda_0(\rho_r^\star\eta_r^\star - \rho_l^\star\eta_l^\star) - \Lambda_r(\rho_r\eta_r - \rho_r^\star\eta_r^\star) + \rho_r\eta_r u_r - \rho_l\eta_l u_l \geq 0. \quad (2.2.91)$$

This concludes the design of a Riemann solver in Eulerian framework derived from its Lagrangian counter-part. Let us now design their associated Godunov numerical schemes.

2.2.9 Godunov-type scheme for Eulerian gas dynamics

The aim of this section is to provide a first-order space and time Finite Volume discretization of the one-dimensional gas dynamics equations written under Eulerian formalism as a derived version from the Lagrangian one. The resulting numerical method is a Godunov-type scheme since the updated cell-averaged value are computed by combining the cell-interface approximate Riemann solvers that have been constructed in section 2.2.8.

2.2.9.a Governing equation and notation

The one-dimensional Eulerian gas dynamics equations read

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = \mathbf{0},$$

where $\mathbf{U} = (\rho, \rho u, \rho e)^t$, $\mathbf{F}(\mathbf{U}) = (\rho u, \rho u^2 + p, \rho e u + p u)^t$, where $e = \varepsilon + \frac{1}{2}u^2$. The set of physically admissible states for this system of conservation laws writes

$$\mathcal{A}_{\mathrm{E}} = \left\{ \mathbf{U} = (\rho, \rho u, \rho e)^t,\ \rho > 0,\ \varepsilon > 0 \right\}. \tag{2.2.92}$$

The computational domain $\omega = [x_{\min}, x_{\max}]$ is partitioned into n_c non overlapping cells $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, where $x_{i+\frac{1}{2}}$ denotes the fixed position of a generic node. The cell volume is $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$. At time t^n, we assume that the solution of the foregoing system of conservation laws is piecewise constant over each cell and defined by the cell-averaged value

$$\mathbf{U}_i^n = \frac{1}{\Delta x_i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{U}(x, t^n)\, \mathrm{d}x. \tag{2.2.93}$$

2.2.9.b Godunov-type scheme

We define the discrete solution of the system of conservation laws at time $t^{n+1} = t^n + \Delta t$ in terms of the piecewise solution at time t^n combining the approximate Riemann solvers located at $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$, refer to figure 2.11. The time step, Δt, has been chosen sufficiently small to ensure that the rightgoing wave emanating from $x_{i-\frac{1}{2}}$ does not interact with the leftgoing one emanating from $x_{i+\frac{1}{2}}$. Then, the updated cell-averaged value writes

$$\Delta x_i\ \mathbf{U}_i^{n+1} = \int_{x_{i-\frac{1}{2}}}^{x_i} \mathbf{W}^{Eul}\left(\mathbf{U}_{i-1}^n, \mathbf{U}_i^n, \frac{x - x_{i-\frac{1}{2}}}{\Delta t}\right) \mathrm{d}x + \int_{x_i}^{x_{i+\frac{1}{2}}} \mathbf{W}^{Eul}\left(\mathbf{U}_i^n, \mathbf{U}_{i+1}^n, \frac{x - x_{i+\frac{1}{2}}}{\Delta t}\right) \mathrm{d}x. \tag{2.2.94}$$

The first term at the right-hand side corresponds to the integral of the approximate Riemann solver at $x_{i-\frac{1}{2}}$ over $[x_{i-\frac{1}{2}}, x_i]$ where $x_i = \frac{1}{2}(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}})$, whereas the second term corresponds to the integral of the approximate Riemann solver at $x_{i+\frac{1}{2}}$ over $[x_i, x_{i+\frac{1}{2}}]$.

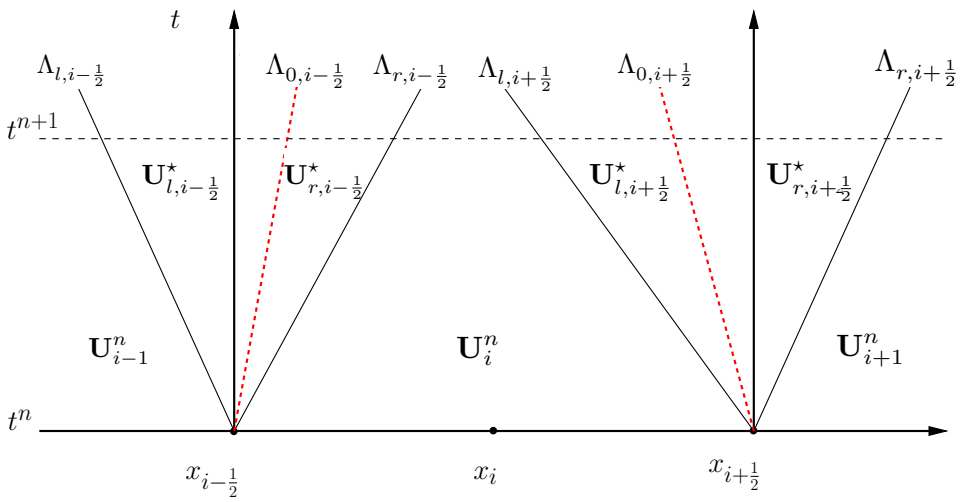


Figure 2.11: Eulerian approximate Riemann solvers at cell interfaces.

After some algebra we arrive at the explicit expressions of these integrals

$$
\begin{aligned}
\int_{x_{i-\frac{1}{2}}}^{x_i} \mathbf{W}^{Eul}\left(\mathbf{U}^n_{i-1}, \mathbf{U}^n_i, \frac{x-x_{i-\frac{1}{2}}}{\Delta t}\right) \mathrm{d}x &= \Lambda^+_{l,i-\frac{1}{2}} \Delta t\, \mathbf{U}^n_{i-1} \\
&+ (\Lambda^+_{0,i-\frac{1}{2}} - \Lambda^+_{l,i-\frac{1}{2}}) \Delta t\, \mathbf{U}^{\star}_{l,i-\frac{1}{2}} \\
&+ (\Lambda^+_{r,i-\frac{1}{2}} - \Lambda^+_{0,i-\frac{1}{2}}) \Delta t\, \mathbf{U}^{\star}_{r,i-\frac{1}{2}} \\
&+ (x_i - x_{i-\frac{1}{2}} - \Lambda^+_{r,i-\frac{1}{2}} \Delta t) \mathbf{U}^n_i, \\
\int_{x_i}^{x_{i+\frac{1}{2}}} \mathbf{W}^{Eul}\left(\mathbf{U}^n_i, \mathbf{U}^n_{i+1}, \frac{x-x_{i+\frac{1}{2}}}{\Delta t}\right) \mathrm{d}x &= (x_{i+\frac{1}{2}} - x_i - \Lambda^-_{l,i+\frac{1}{2}} \Delta t) \mathbf{U}^n_i \\
&+ (\Lambda^-_{l,i+\frac{1}{2}} - \Lambda^-_{0,i+\frac{1}{2}}) \Delta t\, \mathbf{U}^{\star}_{l,i+\frac{1}{2}} \\
&+ (\Lambda^-_{0,i+\frac{1}{2}} - \Lambda^-_{r,i+\frac{1}{2}}) \Delta t\, \mathbf{U}^{\star}_{r,i+\frac{1}{2}} \\
&+ \Lambda^-_{r,i+\frac{1}{2}} \Delta t\, \mathbf{U}^n_{i+1}.
\end{aligned}
\tag{2.2.95}
$$

Here, $\Lambda_{l,i\pm\frac{1}{2}}$, $\Lambda^{\star}_{i\pm\frac{1}{2}}$ and $\Lambda_{r,i\pm\frac{1}{2}}$ are the waves speeds of the Eulerian approximate Riemann solver at $x_{i+\frac{1}{2}}$. These Eulerian speeds are computed from the underlying Lagrangian approximate Riemann solver according to respectively (2.2.79a), (2.2.79b) and (2.2.79c). In addition, $x^+ = \frac{1}{2}(\mid x \mid +x)$ denotes the positive part of the real x, whereas $x^- = \frac{1}{2}(\mid x \mid -x)$ denotes its negative one. These are non-negative functions of x respectively increasing and decreasing. We observe that the positive part (resp. negative part) of the waves emanating from $x_{i-\frac{1}{2}}$ (resp. $x_{i+\frac{1}{2}}$) contributes to update the solution at time t^{n+1}. Summing the foregoing results and collecting the terms in factor of the intermediate states, we finally obtain

$$
\mathbf{U}^{n+1}_i = \frac{\Delta t}{\Delta x_i} \Lambda^+_{l,i-\frac{1}{2}} \mathbf{U}^n_{i-1} \tag{2.2.96}
$$

$$+\frac{\Delta t}{\Delta x_i}(\Lambda^+_{0,i-\frac{1}{2}}\Lambda^+_{l,i-\frac{1}{2}})\mathbf{U}^\star_{l,i-\frac{1}{2}} \tag{2.2.97}$$

$$+\frac{\Delta t}{\Delta x_i}(\Lambda^+_{r,i-\frac{1}{2}}-\Lambda^+_{0,i-\frac{1}{2}})\mathbf{U}^\star_{r,i-\frac{1}{2}} \tag{2.2.98}$$

$$+\left[1-\frac{\Delta t}{\Delta x_i}(\Lambda^+_{r,i-\frac{1}{2}}+\Lambda^-_{l,i+\frac{1}{2}})\right]\mathbf{U}^n_i$$

$$+\frac{\Delta t}{\Delta x_i}(\Lambda^-_{l,i+\frac{1}{2}}-\Lambda^-_{0,i+\frac{1}{2}})\mathbf{U}^\star_{l,i+\frac{1}{2}} \tag{2.2.99}$$

$$+\frac{\Delta t}{\Delta x_i}(\Lambda^-_{0,i+\frac{1}{2}}-\Lambda^-_{r,i+\frac{1}{2}})\mathbf{U}^\star_{r,i+\frac{1}{2}} \tag{2.2.100}$$

$$+\frac{\Delta t}{\Delta x_i}\Lambda^-_{r,i+\frac{1}{2}}\mathbf{U}^n_{i+1}.$$

The underlying approximate Riemann solver, $\mathbf{W}^{Eul}$, being consistent with the integral form of the system of conservation laws, the foregoing numerical scheme is of Godunov-type and can be equivalently written under the flux form

$$\Delta x_i\,(\mathbf{U}^{n+1}_i-\mathbf{U}^n_i)+\Delta t\,(\mathbf{F}^\star_{i+\frac{1}{2}}-\mathbf{F}^\star_{i-\frac{1}{2}})=\mathbf{0}, \tag{2.2.101}$$

where $\mathbf{F}^\star_{i+\frac{1}{2}}$ is the numerical flux at interface $x_{i+\frac{1}{2}}$ resulting from the Eulerian approximate Riemann solver. According to (2.2.87) the expression of the interface flux, with an obvious notation adaptation, reads

$$\begin{aligned}\mathbf{F}^\star_{i+\frac{1}{2}}=\frac{1}{2}(\mathbf{F}^n_i+\mathbf{F}^n_{i+1}) &- \frac{|\,\Lambda_{l,i+\frac{1}{2}}\,|}{2}(\mathbf{U}^\star_{l,i+\frac{1}{2}}-\mathbf{U}^n_i)\\ &- \frac{|\,\Lambda_{0,i+\frac{1}{2}}\,|}{2}(\mathbf{U}^\star_{r,i+\frac{1}{2}}-\mathbf{U}^\star_{l,i+\frac{1}{2}})\\ &- \frac{|\,\Lambda_{r,i+\frac{1}{2}}\,|}{2}(\mathbf{U}^n_{i+1}-\mathbf{U}^\star_{r,i+\frac{1}{2}}).\end{aligned} \tag{2.2.102}$$

2.2.9.c Time step monitoring

We shall compute the time step, Δt, to ensure that the updated cell-averaged value, $\mathbf{U}^{n+1}_i$, is a convex combination of $\mathbf{U}^n_{i-1}$, $\mathbf{U}^\star_{l,i-\frac{1}{2}}$, $\mathbf{U}^\star_{r,i-\frac{1}{2}}$, $\mathbf{U}^n_i$, $\mathbf{U}^\star_{l,i+\frac{1}{2}}$, $\mathbf{U}^\star_{r,i+\frac{1}{2}}$ and $\mathbf{U}^n_{i+1}$. This amounts to investigate the positivity of the coefficients of the linear combination present at the right-hand side of (2.2.96). Firstly, the positive part and the negative part of the wave speeds are non-negative. Secondly, the wave speeds of the approximate Riemann solvers at $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$ are ordered according to $\Lambda_{l,i\pm\frac{1}{2}}\leq\Lambda_{0,i\pm\frac{1}{2}}\leq\Lambda_{r,i\pm\frac{1}{2}}$. Bearing this in mind and recalling that $x\mapsto x^+$ (resp. $x\mapsto x^-$) is an increasing (resp. decreasing) function yields

$$\Lambda^+_{l,i\pm\frac{1}{2}}\leq\Lambda^+_{0,i\pm\frac{1}{2}}\leq\Lambda^+_{r,i\pm\frac{1}{2}},\quad\text{and}\quad\Lambda^-_{l,i\pm\frac{1}{2}}\geq\Lambda^-_{0,i\pm\frac{1}{2}}\geq\Lambda^-_{r,i\pm\frac{1}{2}}.$$

Finally, all the coefficients at the right-hand side of (2.2.96) are unconditionally positive except the one in factor of $\mathbf{U}_i^n$. This latter coefficient is positive if the time step Δt satisfies

$$\Delta t \leq \frac{\Delta x_i}{\Lambda^+_{r,i-\frac{1}{2}} + \Lambda^-_{l,i+\frac{1}{2}}}. \tag{2.2.103}$$

Therefore, $\mathbf{U}_i^{n+1}$ is a convex combination of $\mathbf{U}_i^n$ and the intermediate states of the approximate Riemann solvers located at $x_{i\pm\frac{1}{2}}$ provided that Δt satisfies condition (2.2.103), which can be viewed as a CFL-like condition. Developing the expression of the Eulerian wave speeds in terms of the underlying Lagrangian mass fluxes, *i.e.*, $\Lambda_{r,i-\frac{1}{2}} = u_i^n + \dfrac{\lambda_{r,i-\frac{1}{2}}}{\rho_i^n}$ and $\Lambda_{l,i+\frac{1}{2}} = u_i^n - \dfrac{\lambda_{l,i+\frac{1}{2}}}{\rho_i^n}$, we get

$$\begin{aligned}
\Lambda^+_{r,i-\frac{1}{2}} + \Lambda^-_{l,i+\frac{1}{2}} &= \tfrac{1}{2}\left(\mid u_i^n - \frac{\lambda_{l,i+\frac{1}{2}}}{\rho_i^n} \mid + \mid u_i^n + \frac{\lambda_{r,i-\frac{1}{2}}}{\rho_i^n} \mid + \frac{\lambda_{r,i-\frac{1}{2}}}{\rho_i^n} + \frac{\lambda_{l,i+\frac{1}{2}}}{\rho_i^n} \right) \\
&\leq \mid u_i^n \mid + \frac{\lambda_{r,i-\frac{1}{2}} + \lambda_{l,i+\frac{1}{2}}}{\rho_i^n}.
\end{aligned}$$

Employing this bound leads to replace (2.2.103) by the more explicit condition for all i:

$$\Delta t \leq \frac{\Delta x_i}{\mid u_i^n \mid + \dfrac{\lambda_{r,i-\frac{1}{2}} + \lambda_{l,i+\frac{1}{2}}}{\rho_i^n}}. \tag{2.2.104}$$

Therefore, the time-step is is practically computed as

$$\Delta t = CFL \frac{\Delta x_i}{\max_i \left(\mid u_i^n \mid + \dfrac{\lambda_{r,i-\frac{1}{2}} + \lambda_{l,i+\frac{1}{2}}}{\rho_i^n} \right)}. \tag{2.2.105}$$

In the following we focus on the positivity and entropy issues.

2.2.9.d Positivity of mass density and specific internal energy

The Finite Volume scheme (2.2.101) for Eulerian gas dynamics has been constructed through the use of the Eulerian approximate Riemann solver described in section 2.2.8, which inherits all the properties from its Lagrangian counterpart. More precisely, provided that the parameters (λ_l, λ_r) of the Lagrangian Riemann solver satisfies condition (2.2.36) the positivity of specific volume and specific internal energy is ensured. Therefore, the waves speeds of the Riemann solvers are ordered according to (2.2.80) and the positivity of mass density and specific energy for the intermediate states is also ensured. Under the time step condition (2.2.103), $\mathbf{U}_i^{n+1}$ is a convex combination of $\mathbf{U}_i^n$ and the intermediate states of the approximate Riemann solvers located at the cell interfaces, refer to (2.2.96). Therefore, if $\rho_i^n > 0$ and $\varepsilon_i^n > 0$ then $\rho_i^{n+1} > 0$ and $\varepsilon_i^{n+1} > 0$. This

shows that Finite Volume scheme (2.2.101) preserves the positivity of mass density and specific internal energy under the time step condition (2.2.103).

2.2.9.e Entropy inequality

In this paragraph, we aim at constructing the discrete entropy inequality satisfied by the Godunov-type Finite Volume scheme (2.2.101) following Chalons (2010). We shall start by recalling the expression of the updated cell-averaged value $\mathbf{U}_i^{n+1}$

$$\begin{aligned}\mathbf{U}_i^{n+1} =& \frac{\Delta t}{\Delta x_i}\Lambda^+_{l,i-\frac{1}{2}}\mathbf{U}^n_{i-1} + \frac{\Delta t}{\Delta x_i}(\Lambda^+_{0,i-\frac{1}{2}} - \Lambda^+_{l,i-\frac{1}{2}})\mathbf{U}^\star_{l,i-\frac{1}{2}} + \frac{\Delta t}{\Delta x_i}(\Lambda^+_{r,i-\frac{1}{2}} - \Lambda^+_{0,i-\frac{1}{2}})\mathbf{U}^\star_{r,i-\frac{1}{2}}\\ &+\left[1 - \frac{\Delta t}{\Delta x_i}(\Lambda^+_{r,i-\frac{1}{2}} + \Lambda^-_{l,i+\frac{1}{2}})\right]\mathbf{U}^n_i\\ &+\frac{\Delta t}{\Delta x_i}(\Lambda^-_{l,i+\frac{1}{2}} - \Lambda^{\star,-}_{i+\frac{1}{2}})\mathbf{U}^\star_{l,i+\frac{1}{2}} + \frac{\Delta t}{\Delta x_i}(\Lambda^{\star,-}_{i+\frac{1}{2}} - \Lambda^-_{r,i+\frac{1}{2}})\mathbf{U}^\star_{r,i+\frac{1}{2}} + \frac{\Delta t}{\Delta x_i}\Lambda^-_{r,i+\frac{1}{2}}\mathbf{U}^n_{i+1}.\end{aligned}$$

Let us recall that under time step condition (2.2.103), $\mathbf{U}_i^{n+1}$ is a convex combination of $\mathbf{U}^n_{i-1}$, $\mathbf{U}^\star_{l,i-\frac{1}{2}}$, $\mathbf{U}^\star_{r,i-\frac{1}{2}}$, $\mathbf{U}^n_i$, $\mathbf{U}^\star_{l,i+\frac{1}{2}}$, $\mathbf{U}^\star_{r,i+\frac{1}{2}}$ and $\mathbf{U}^n_{i+1}$. The strict concavity of $\mathbf{U}\mapsto(\rho\eta)(\mathbf{U})$ with respect to $\mathbf{U}$, refer to Godlewski & Raviart (1996), and Jensen's inequality yield the following inequality for the cell-averaged entropy time-increment

$$\begin{aligned}(\rho\eta)(\mathbf{U}_i^{n+1}) - (\rho\eta)(\mathbf{U}_i^n) \geq& -\frac{\Delta t}{\Delta x_i}\left\{\Lambda^+_{l,i-\frac{1}{2}}\left[(\rho\eta)(\mathbf{U}^\star_{l,i-\frac{1}{2}}) - (\rho\eta)(\mathbf{U}^n_{i-1})\right]\right\}\\ -\frac{\Delta t}{\Delta x_i}&\left\{\Lambda^+_{0,i-\frac{1}{2}}\left[(\rho\eta)(\mathbf{U}^\star_{r,i-\frac{1}{2}}) - (\rho\eta)(\mathbf{U}^\star_{l,i-\frac{1}{2}})\right] + \Lambda^+_{r,i-\frac{1}{2}}\left[(\rho\eta)(\mathbf{U}^n_i) - (\rho\eta)(\mathbf{U}^\star_{r,i-\frac{1}{2}})\right]\right\}\\ &+\frac{\Delta t}{\Delta x_i}\left\{\Lambda^-_{l,i+\frac{1}{2}}\left[(\rho\eta)(\mathbf{U}^\star_{l,i+\frac{1}{2}}) - (\rho\eta)(\mathbf{U}^n_i)\right]\right\}\\ +\frac{\Delta t}{\Delta x_i}&\left\{\Lambda^-_{0,i+\frac{1}{2}}\left[(\rho\eta)(\mathbf{U}^\star_{r,i+\frac{1}{2}}) - (\rho\eta)(\mathbf{U}^\star_{l,i+\frac{1}{2}})\right] + \Lambda^-_{r,i+\frac{1}{2}}\left[(\rho\eta)(\mathbf{U}^n_{i+1}) - (\rho\eta)(\mathbf{U}^\star_{r,i+\frac{1}{2}})\right]\right\}.\end{aligned}$$

It remains to exhibit a lower bound to the right-hand side of the foregoing inequality. This is achieved invoking the consistency of the underlying Eulerian approximate Riemann solver with the integral form of the Eulerian entropy inequality. Namely, (2.2.91) holds true

$$-\Lambda_l(\rho^\star_l\eta^\star_l - \rho_l\eta_l) - \Lambda_0(\rho^\star_r\eta^\star_r - \rho^\star_l\eta^\star_l) - \Lambda_r(\rho_r\eta_r - \rho^\star_r\eta^\star_r) + \rho_r\eta_r u_r - \rho_l\eta_l u_l \geq 0.$$

Substituting the foregoing inequality at the right-hand side of (2.2.106) for the terms located at $x_{i-\frac{1}{2}}$ (resp. $x_{i+\frac{1}{2}}$) and developing the expressions of the negative and the positive parts of the wave speeds, we finally obtain the cell entropy inequality

$$(\rho\eta)(\mathbf{U}_i^{n+1}) - (\rho\eta)(\mathbf{U}_i^n) + \frac{\Delta t}{\Delta x_i}\left[(\rho\eta u)^\star_{i+\frac{1}{2}} - (\rho\eta u)^\star_{i-\frac{1}{2}}\right] \geq 0, \tag{2.2.106}$$

where the entropy flux, $\rho\eta u$, at $x_{i-\frac{1}{2}}$ writes

$$\begin{aligned}(\rho\eta u)^{\star}_{i+\frac{1}{2}} &= \frac{1}{2}\left[(\rho\eta u)(\mathbf{U}^n_i) + (\rho\eta u)(\mathbf{U}^n_{i+1})\right] \\ &- \frac{\mid \Lambda_{l,i+\frac{1}{2}} \mid}{2}\left[(\rho\eta)(\mathbf{U}^{\star}_{l,i+\frac{1}{2}}) - (\rho\eta)(\mathbf{U}^n_i)\right] \\ &- \frac{\mid \Lambda_{0,i+\frac{1}{2}} \mid}{2}\left[(\rho\eta)(\mathbf{U}^{\star}_{r,i+\frac{1}{2}}) - (\rho\eta)(\mathbf{U}^{\star}_{l,i+\frac{1}{2}})\right] \\ &- \frac{\mid \Lambda_{r,i+\frac{1}{2}} \mid}{2}\left[(\rho\eta)(\mathbf{U}^n_{i+1}) - (\rho\eta)(\mathbf{U}^{\star}_{r,i+\frac{1}{2}})\right].\end{aligned}$$

This concludes the design of the Eulerian scheme. The next section focuses on its numerical validation.

2.2.9.f Summary and algorithm

The algorithm starts from the data of $\mathbf{U}^n_i$ at initial time index $n = 0$ for all cells index i and from the data for the boundary conditions. We then execute successively the steps until the final time $t_{\text{final}} = t^{n+1}$ is reached:

– **In.** Data $\mathbf{U}^n_i$ for all cells.

1. **Riemann problem:** Compute $\mathbf{F}^*$ for any interface.
   a. *State evaluation.* Evaluate left and right states $\mathbf{U}_l$ and $\mathbf{U}_r$ of each interface.
   b. *Evaluate* λ_l, λ_r. using **Method 1**, **Method 2** or **Method 3** (refer to section 2.2.4) to ensure positivity and stable entropy.
   c. *Evaluate the intermediate wave speed* u^* *with* (2.2.29) *that gives the contact discontinuity* Λ_0.
   c. *Evaluate the left and right wave speeds* Λ_l *and* Λ_r.
   d. *Evaluate the intermediate states.*
2. **Compute numerical fluxes** $\mathbf{F}^*$ with (2.2.102).
3. **Evaluate** Δt using (2.2.105).
4. **Update** $\mathbf{U}^{n+1}_i$ using (2.2.101).

– **Out.** Data $\mathbf{U}^{n+1}_i$ for all cells.

2.2.10 One-dimensional Eulerian numerical validation

We first run the test suite from Toro (1999)'s book, In his book, Toro has gathered some now-classical test cases. All tests are simulated on domain $\Omega = [0, 1]$ with $\gamma = 7/5$, a CFL set to 0.9 and the number of cells is 100. The initial configurations are recalled in

table 2.1 where the discontinuity is located at x_0 and the final time is set to t_{final}. We refer the readers to Toro (1999) for the phenomenological description of the test cases. The results are gathered in figure 2.12 for tests 2 to 5 for the current Eulerian scheme. We present the density, velocity and internal energy variables against the exact solution. For such first order results, the numerical scheme behaves as expected. For such first order results, the numerical scheme behaves as expected. The obtained numerical results are perfectly consistent with what can be obtained by a first-order accurate Eulerian Finite Volume scheme. Next in figure 2.13 we present the numerical results obtained by the current scheme and the classical HLL one on tests 6 and 7, which correspond respectively to an isolated stationary contact wave and a moving contact wave. The current numerical scheme is able to exactly resolve a stationary contact by construction (left panel), and can capture the moving contact discontinuity but with some numerical dissipation. Notice that the HLL scheme has no embedded mechanism to capture such stationary contact, while the family of HLLC schemes has, likewise for our approach. Recall that the

Label	Left state $(\rho, u, p)_l$	Right state $(\rho, u, p)_r$	Discont. x_0	Final time t_{final}
2	$(1, -2.0, 0.4)$	$(1, 2, 0.4)$	0.5	0.15
3	$(1, 0, 1000)$	$(1, 0, 0.01)$	0.5	0.012
4	$(5.99924, 19.5975, 460.894)$	$(5.99242, -6.19633, 46.0950)$	0.4	0.035
5	$(1, -19.59745, 1000)$	$(1, -19.59745, 0.01)$	0.8	0.012
6	$(1.4, 0, 1)$	$(1, 0, 1)$	0.5	2.0
7	$(1.4, 0.1, 1)$	$(1, 0.1, 1)$	0.5	2.0

Table 2.1: Initial conditions for the 1D test cases from Toro (1999).

Lagrangian scheme naturally adapts its mesh to the flow: some smaller and compressed cells are encountered after a shock wave and expanded ones after a rarefaction wave. Moreover, moving contact discontinuities are exactly preserved. This however comes with the drawback of entropy deposition even in rarefaction waves. On the contrary, an Eulerian scheme considers a fixed mesh, hence ensuring a better accuracy in rarefaction waves, but less accurate shocks and diffused moving contact discontinuities. However, the extension to multi-dimensions and high-orders of accuracy is somewhat simpler with Eulerian schemes since in this formalism the mesh is not moving. This set of test cases have validated the first order Eulerian Finite Volume scheme and its theoretical behaviors are numerically reproduced or observed.

We now test the first-order Finite Volume scheme for Eulerian gas dynamics on other complex flows. We fix the CFL to 0.9 and $\gamma = 7/5$ unless otherwise noticed.

Modified Sod shock tube The modified Sod test simulates the impact of a dense fluid onto a light one at rest. The simulation is ran on the interval of $\Omega = [0, 1]$ with the interface located at $x = 0.3$. The initial set up reads

$$(\rho, u, p)_l = (1, 0, 75, 1) \quad , \quad (\rho, u, p)_r = (0.125, 0, 0.1) \quad , \quad t_{final} = 0.2.$$

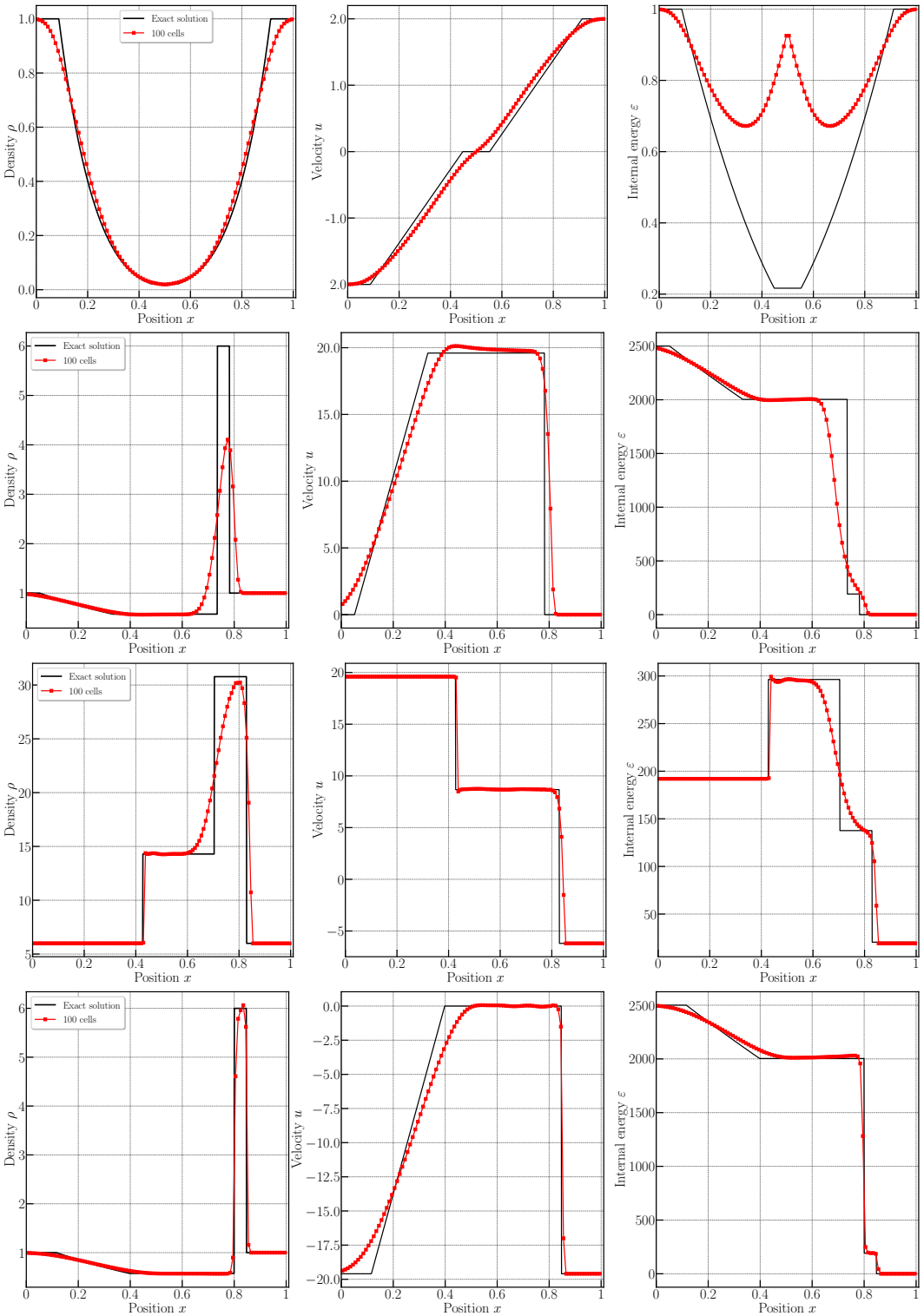


Figure 2.12: Test 2 to 5 of Toro (top to bottom) — Eulerian numerical scheme $N = 100$ cells, $CFL = 0.9$ — Density, velocity, and specific internal energy (left to right) for the exact solution (black line) vs the numerical results (symbols).

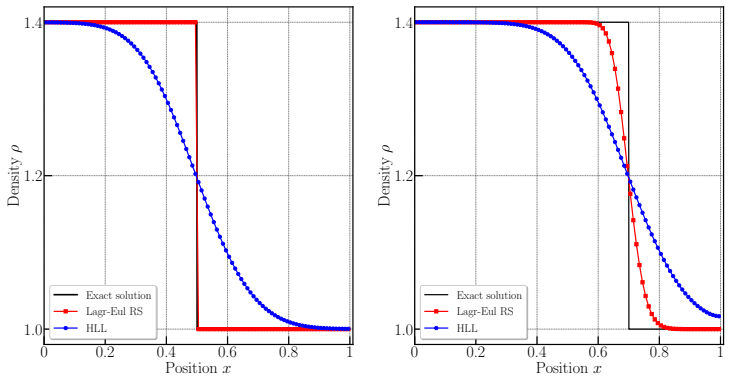


Figure 2.13: Test 6 and 7 of Toro (left to right) — Eulerian numerical schemes $N = 100$ cells, $CFL = 0.9$ —Density for the current scheme (red crosses) against the classical HLL scheme (blue bullets) and the exact solution (black line).

A piston boundary condition is imposed on the left with $u^\star = u_l = 0.75$, while $u^\star = 0$ is imposed on the right. This test case ensures that the sonic glitch in the rarefaction wave is absent. The numerical results at final time $t_{\text{final}} = 0.2$ with $N = 100$ to 400 cells are presented in figure 2.14. The numerical solution seems to converge towards the exact one. More importantly, we do not observe any sonic glitch in the rarefaction wave, as illustrated in figure 2.15.

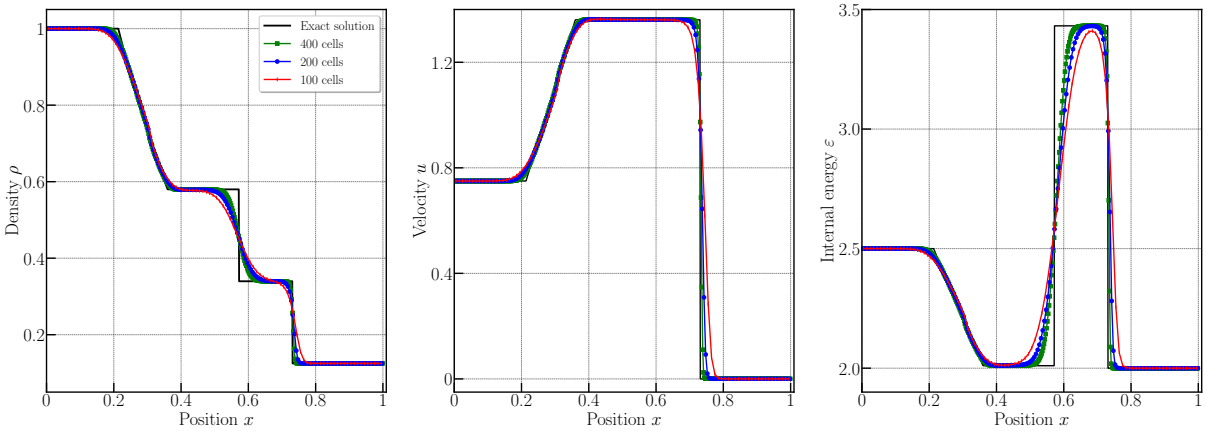


Figure 2.14: Modified Sod problem — Eulerian numerical scheme —Density, velocity, and specific internal energy for the exact solution (black line) vs the numerical results (symbols) with grid convergence such as $N = 100$ cells (red cross), 200 (blue circle) and 400 (green square).

Collela-Woodward blastwave In figure 2.16, we present the numerical density obtained for $N = 800$ to 6400 cells, with a zoom of the central area. The quality of the solution improves with the mesh refinement. Then we present on the middle panel the time-step evolution which clearly adapts to the numerical solution. On the right panel, the plot of the normalized right impedance as a function of time allows us again to track the main shock waves and their interactions.

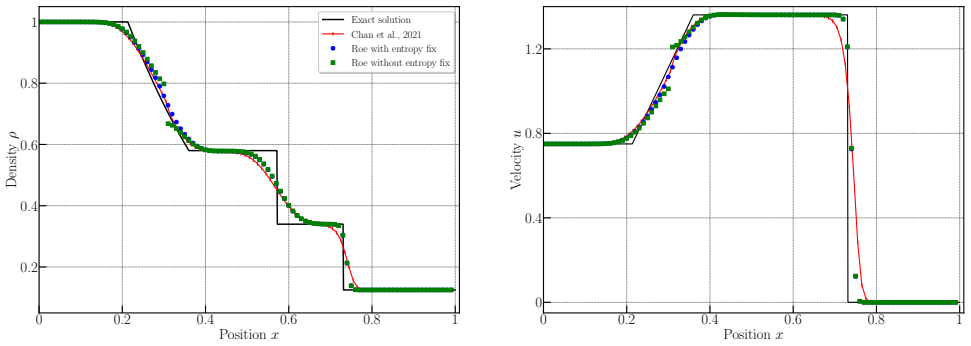


Figure 2.15: Modified Sod problem — Comparison of numerical scheme and Roe scheme with and without entropy fix — Density and velocity for the Eulerian numerical solution (Chan *et al.* (2021) in red cross) vs the numerical results of Roe without entropy fix (green squares) and Roe with entropy fix (blue circle) on $N = 100$ and $CFL = 0.9$.

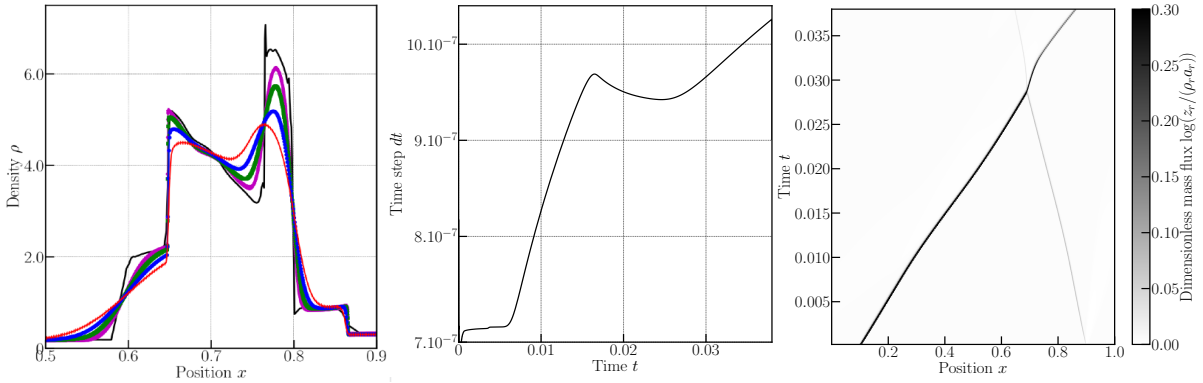


Figure 2.16: Woodward-Collela blastwave problem — Eulerian numerical scheme — Density for the reference solution (black line) vs the numerical results — Left: Grid convergence of the density variable on $N = 800$ cells (red cross), 1600 (blue circle), 3200 (green square) and $N = 6400$ cells (magenta diamond) — Middle: Time step evolution — Right: Evolution of the normalized right impedance λ_r with each time step with $N = 1600$.

Le Blanc shock tube We simulate the Le Blanc shock tube with the Eulerian scheme. Figure 2.17 (left panel) shows the numerical density plotted in the log scale for the Le Blanc shock tube problem for $N = 900$, 1600 and 3200 cells. The panel in the middle presents the time-step evolution showing that the main adaptation is related to the separation of the simple waves Then the evolution of the normalized right impedance that traces the path of the right moving shock is presented on the right panel. Again because only one single shock is present in this problem, then only one curves can be observed.

Extreme double rarefaction We present the results for the extreme double rarefaction problem in figure 2.18. No positivity issue is encountered, and the minimal density is about $\rho_i = 1.48 \times 10^{-3}$ for $N = 800$ cells ($\rho_i = 2.43 \times 10^{-3}$ for 400 cells and $\rho_i = 9.09 \times 10^{-3}$ for 100 cells). The grid convergence results for the density, velocity and specific internal energy show that the numerical scheme improves the quality of the numerical solution

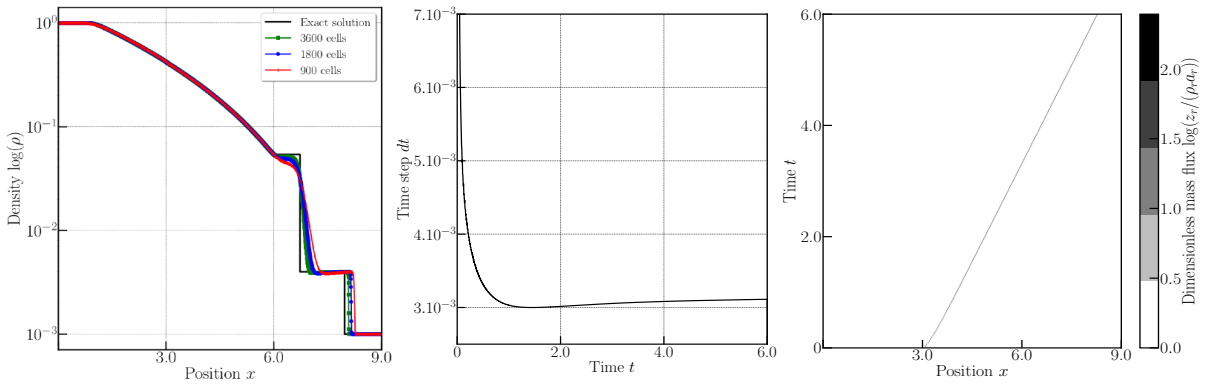


Figure 2.17: Le Blanc shock tube problem — Eulerian numerical scheme — Left : Density plotted on the *log* scale with grid convergence on $N = 900$ (red cross), $N = 1800$ (blue circle) and $N = 3600$ (green squares) cells — Middle: Time step evolution — Right: Evolution of the normalized right impedance λ_r with each time step with $N = 1800$.

when the mesh size decreases.

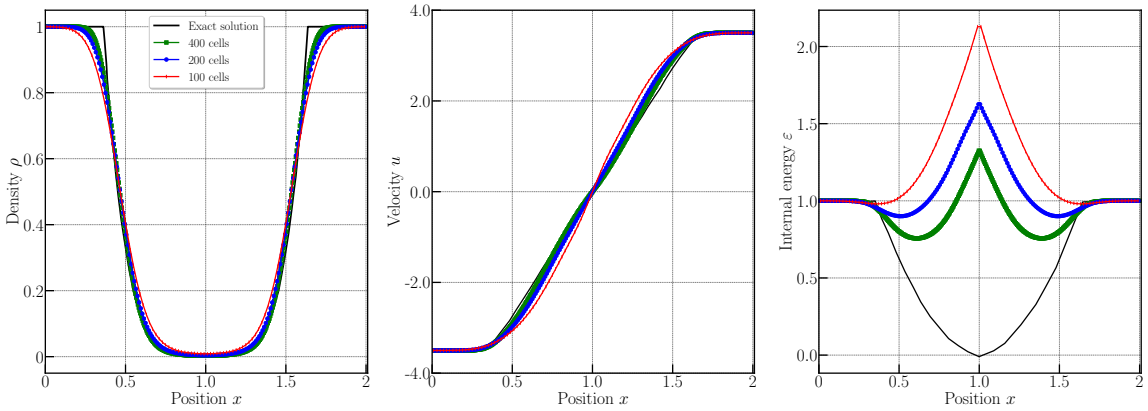


Figure 2.18: Extreme 123 problem — Eulerian numerical scheme — Density, velocity, and specific internal energy for the exact solution (black line) vs the numerical results (symbols) — Top panels: numerical results for $N = 100$ cells — Bottom panels: Grid convergence, $N = 100$ cells (red cross), 200 (blue circle) and 400 (green square).

The next step of this work consists in using this scheme as a so-called 'parachute' scheme in an *a posteriori* MOOD limited high-order scheme.

2.3 Extension to high-order of accuracy

In this section we present an extension of the previous first-order Eulerian scheme into a high-order accurate space/time scheme. Time discretization proceeds via classical Strong Stability preserving Runge-Kutta (SSPRK) scheme, refer to Gottlieb *et al.* (2001); Gottlieb & Shu (1998). The space discretization relies on classical polynomial reconstruction with not-so-classical *a posteriori* MOOD loop limiting procedure, see Clain *et al.* (2011). The MOOD paradigm for FV schemes is built upon a try-and-fail concept. Starting from data, $(\mathbf{U}_i^n)_{1\leq i\leq N}$ at t^n a high-order scheme produces a 'candidate' solution $(\mathbf{U}_i^{*,n+1})_{1\leq i\leq N}$ at t^{n+1}. This candidate solution is tested against specific 'detection' criteria that determines whether the computed cell value at t^{n+1} is physically and numerically valid. In the case of validity the cell is accepted, otherwise the cell is flagged as troubled/bad and sent back to t^n for re-computation. The re-computation is performed using a lower order scheme from a 'cascade' of schemes ordered from the more accurate to the more robust one. The rudiment scheme of this cascade is called the 'parachute' and must always produce a valid solution according to the detection criteria. This parachute scheme must be extremely robust and provably adapted to the detection criteria. In the previous sections we have designed such a first-order Eulerian scheme with good properties: conservation, robustness, positivity preservation under explicit CFL, entropy inequality, etc. The entire iterative procedure is called a MOOD loop and always converges to an acceptable numerical solution for which some cells have been updated with a high order scheme, while others with the parachute. Figure 2.19 shows an illustration to build a high-order numerical scheme with an *a posteriori* MOOD loop.

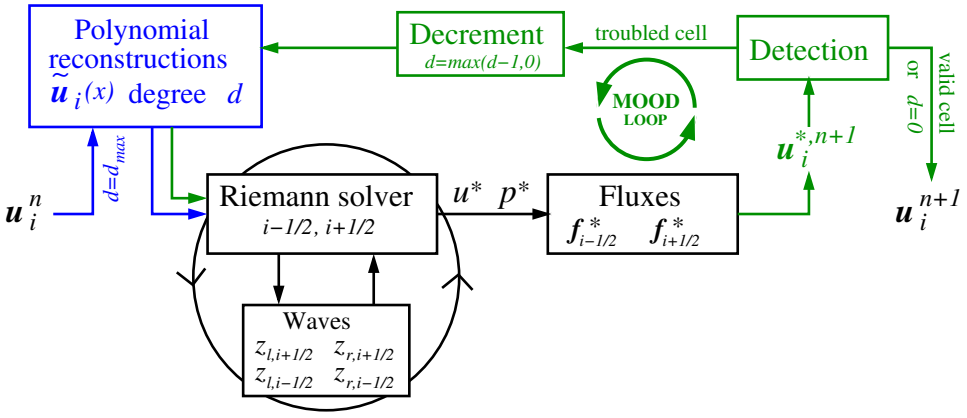


Figure 2.19: Sketch of the high order numerical scheme based on polynomial reconstructions (blue) and *a posteriori* MOOD loop (green) embracing the general scheme (black).

Several works using an *a posteriori* MOOD loop in different numerical contexts can be found for Lagrangian, ALE, Discontinuous Galerkin limiter, Smoother-Particle-Hydrodynamics, Adaptative Mesh Refinement, WENO schemes, slope limiter, such as the works of Nogueira *et al.* (2016); Vilar (2019); Boscheri *et al.* (2018, 2015); Zanotti *et al.* (2015); Desveaux & C.Berthon (2013); Loubère *et al.* (2014); Farmakis *et al.* (2020); Kitamura & Hashimoto (2017), and, physical contexts like Euler or Navier-Stokes equa-

	Cell indexes							Stencil size	Linear system size
Degree d	$i-3$	$i-2$	$i-1$	i	$i+1$	$i+2$	$i+3$	$2K_i = \lvert\mathcal{S}_i^d\rvert$	$(2K_i)\times d$
0				–				0	—
1			✓	–	✓			2	2×1
2		✓	✓	–	✓	✓		4	4×2
3		✓	✓	–	✓	✓		4	4×3
4	✓	✓	✓	–	✓	✓	✓	6	6×4
5	✓	✓	✓	–	✓	✓	✓	6	6×5

Table 2.2: Centered stencils $\mathcal{S}_i^d$ used for the polynomial reconstructions of degree d, their size and the linear system size associated to the reconstruction operator.

tions, turbulence simulation, shallow-water equations, astrophysics, MHD, Baer-Nunziato multi-phase model, such as the works of Jiang *et al.* (2018); Clain & Figueiredo (2017); Zanotti *et al.* (2015); Fambri *et al.* (2017); Kemm *et al.* (2020)

2.3.1 High order space discretization

Let us denote by $o = d+1 > 0$ the target order of accuracy fixed by the user. At discrete time t^n the polynomial reconstruction operator $\mathcal{R}$ in cell Ω_i is the operator which given a large enough set of $2K_i$ neighbor mean values on stencil $\mathcal{S}_i^d = \{\mathbf{U}_{i\pm k},\ k = 1,\dots,K_i\}$ produces the polynomial $\widetilde{\mathbf{U}}_i \in \mathbb{P}_d(\Omega)$. We skip the time exponent in this section as no confusion can occur. This polynomial has a degree $d \geq 0$ and is such that its mean value exactly matches $\mathbf{U}_i$ and best fits the neighbor mean values $\mathbf{U}_{i\pm k}$, that is

$$\mathcal{R} : \left(x \in \Omega_i, \mathbf{U}_i, d, \mathcal{S}_i^d\right) \longrightarrow \widetilde{\mathbf{U}}_i(x). \tag{2.3.1}$$

$\widetilde{\mathbf{U}}_i$ is expressed under a Taylor basis as $\widetilde{\mathbf{U}}_i(x) = \mathbf{U}_i + \sum_{m=1}^{d} r_{i,m}\psi_{i,m}(x)$ where the unknowns are the polynomial coefficients $r_{i,m}$. The polynomial basis functions $\{\psi_{i,m}\}_{m=1,\dots,d}$ are such that for all m: $\psi_{i,m}(x) = (x-x_i)^m - \frac{1}{\Delta x_i}\int_{\Omega_i}(x-x_i)^m\,dx$, where $x_i = \frac{1}{|\Omega_i|}\int_{\Omega_i} x dx$ is the cell centroid. The polynomial coefficients are then optimized in the least-squares sense by the minimization of the cost function

$$J\left(\{r_{i,m}\}_{m=1,\dots,d}\right) = \sum_{k=1}^{K_i}\left|\mathbf{U}_{i\pm k} - \frac{1}{\Delta x_{i\pm k}}\int_{\Omega_{i\pm k}}\widetilde{\mathbf{U}}_i^d(x)\,dx\right|^2, \tag{2.3.2}$$

which amounts to solve an over-determined linear system if the size of the stencil $\mathcal{S}_i^d$ is such that $K_i > d$, and usually we take $K_i \simeq 1.5d$. The size of the linear system is $(2K_i)\times d$. The reconstruction operator is accurate at order $(d+1)$ for smooth enough solution. For the sake of simplicity the stencils $\mathcal{S}_i^d$ are always the centered ones and adapt their width as d changes, see table 2.2. In this work the primitive variables are reconstructed. Notice that conservative or characteristics variables could also be considered leading to slightly more or less oscillatory results. The reconstructed values are used in substitution to the

mean values within the Riemann solvers as

$$\begin{aligned} \mathbf{U}_{l,i-1/2} &\equiv \widetilde{\mathbf{U}}_{i-1}(x_{i-1/2}), \qquad & \mathbf{U}_{r,i-1/2} &\equiv \widetilde{\mathbf{U}}_{i}(x_{i+1/2}), \\ \mathbf{U}_{l,i+1/2} &\equiv \widetilde{\mathbf{U}}_{i}(x_{i+1/2}), & \mathbf{U}_{r,i+1/2} &\equiv \widetilde{\mathbf{U}}_{i+1}(x_{i+1/2}). \end{aligned} \tag{2.3.3}$$

Also remark that the reconstruction operates at time t^n, and, once a maximal polynomial degree is set at $d \geq 0$, then at maximum $(d+1)$ reconstructions are possible for degrees $0 \leq k \leq d$. Obviously for $d = 0$ the reconstruction is trivially equal to the cell mean value.

2.3.2 *a posteriori* High-order limiting: detection, parachute scheme and MOOD loop

As is well known, the polynomial reconstruction operator suffers from the Runge phenomena in the case of non-smooth enough solution, leading to oscillatory polynomials $\widetilde{\mathbf{U}}_i$ as soon as $d \geq 1$. In classical second-order Finite Volume schemes ($d = 1$) a so-called slope (or flux) limiting procedure is supplemented to avoid those spurious oscillations. Unfortunately, there exists no agreement on how to limit higher order polynomial reconstructions. In this work we rely on the *a posteriori* Multi-dimensional Optimal Order Detection (MOOD) algorithm referring to Clain *et al.* (2011). For this paradigm three entities are needed:

(i) a list of *detection criteria* $\mathcal{D}$ to assess the validity of a numerical solution. These criteria must ensure the physical admissibility of the numerical solution (*i.e.* the positivity). Once the physical admissibility is assured then some numerical acceptability properties may be required. Often an essentially-non-oscillatory (ENO) behavior, and, possibly the computer representation validity (*i.e.* no `NaN`) are enforced. Bad/troubled cells are gathered into set $\mathcal{B}$.

(ii) a *parachute scheme* which must be a genuinely robust scheme producing a numerical solution fulfilling the detection criteria $\mathcal{D}$. Here we consider the first order Eulerian scheme developed in this work. It gathers several useful properties indeed: mathematical firmly based grounds, Lagrangian↔Eulerian mapping, demonstrated ordered wave speeds, positivity under explicit CFL condition, entropy inequality and validation of its numerical behaviors on classical test cases (carbuncle-free, robustness, efficiency);

(iii) a *cascade of schemes* ranked from the more accurate and prone to spurious instability to the more robust one, that is the parachute scheme. This cascade may be parametrized by the polynomial degree d, for instance we can consider the following cascade: $d = 4 \to 3 \to 2 \to 1 \to 0$;

Detection criteria. The physical admissibility criteria $\mathcal{P}$ are firmly based on the system of PDEs solved, they are obviously the positivity of the density and internal energy for

the Eulerian scheme, hence

$$\mathcal{P} \quad : \quad \rho_i^* > 0, \quad \text{and} \quad \varepsilon_i^* > 0 \tag{2.3.4}$$

Next, the numerical criteria ensuring the ENO behavior are more subjective and somewhat attached to the developer preferences. Here we rely on a relaxed discrete maximum principle (RDMP) but applied to the output data of the nodal solver. Indeed the nodal solver is (one of) the constitutive brick, the locality of which is restricted to the neighbor cells. Moreover, the outcome of the nodal solver are the nodal velocity and sub-pressures, which further serve to determine the impedances, and, ultimately the star states and the numerical fluxes to update the state vector. A candidate variable q_p^* fulfills the RDMP if

$$\mathcal{N} \quad : \quad -\delta q_p + \min_{k\in\mathcal{C}(p)} (q_k^n) \leq q_p^* \leq \max_{k\in\mathcal{C}(p)} (q_k^n) + \delta q_p, \tag{2.3.5}$$

where $\mathcal{C}(p)$ is the neighborhood associated to point p, $\delta q_p = \max\left(10^{-4}, 10^{-3}(M_p - m_p)\right)$, and, M_p and m_p are the maximal and minimal local values of q If a nodal solver result does not fulfill (2.3.5) then the surrounded cells are marked as 'bad' ones.
At last we also test if the numerical method has generated any un-representable value such as `NaN`. If a cell has been declared as 'invalid', then only this cell and its neighbors are sent back to t^n for a re-computation with local lower order reconstructions. Possibly some cells are re-computed several times, and, may ultimately be updated with the parachute scheme. The troubled/bad cells are the cells which do not fulfill at least one of these detection criteria

$$\mathcal{D} : \begin{cases} \mathcal{P} & : \text{ physically admissible, } i.e \text{ positivity under explicit } \Delta t \text{ with respect to (2.3.4)} \\ \mathcal{C} & : \text{ representable in a computer, } i.e \text{ no } \texttt{Nan}. \\ \mathcal{N} & : \text{ numerically valid, } i.e \text{ essentially non oscillating with respect to (2.3.5).} \end{cases}$$

The two neighbors of a bad cell are also recomputed, and, as such flagged as bad ones without polynomial degree decrementing.
This type of limiting is referred to an *a posteriori* procedure because the candidate solution at t^{n+1} is always confronted to the detection criteria and must pass them. Fortunately in general only few cells need to be re-computed which renders this *a posteriori* MOOD loop relatively efficient in comparison to other limiting approaches.

MOOD loop. The MOOD loop always converges in a finite number of iterations. Indeed there is only a few and fixed number of schemes in the cascade and a finite number of cells. Moreover, a cell is always considered as valid if its reconstruction degree has dropped to $d_i = 0$. Consequently the number of MOOD loop iterations is necessarily finite. The final valid solution is then constituted of pieces of high accurate and low accurate approximations fulfilling the detection criteria. As a consequence the detection criteria do play a paramount role in the overall quality of the solution.

2.3.3 High-order time discretization

For the time discretization we rely on successive explicit sub-steps, which altogether allows to reach a nominal dth accuracy. Depending on the order of accuracy d, we employ the family of classical Runge-Kutta (RK) explicit schemes by Gottlieb & Shu (1998). For $d = 0$ a forward Euler scheme is used, while for $d = 1$ an explicit midpoint method (predictor-corrector or equivalently RK2 scheme) is employed. When $d = 2$ the classical SSPRK3 scheme is chosen. Beyond $d \geq 3$ there exists no SSPRK method anymore, and we then rely either on RK3 with appropriate time-step limitation or on a RK4 scheme, see table 2.4 for the associated Butcher tableaus. We denote by $\mathrm{K} > 1$ the number of RK iterates of the time discretization scheme, each indexed by κ and $1 \leq \kappa \leq \mathrm{K}$.
The restriction on the time step Δt is the one derived in the first sections which ensures the positivity with a safety coefficient $CFL \leq 1$. The time step $\Delta t \equiv \Delta t^{\kappa=1}$ is determined at the end of the first RK iterate as a function of the minimal/maximal impedances $\lambda^{\kappa=1}_{l,i+1/2}$ and $\lambda^{\kappa=1}_{r,i+1/2}$. It may happen that, during the physical evolution the time step required at RK iterate κ is such that: $\Delta t^{\kappa} < \Delta t$. In this case the cell is flagged as a 'bad' one and will be limited (in space and time) by the MOOD loop, possibly dropping its time accuracy to a first order scheme for which $\Delta t \equiv \Delta t^{\kappa=1}$ always ensures the positivity.

c_1	a_{11}	a_{12}	$\cdots$	a_{1s}
c_2	a_{21}	a_{22}	$\cdots$	a_{2s}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	
c_s	a_{s1}	a_{s2}	$\cdots$	a_{ss}
	b_1	b_2	$\cdots$	b_s

Generic RK

Table 2.3: Butcher tableau for the Generic Runge-Kutta explicit time discretization schemes

0	0	0
1	1	0
	1/2	1/2

RK2

0	0	0	0
1	1	0	0
1/2	1/4	1/4	0
	1/6	1/6	2/3

SSPRK3

0	0	0	0	0
1/2	1/2	0	0	0
1/2	0	1/2	0	0
1	0	0	1	0
	1/6	1/3	1/3	1/6

RK4

Table 2.4: Three examples of Butcher tableaus for Runge-Kutta time discretization schemes used in this work.

2.3.4 Numerical validation of high order extension

In the previous numerical sections we have tested the first order Lagrangian and Eulerian schemes. Here we solely focus on test cases with more complex interacting wave problems, such as the Collela-Woodward blastwave and the Shu-Osher test. Above all, we perform a

convergence analysis of our numerical scheme using a smooth solution of the gas dynamics equations.

The following schemes are under scrutiny

- `Eul-0`: 1st order Eulerian scheme;
- `Eul-X`: (X+1)th order (unlimited) Eulerian scheme;
- `Eul-XMOOD`: (X+1)th order Eulerian scheme with MOOD limiting, with `Eul-0` as parachute scheme.

which will be compared to some classical ones such as: `Lag-0`, the 1st order Lagrangian scheme from section 2.2.5, and, `Eul-1Lim`, the nominally 2nd order Eulerian scheme i.e with piece-wise linear reconstruction on primitive variables supplemented with classical van Leer slope limiter.

Density sine wave This paragraph shows that the high order reconstructions and RK schemes allow respectively for 1^{st}, 2^{nd}, 3^{rd}, 4^{th} and 5^{th} order convergence in space for a smooth solution of the 1D compressible Euler equations. We consider the advection of a sine wave on a computational domain of $\Omega = [0, 1]$ with periodic boundaries. The sine wave is characterized by:

$$(\rho, u, p)_0 = (1 + 0.1\sin(2\pi\Omega), 1, 1/\gamma),$$

The results of the convergence analysis performed with polynomial reconstruction going from $d = 0$ to $d = 4$ are displayed in table 2.5, and the L^2 error is shown in figure 2.20. The expected rate of convergence is reached and confirms the high-order accuracy for our numerical scheme.

Next we make use of this test case to compare the efficiency of the high-order schemes, *i.e.* the accuracy as a function of the computation time. In figure 2.21, we plot the relative L_2 error versus the CPU time for the Eul-0, Eul-1, Eul-2, Eul-3 and Eul-4 schemes. Some efficiency plots have been extrapolated to ease the visual comparison. In table 2.6, the efficiencies of the different numerical schemes are compared for a fixed error ($\varepsilon^0 = 10^{-8}$), and, then for a fixed execution time (10ms). For large enough CPU time or small error the high-order schemes out-perform low-order ones.

Modified smooth solution problem The second test case is a smooth test case derived in the isentropic case, with a polytropic index of $\gamma = 3$ for the perfect gas EOS. In this situation, the characteristic curves of the Euler equations become straight lines. The governing equations reduce to two Burgers equations, allowing us to solve this problem analytically. For the numerical simulation, the computational domain is once again $\Omega \in [0, 1]$ with the initial condition modified to yield a more challenging example, as

$$(\rho, u, p)_0 = (1 + 0.1\sin(2\pi\Omega), 0, \rho_0^\gamma),$$

Density Sine wave problem							
	N	L^1 error	L^2 error	L^∞ error	L^1 order	L^2 order	L^∞ order
Eul-0	50	2.0762E-02	2.3002E-02	2.9628E-02	—	—	—
	100	1.1403E-02	1.2634E-02	1.6282E-02	0.86	0.87	0.86
	200	5.9829E-03	6.6288E-03	8.5436E-03	0.93	0.93	0.93
	400	3.0653E-03	3.3962E-03	4.3773E-03	0.96	0.96	0.96
Eul-1	50	9.4635E-04	1.0878E-03	2.0798E-03	—	—	—
	100	2.5988E-04	3.3788E-04	9.7322E-04	1.69	1.86	1.11
	200	6.7953E-05	1.0356E-04	4.0150E-04	1.71	1.96	1.28
	400	1.6946E-05	3.1778E-05	1.6358E-04	1.70	1.99	1.30
Eul-2	50	9.7772E-05	1.3478E-04	2.5498E-04	—	—	—
	100	8.2715E-06	1.0654E-05	1.7347E-05	3.66	3.56	3.87
	200	1.0335E-06	1.1828E-06	1.6504E-06	3.17	3.00	3.39
	400	1.2919E-07	1.4423E-07	1.8995E-07	3.03	2.99	3.11
Eul-3	50	8.5786E-07	9.4984E-07	1.2252E-06	—	—	—
	100	5.2418E-08	5.8083E-08	7.4862E-08	4.03	4.03	4.03
	200	3.2586E-09	3.6102E-09	4.6527E-09	4.00	4.01	4.01
	400	2.0347E-10	2.2543E-10	2.9054E-10	4.00	4.00	4.00
Eul-4	50	2.1204E-07	2.3511E-07	3.0259E-07	—	—	—
	100	7.0001E-09	7.7569E-09	9.9979E-09	4.25	4.26	4.25
	200	2.6359E-10	2.9206E-10	3.7642E-10	4.73	4.73	4.73
	400	1.3827E-11	1.5320E-11	1.9752E-11	4.92	4.92	4.91

Table 2.5: L^1, L^2 and L^∞ norm errors on density ρ between the numerical solution and the exact solution of a density since wave problem until $t_{\text{final}} = 1$.

with periodic boundary conditions. The initial density and pressure being very close to zero would confirm the positivity-preserving property of the numerical scheme. Figure 2.22 display the results of the third-order scheme with 100 cells with a final time of $t_{\text{final}} = 0.8$ and $CFL = 0.1$. Making use of the analytical solution, the global truncation error corresponding to the first- to fourth-order scheme are shown in table 2.7 and in figure 2.23. These results confirm the high accuracy of this scheme.

Shu-Osher oscillatory test case This test Jiang & Shu (1996) is a 1D hydrodynamic shock tube. The downstream flow has a sinusoidal density fluctuation $\rho = 1 - \varepsilon \sin(\lambda \pi x)$ with a wave length $\lambda = 5$ and an amplitude $\varepsilon = 0.2$. A Mach 3 shock front is initially located at $x_0 = -4$ on domain $\Omega \in [-5; 5]$ and a CFL of 0.1. The left and the right states are as follows:

$$(\rho, u, p)_l = (3.857143, 2.629369, 10.33333) \ , \ (\rho, u, p)_r = (1 + 0.2\sin(5\Omega), 0 = 1). \quad (2.3.6)$$

This problem involves small scales after the shock has interacted with the sine wave. These small scale features can be captured either with a fine enough mesh or with high order accurate method. Figure 2.24 shows the comparison of numerical solution using

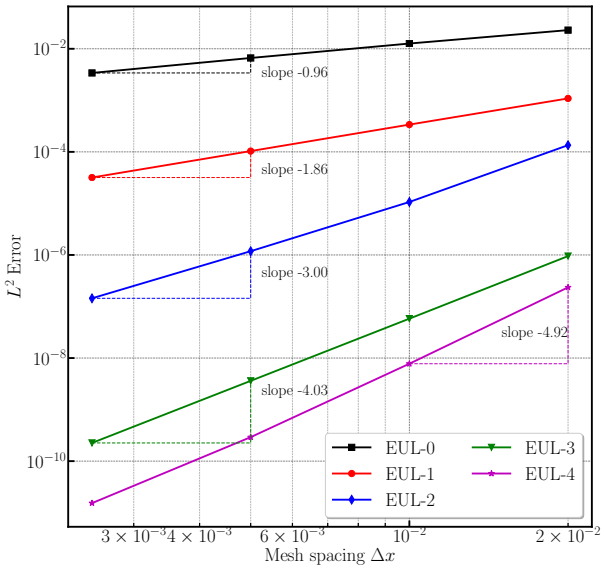


Figure 2.20: Density sine wave test case — Eulerian numerical scheme — L^2 error for the 1^{st}, 2^{nd}, 3^{rd}, 4^{th} and 5^{th} order schemes — Numerical results for $N = 50$, $N = 100$, $N = 200$ and $N = 400$ cells.

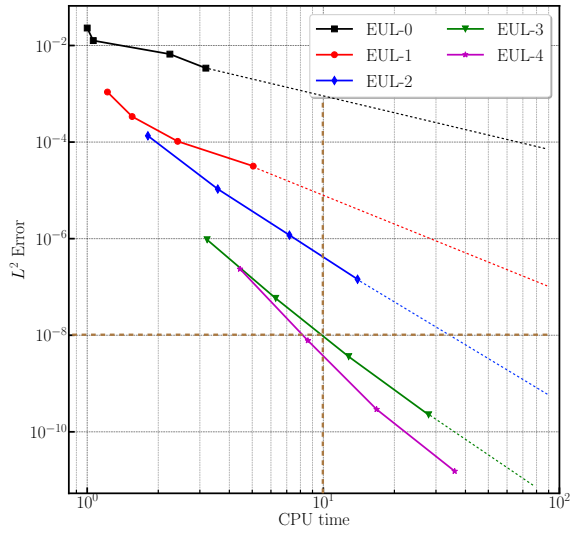


Figure 2.21: Density sine wave test case — Efficiency of the Eulerian numerical schemes — L^2 error versus CPU time for 1^{st}, 2^{nd}, 3^{rd}, 4^{th} and 5^{th} order schemes.

different polynomial reconstruction from `Eul-0` to `Eul-4MOOD` and a reference solution with 200, 400 and 800 cells. As expected the accuracy of the numerical solution improves with the order of polynomial reconstruction. The reference solution taken here is the numerical solution of the `Eul-0` scheme with $N = 10000$ cells (the CPU time is about $6100\mu s$). Figure 2.25 exhibits the associated cell reconstruction type with each time step and as expected, the shock wave is captured in its motion. Then in table 2.8 we present the percentages of cells updated with each polynomial degree from 0 to 4 and for three different meshes. We observe that about 90% of cells are updated with the highest order

	Eul-0	Eul-1	Eul-2	Eul-3	Eul-4
Fixed $\epsilon^0 = 10^{-8}$	$\gg 1000ms$	$\approx 100ms$	$12.5ms$	$10ms$	$8ms$
Fixed CPU $= 10ms$	$\approx 10^{-3}$	$\approx 10^{-5}$	$\approx 5 \times 10^{-7}$	$\approx 1 \times 10^{-8}$	$\approx 5 \times 10^{-9}$

Table 2.6: Density sine wave test case — Efficiency of the high-order Eulerian numerical schemes — Comparison of efficiency for a fixed error (horizontal line) and at a fixed CPU time (vertical line).

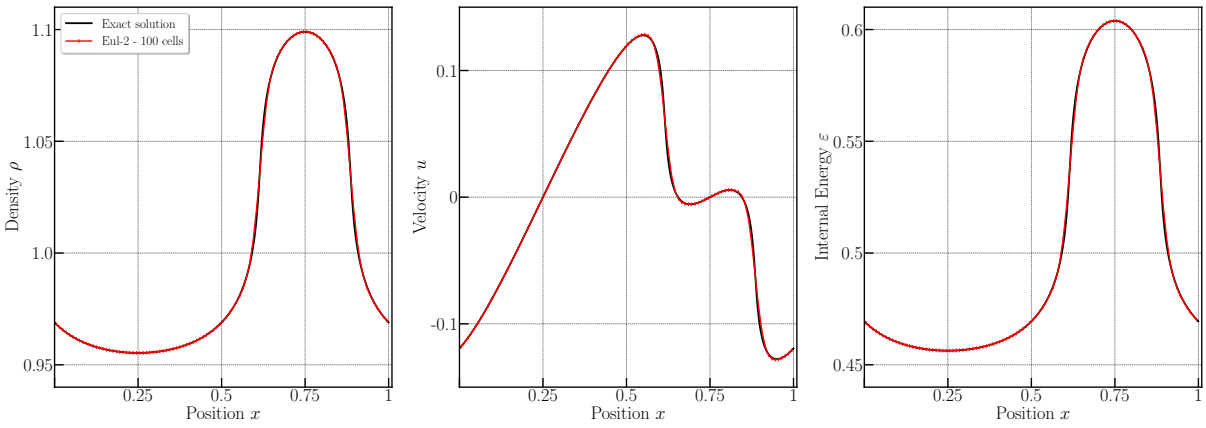


Figure 2.22: Smooth solution test case — Third order ($d = 3$) Eulerian numerical scheme — Density, velocity, and specific internal energy for the exact solution (black line) vs the numerical results (symbols) — Numerical results for $N = 100$ cells (red cross) — The curves are almost superimposed.

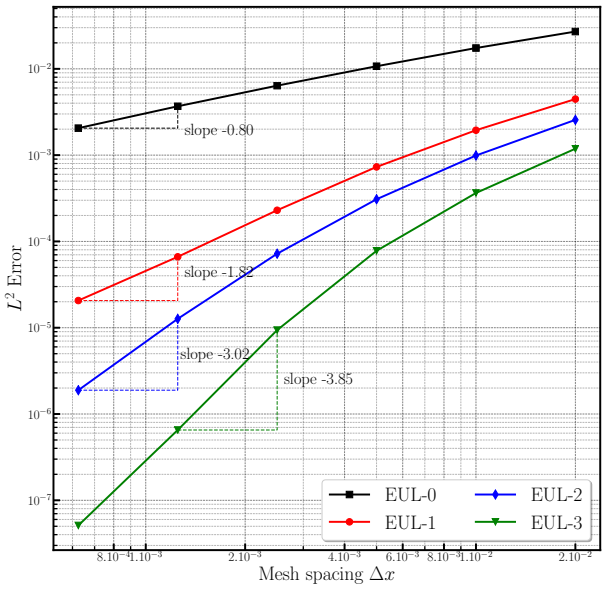


Figure 2.23: Modified smooth solution test case — Eulerian numerical scheme — L^2 error for the 1st, 2nd, 3rd and 4th order accurate schemes — Numerical results for $N = 50$, 100, 200, 400 and 800 cells.

scheme, the 10% left are updated with generally the parachute scheme. In the very same table we gather the CPU time needed to get the final solution as a function of the nominal accuracy of the scheme employed. $d = 0$ means that the first order scheme is run, while $d = 4$ means the $5th$ order MOOD scheme is used. As is observed the cost of second

Modified smooth solution problem							
	N	L^1 error	L^2 error	L^∞ error	L^1 order	L^2 order	L^∞ order
Eul-0	50	2.3172E-02	2.7155E-02	2.2356E-02	—	—	—
	100	1.4534E-02	1.7465E-02	1.3051E-02	0.67	0.64	1.02
	200	8.5132E-03	1.0735E-02	6.0451E-03	0.77	0.70	0.94
	400	4.7798E-03	6.3890E-03	2.6579E-03	0.83	0.75	0.95
	800	2.6049E-03	3.6838E-03	1.2324E-03	0.87	0.80	0.98
Eul-1	50	2.9027E-03	4.7513E-03	5.8641E-04	—	—	—
	100	1.0127E-03	2.0078E-03	1.5676E-04	1.52	1.25	1.68
	200	3.2378E-04	7.4750E-04	4.3627E-05	1.65	1.43	1.72
	400	9.1613E-05	2.3315E-04	1.1739E-05	1.82	1.68	1.73
	800	2.3826E-05	6.6826E-05	3.1284E-06	1.94	1.82	1.75
Eul-2	50	1.3677E-03	2.5634E-03	7.8677E-03	—	—	—
	100	3.7045E-04	9.9020E-04	3.8999E-03	1.89	1.92	1.71
	200	9.3415E-05	3.0817E-04	1.4564E-03	2.00	2.21	2.06
	400	1.6942E-05	7.2047E-05	4.1907E-04	2.47	2.62	2.16
	800	2.5154E-06	1.2679E-05	8.4963E-05	2.75	3.02	2.68
Eul-3	50	5.8769E-04	1.1870E-03	4.0506E-03	—	—	—
	100	1.1633E-04	3.6426E-04	1.8608E-03	2.34	1.70	1.12
	200	2.2343E-05	7.7895E-05	4.4407E-04	2.38	2.23	2.07
	400	2.3799E-06	9.3888E-06	8.3503E-05	3.23	3.05	2.41
	800	2.2748E-07	6.5266E-07	6.6083E-06	3.39	3.85	3.66

Table 2.7: L^1-, L^2- and $L^\infty-$norm errors on density ρ between the numerical solution and the exact solution of the modified smooth solution problem until $t_{\text{final}} = 0.8$.

order scheme is about 3 times more expensive than a first order one (respectively 6, 7 and 9 times for third, fourth and fifth order ones).

	Percentage of cell reconstruction type(%)					**Comparison of CPU time (μs)**				
	$d=0$	$d=1$	$d=2$	$d=3$	$d=4$	$d=0$	$d=1$	$d=2$	$d=3$	$d=4$
N = 200	3.0	0.25	3.0	0.25	92.50	129	288	423	869	1254
N = 400	10.25	0	0.5	0.25	89.75	248	980	1716	1791	2292
N = 800	10.5	0	0.15	0.1	89.75	500	2130	3454	4079	4388

Table 2.8: Shu-Osher oscillatory test case — High-order extension of the Eulerian numerical scheme with 200, 400 and 800 cells respectively — Percentage of cells with 1^{st}, 2^{nd}, 3^{rd} and 4^{th} order cell reconstruction (Left) — Comparison of the CPU time with 1^{st}, 2^{nd}, 3^{rd} and 4^{th} order cell reconstruction for 200, 400 and 800 cells (Right).

Woodward-Colella blastwave The complex flow pattern in the Woodward-Collela blastwave problem is a good test case to evaluate the high-order extension of our numerical scheme. Similar to the Shu-Osher problem, we present the numerical density with a zoom at the central area obtained by the `Eul-0`, `Eul-1MOOD`, `Eul-2MOOD` and `Eul-3MOOD` schemes compared with a reference solution with 200, and 400 cells in figure 2.26. The

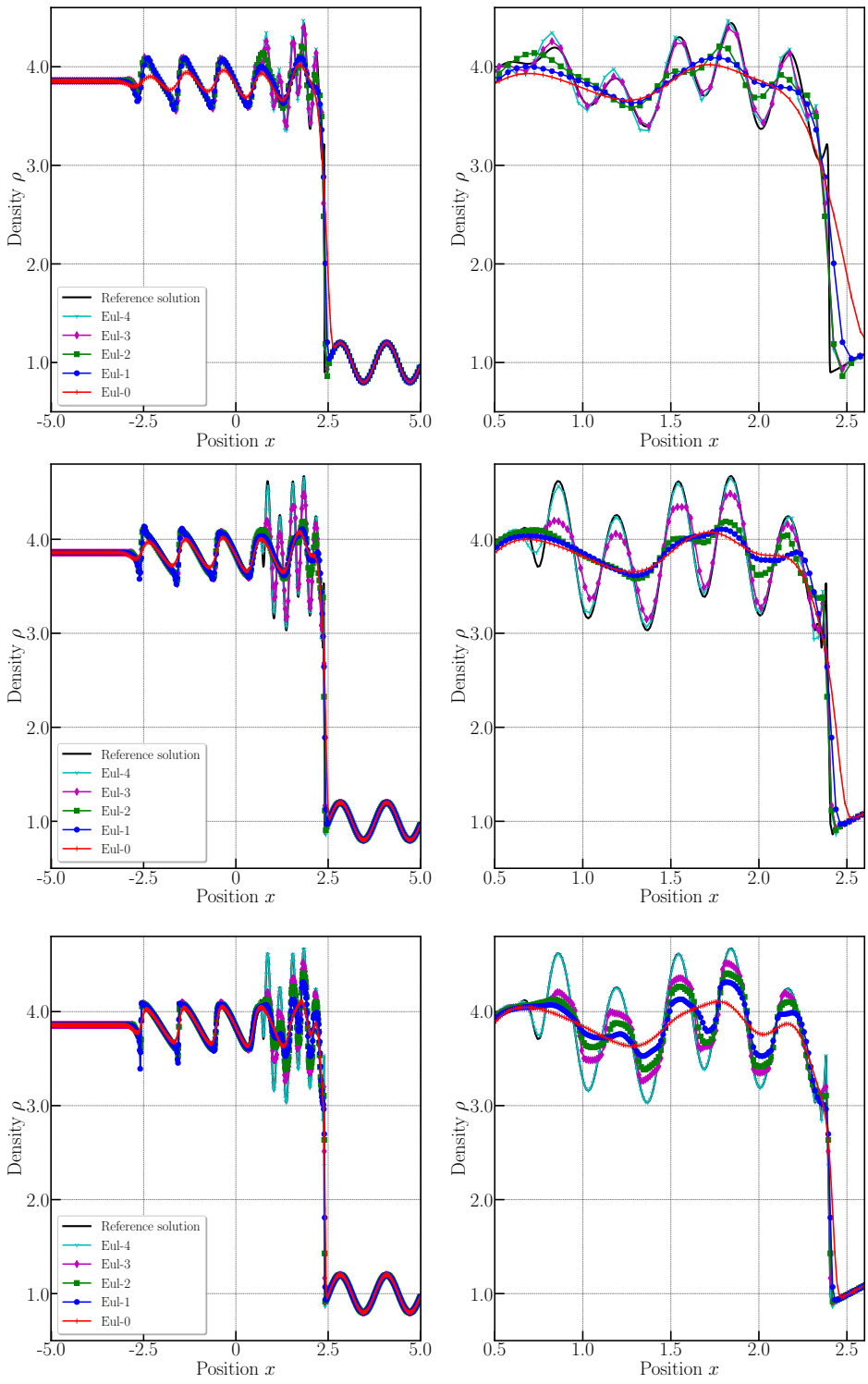


Figure 2.24: Shu-Osher oscillatory test case — High-order extension of the Eulerian numerical scheme with a mesh of $N = 200$ (top), $N = 400$ (middle) and $N = 800$ (bottom) — Evolution of the polynomial reconstruction degree d of each cell with each time step — Black cells are cells decremented to $d = 0$, red cells are cells decremented to $d = 1$, blue cells are cells decremented to $d = 2$, green cells are cells decremented to $d = 3$ and white cells are cells that stayed at $d = 4$.

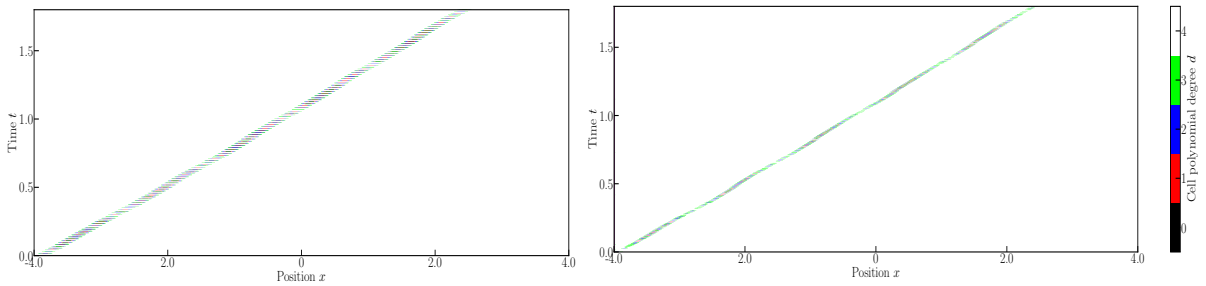


Figure 2.25: Shu-Osher oscillatory test case — High-order extension of the Eulerian numerical scheme with a mesh of$N = 200$ (left), $N = 400$ (right) — Evolution of the polynomial reconstruction degree d of each cell with each time step — Black cells are cells decremented to $d = 0$, red cells are cells decremented to $d = 1$, blue cells are cells decremented to $d = 2$, green cells are cells decremented to $d = 3$ and white cells are cells that stayed at $d = 4$.

reference solution taken is the numerical solution computed with the Lagrangian solver. Once more, we confirm the mesh convergence and the accuracy convergence as the order of the polynomial reconstruction increases. At last in figure 2.27 we present the polynomial degree used in each cell at each time-step for $N = 200$ (left) and 800 cells (right). It can be observed that the main waves are followed by the lower polynomial degree (black and red) while the maximal accuracy (white) is maintained away from them as expected.
Then in table 2.9 we present the percentages of cells updated with each polynomial degree from 0 to 3 and for three different meshes. We observe that about $85 - 93\%$ of cells are updated with the highest order scheme, the remaining $7 - 15\%$ left are updated with the other schemes, mainly the parachute one. At last the CPU times show that the cost of second order scheme is about 5 times more expensive than a first order one (respectively 11 times and 16 times for third and fourth order ones).

	Percentage of cell reconstruction type (%)				**Comparison of CPU time** (ms)			
	$d = 0$	$d = 1$	$d = 2$	$d = 3$	$d = 0$	$d = 1$	$d = 2$	$d = 3$
N = 200	6.5	3.5	2.1	85.9	142	923	1062	2126
N = 400	7.5	2.3	0.7	89.75	289	1163	3178	4317
N = 800	0.88	3.75	2.74	92.63	510	2207	7836	8596

Table 2.9: Woodward-Collela blastwave— High-order extension of the Eulerian numerical scheme with 200, 400 and 800 cells respectively — Percentage of cells with 1^{st}, 2^{nd}, 3^{rd} and 4^{th} order cell reconstruction (Left) — Comparison of the CPU time with 1^{st}, 2^{nd}, 3^{rd} and 4^{th} order cell reconstruction for 200, 400 and 800 cells (Right).

2.4 Chapter summary

Following the one-dimensional methodology introduced in the pioneering work of Gallice (2003), we have constructed a Lagrangian simple approximate Riemann solver which preserves contact discontinuities. The monitoring of wave speeds allows to derive explicit conditions for ensuring positivity and entropy stability. The resulting one-dimensional

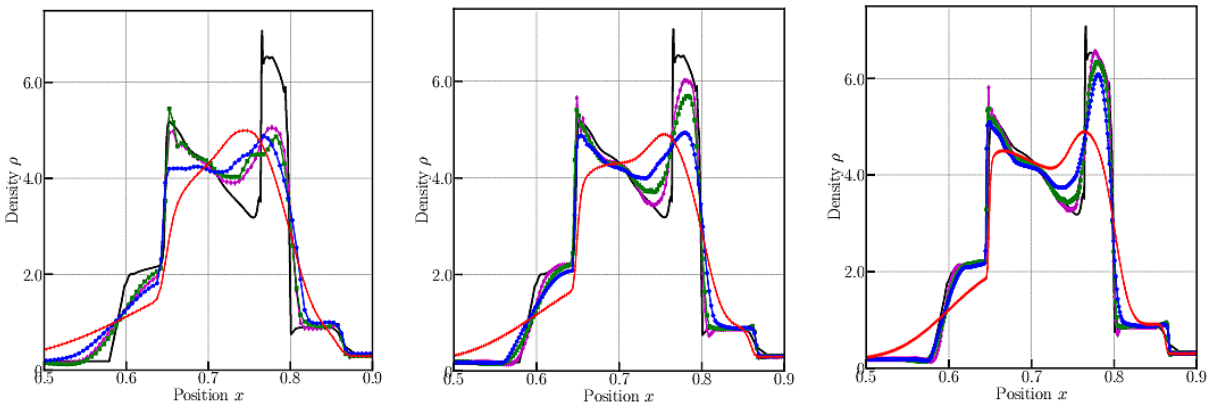


Figure 2.26: Woodward-Collela blastwave — High-order MOOD extension of the Eulerian numerical scheme with a mesh of $N = 200$ (left), $N = 400$ (middle) and $N = 800$ (right) — Numerical density of the Eul-0 (red cross), Eul-1 (blue circle), Eul-2 (green square) and Eul-3 (magenta diamond) compared with the reference solution (black line)

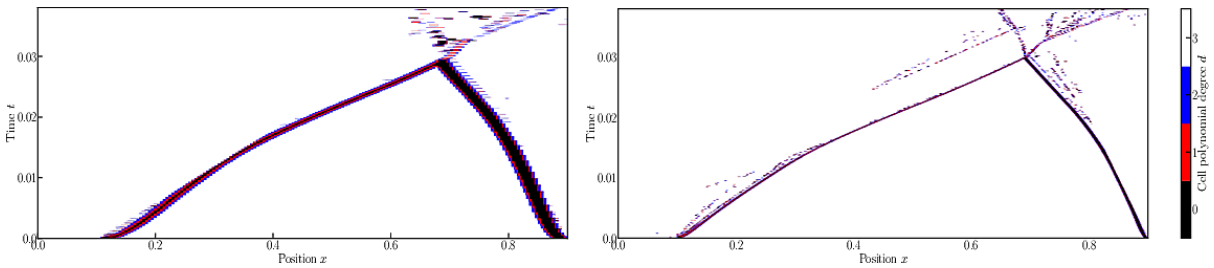


Figure 2.27: Woodward-Collela blastwave — High-order extension of the Eulerian numerical scheme with 200 (left) and 800 (right) cell — Evolution of the polynomial reconstruction degree d of each cell with each time step — Black cells are cells decremented to $d = 0$, red to $d = 1$, blue to $d = 2$, white cells are cells at highest accuracy $d = 3$.

Godunov-type Finite Volume scheme preserves the positivity of specific volume and internal energy and also satisfies an entropic inequality provided an explicit time step condition is satisfied.

Employing the general formalism described in Gallice (2002*a*, 2003) an one-dimensional Eulerian simple approximate Riemann solver is deduced from its Lagrangian counterpart. In this framework, the Eulerian wave speeds are deduced from the Lagrangian ones and are naturally ordered provided that the Lagrangian approximate Riemann solver preserves the positivity of specific volume. Moreover, this framework provides also the transfer of the good properties (positivity and entropy stability) of the Lagrangian approximate Riemann solver to its Eulerian counterpart. We observe that the proposed approximate Riemann solver is nothing but the one initially introduced in Gallice (2002*a*, 2003) in a more somewhat general and theoretical context. Here, we proposed a less formal derivation based on more intuitive arguments. We also want to acknowledge that the numerical flux induced by present Eulerian approximate solver has the same structure as the one induced by the famous HLLC solver. However, the main difference relies on the fact that here the wave speeds are consistently derived and ordered. This positive and entropic Eulerian simple approximate Riemann solver is the cornerstone upon which we build a positive and entropic Godunov-type Finite Volume scheme provided an adhoc time step condition is fulfilled.

The robustness and the accuracy of the basic first-order Finite Volume schemes are assessed against various classical and demanding numerical tests. The results achieved are pertinent *i.e.*, the positivity-preserving and entropy-stability properties are correctly enforced, and mesh convergence was also achieved. Furthermore, we note that the positivity-preserving property is mainly enforced during compression, thus, the trajectory of shock waves can be tracked along with the evolution of the waves speeds.

Due to its good properties, the first-order Eulerian scheme is then used in a last part as a parachute scheme for a high-order extension under MOOD paradigm. A high-order cell polynomial reconstruction in space, the Runge-Kutta method in time and the MOOD *a posteriori* limiting is implemented. The high-order extension has similar positivity-preserving property as the first-order numerical scheme while improving the numerical accuracy. The nominal accuracy is extended up to 5th order and may be decremented to the 1st order when needed. Two test cases with smooth solutions are first presented as a proof of rate of convergence and to validate the cell polynomial reconstruction. On the advanced test cases presenting complex wave interactions, the numerical results improve with the order of the reconstruction and the MOOD limiting ensures the robustness of the scheme.

The next chapter focuses on multidimensional Godunov-type Finite Volume method where the one-dimensional Lagrangian scheme developed in this chapter will be used as a starting point to construct the approximation for the multidimensional numerical flux.

3 Multidimensional Godunov-type Finite Volume scheme: Theory

Contents

This chapter is based on the work presented in G. Gallice & Maire (2022). In this chapter, we will be describing a novel subface-based Finite Volume method for discretizing multidimensional hyperbolic systems of conservation laws on general unstructured grids. Classically, multidimensional solvers make use of the one-dimensional ones and implemented in the normal direction of an interface. By doing so, the solver only takes into account the face-based neighboring cells, neglecting information from the transversal neighbors. Furthermore, these classical solvers invokes some numerical instabilities, such as the odd-even decoupling or the carbuncle phenomena (refer to Quirk (1994)). In this work, we will be introducing an original multidimensional numerical method that will be subface-based where subfaces are the decomposition of a face by means of the partition of a cell.

The big picture of the method is that the discretization will now be based on a subcell and requires a subface flux. The subface flux numerical approximation will rely on the notion of simple Eulerian Riemann solver introduced in the seminal work of Gallice (2003), adding to it an extra parameter, i.e. the nodal velocity. The Eulerian Riemann solver is constructed from its Lagrangian counterpart by means of the Lagrange-to-Euler mapping. This systematic procedure ensures the transfer of good properties such as positivity preservation and entropy stability. An entropy inequality as well as the flux entropy can be identified. A specific time step condition can also be derived that guarantees the solution in the mesh at time $t^n + \Delta t$. The subface-based discretization also calls for a

specific conservation condition, namely the node-based conservation condition to ensure conservativity.

Therefore, the aim of this chapter is to:

- Formulate a generic Eulerian multidimensional Finite Volume scheme for hyperbolic systems on unstructured grids based on subface fluxes that guarantees good properties;
- Recall the Lagrange-to-Euler mapping and establish the fundamental relation to relate the Eulerian and Lagrangian frameworks;
- Recalling the node-based conservation condition to retrieve conservativity of the numerical scheme;
- Develop a first-order Eulerian Godunov-type scheme with positivity preserving and entropy stable properties under an explicit CFL condition which ensures that the cell-averaged value at time $t + dt$ is a convex combination of the cell-average value at time t plus the intermediate states of the simple Riemann solver.

3.1 Governing equations and notation

Governing equations We aim at designing multidimensional Finite Volume schemes for solving the hyperbolic system of conservation laws

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbb{F}(\mathbf{U}) = \mathbf{0}. \tag{3.1.1}$$

Here, $\mathbf{U} = \mathbf{U}(\mathbf{x}, t)$, for $\mathbf{x} \in \mathbb{R}^{\mathrm{d}}$ and $t \geq 0$, is the vector of conservative variables which takes values in $\mathbb{R}^{\mathrm{q}}$ and $\mathbb{F} = \mathbb{F}(\mathbf{U})$ is the flux tensor in $\mathbb{R}^{\mathrm{q}} \times \mathbb{R}^{\mathrm{d}}$. The positive integers d and q denote respectively the space dimension and the size of the foregoing hyperbolic system. Let $\mathbf{e}_j$, for $j = 1 \ldots \mathrm{d}$, be the j-th vector of the Cartesian basis of $\mathbb{R}^{\mathrm{d}}$ then $\mathbf{F}_j = \mathbb{F}\mathbf{e}_j \in \mathbb{R}^{\mathrm{q}}$ is the j-th component of the tensor flux and its divergence writes

$$\nabla \cdot \mathbb{F}(\mathbf{U}) = \sum_{j=1}^{\mathrm{d}} \frac{\partial \mathbf{F}_j}{\partial x_j}.$$

We assume that system (3.1.1) is equipped with the entropy-entropy flux pair $(\Sigma, \mathbf{Q})$. Namely, $\mathbf{U} \mapsto \Sigma(\mathbf{U})$ is convex and the following compatibility condition holds true for the entropy flux (refer to Godlewski & Raviart (1996)),

$$\left(\frac{\partial \mathbf{F}_j}{\partial \mathbf{U}}\right)^t \frac{\partial \Sigma}{\partial \mathbf{U}} = \frac{\partial Q_j}{\partial \mathbf{U}}, \text{ for } j = 1 \ldots \mathrm{d},$$

where $Q_j = \mathbf{Q} \cdot \mathbf{e}_j$. We are looking for the entropic solutions of (3.1.1), that is the ones satisfying the following entropy inequality

$$\frac{\partial \Sigma}{\partial t} + \nabla \cdot \mathbf{Q}(\mathbf{U}) \leq 0, \tag{3.1.2}$$

which turns into an equality for smooth solutions.

Let $\mathcal{D} \subset \mathbb{R}^q$ be the domain of definition of $\mathbb{F}(\mathbf{U})$, $\Sigma(\mathbf{U})$ and $\mathbf{Q}(\mathbf{U})$. We ase that $\mathcal{D}$ is a convex subset of $\mathbb{R}^q$. For the sake of simplicity, we shall limit the presentation of the numerical methods to the two-dimensional case, *i.e.*, d $= 2$, knowing that the three-dimensional extension is quite straightforward.

Notations The computational domain is a polygonal portion of $\mathbb{R}^2$, and we pave it with a collection of non overlapping polygonal cells ω_c where c is the generic label of the cell. The generic label of a point is p and $\mathbf{x}_p$ denotes its vector position. In the counterclockwise ordered list of points of ω_c, p^- and p^+ are respectively the previous and the next points with respect to p, refer to figure 3.1. Each face f of cell c is decomposed into subfaces by means of the partition of c induced by the subcells pc. The subcell ω_{pc} related to cell c and point p is the quadrangle formed by joining the cell centroid, $\mathbf{x}_c$, to the midpoints of $[\mathbf{x}_{p^-}, \mathbf{x}_p]$, $[\mathbf{x}_p, \mathbf{x}_{p^+}]$ and to $\mathbf{x}_p$.

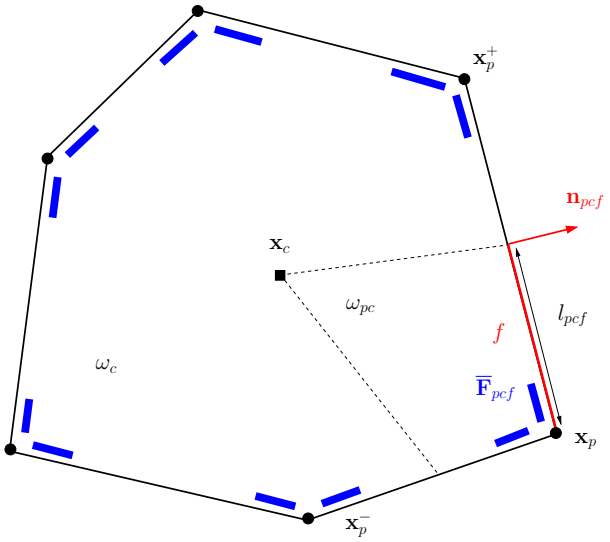


Figure 3.1: Geometrical entities attached to the polygonal cell ω_c.

The following topological sets are also defined:

- $\mathcal{P}(c)$: The set of vertices (points) of ω_c;
- $\mathcal{F}(c)$: The set of faces of cell ω_c;
- $\mathcal{SF}(pc)$: The set of subfaces attached to the corner pc, which is nothing but the set of faces of subcell ω_{pc} impinging at point p, for instance $[\mathbf{x}_p, \frac{1}{2}(\mathbf{x}_p + \mathbf{x}_{p^+})]$ belongs to $\mathcal{SF}(pc)$.

- The set of subcells ω_{pc} for $p \in \mathcal{P}(c)$ constitutes a partition of the cell ω_c, such that

$$\omega_c = \bigcup_{p\in\mathcal{P}(c)} \omega_{pc};$$

- The set of subfaces $\mathcal{SF}(pc)$ for $p \in \mathcal{P}(c)$ constitutes a partition of the set of faces of ω_c, that is,

$$\mathcal{F}(c) = \bigcup_{p\in\mathcal{P}(c)} \mathcal{SF}(pc).$$

Lastly, we denote respectively by l_{pcf} and $\mathbf{n}_{pcf}$ the measure and the unit outward normal of the subface f.

3.2 A generic Godunov-type Finite Volume scheme

With the aforementioned governing equations and notations, this sections presents an original Godunov-type Finite Volume scheme for solving hyperbolic systems of conservation laws on unstructured grids in a generic form.

3.2.1 Subface-based Finite Volume scheme

We start by integrating the system of conservation laws (3.1.1) over ω_c and employing the Green formula leads to

$$|\omega_c|\frac{\mathrm{d}\mathbf{U}_c}{\mathrm{d}t} + \int_{\partial\omega_c} \mathbb{F}(\mathbf{U})\mathbf{n}\,\mathrm{d}s = \mathbf{0}, \tag{3.2.1}$$

where $\mathbf{U}_c(t) = \frac{1}{|\omega_c|}\int_{\omega_c} \mathbf{U}(\mathbf{x},t)\,\mathrm{d}v$ is the cell-averaged value of $\mathbf{U}$ over ω_c. Employing a classical first-order explicit time integration turns (3.2.1) into

$$\mathbf{U}_c^{n+1} - \mathbf{U}_c^n + \frac{\Delta t}{|\omega_c|}\int_{\partial\omega_c} \mathbb{F}(\mathbf{U}^n)\mathbf{n}\,\mathrm{d}s = \mathbf{0}. \tag{3.2.2}$$

Here, $\mathbf{U}_c^n$ denotes the approximation of $\mathbf{U}_c(t)$ at time t^n, and, $t^{n+1} = t^n + \Delta t$ where $\Delta t > 0$ is the time step. The design of the Finite Volume scheme (3.2.2) requires the construction of an approximation of the normal flux integral $\int_{\partial\omega_c} \mathbb{F}(\mathbf{U}^n)\mathbf{n}\,\mathrm{d}s$. In what follows, we are going to define an original node-based approximation of this integral term relying on the partition of ω_c into subcells, that is,

$$\int_{\partial\omega_c} \mathbb{F}(\mathbf{U}^n)\mathbf{n}\,\mathrm{d}s = \sum_{p\in\mathcal{P}(c)} \int_{\partial\omega_{pc}\cap\partial\omega_c} \mathbb{F}(\mathbf{U}^n)\mathbf{n}\,\mathrm{d}s. \tag{3.2.3}$$

The surface integral term at the right-hand side of (3.2.3) is approximated along the subfaces as follows

$$\int_{\partial\omega_{pc}\cap\partial\omega_c} \mathbb{F}(\mathbf{U}^n)\mathbf{n}\,\mathrm{d}s = \sum_{f\in\mathcal{SF}(pc)} l_{pcf}\overline{\mathbf{F}}_{pcf},$$

where $\overline{\mathbf{F}}_{pcf}$ is the subface flux related to the subface f attached to the corner pc. Substituting this subface-based approximation of the flux into (3.2.2) yields the subface-based generic Finite Volume scheme

$$\mathbf{U}_c^{n+1} - \mathbf{U}_c^n + \frac{\Delta t}{|\omega_c|}\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} l_{pcf}\overline{\mathbf{F}}_{pcf} = \mathbf{0}, \tag{3.2.4}$$

which is characterized by the subface flux $\overline{\mathbf{F}}_{pcf}$ displayed in figure 3.1 by the blue rectangles. Viewed from cell ω_c, this peculiar Finite Volume discretization introduces two subface fluxes per cell face. We observe that this type of Finite Volume discretization which consists in splitting the faces into subfaces has been already utilized not only in the framework of cell-centered Lagrangian hydrodynamics from Després & Mazeran (2005); Maire (2009); Loubère *et al.* (2016) but also for developing cell-centered diffusion schemes such as done by Maire & Breil (2012); Jacq *et al.* (2014). We note in passing that this formalism encompasses the classical face-based Finite Volume discretization Godlewski & Raviart (1996).

At this point, it remains to provide a consistent numerical approximation of the subface flux and this will be done by means of an approximate Riemann solver.

3.2.2 Subface flux approximation

Let f be the generic subface attached to cell c and vertex p, characterized by its unit outward normal $\mathbf{n}_{pcf}$. We assume that the subface flux $\overline{\mathbf{F}}_{pcf}$ attached to the subcell f depends respectively on the adjacent cell averaged values $\mathbf{U}_c$, $\mathbf{U}_{d(c,f)}$, where $d(c,f)$ is the neighbor of cell c such that $f\subset(\omega_c\cap\omega_d)$, on the unit normal $\mathbf{n}_{pcf}$ and also on a vectorial parameter $\mathbf{v}_p\in\mathbb{R}^2$ related to the nodal velocity p. This leads us to write

$$\overline{\mathbf{F}}_{pcf} = \overline{\mathbf{F}}_{pcf}(\mathbf{U}_c, \mathbf{U}_{d(c,f)}, \mathbf{n}_{pcf}, \mathbf{v}_p). \tag{3.2.5}$$

Contrary to the classical face-based Finite Volume discretization, the foregoing subface flux expression exhibits a dependency on the nodal vectorial parameter $\mathbf{v}_p$, which is unknown for the moment. Such dependency has been already employed for designing flux approximation dedicated to the cell-centered discretization of Lagrangian hydrodynamics in Maire (2009) and also more recently, in-spired by the latter works, in the framework of cell-centered Eulerian hydrodynamics Shen *et al.* (2014). In both cases, the parameter vector, $\mathbf{v}_p$, corresponds to a nodal approximation of the velocity field. For Lagrangian hydrodynamics discretization, the nodal velocity approximation is required to move the computational grid, whereas for Eulerian hydrodynamics its need is less obvious. The fundamental role of this nodal vector parameter z will be described later on in 3.4.

Bearing this in mind, the subface flux numerical approximation is constructed via the

introduction of an approximate Riemann solver, which is nothing but the approximate solution of the one-dimensional Riemann problem defined in the $\mathbf{n}_{pcf}$ direction

$$(\mathcal{RP}) \begin{cases} \dfrac{\partial \mathbf{U}}{\partial t} + \dfrac{\partial [\mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U})]}{\partial x_{\mathbf{n}_{pcf}}} &= \mathbf{0}, \\ \mathbf{U}(x_{\mathbf{n}_{pcf}}, 0) &= \begin{cases} \mathbf{U}_c & \text{if } x_{\mathbf{n}_{pcf}} < 0, \\ \mathbf{U}_{d(c,f)} & \text{if } x_{\mathbf{n}_{pcf}} \geq 0. \end{cases} \end{cases}$$

Here, $x_{\mathbf{n}_{pcf}} = \mathbf{x} \cdot \mathbf{n}_{pcf}$ is the coordinate in the unit normal direction, $\mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}) = \mathbb{F}(\mathbf{U})\mathbf{n}_{pcf}$ is the projection of the tensor flux onto the unit normal direction. The resulting one-dimensional approximate Riemann solver depends on the states on both sides of the interface, the self-similar variable $\xi = \dfrac{x_{\mathbf{n}_{pcf}}}{t}$ and a nodal vector parameter. With these arguments, the Riemann solver writes

$$\mathbf{W}_{pcf} = \mathbf{W}_{pcf}(\mathbf{U}_c, \mathbf{U}_{d(c,f)}, \mathbf{n}_{pcf}, \xi, \mathbf{v}_p). \tag{3.2.6}$$

For all $\mathbf{U}_l$ (left state), $\mathbf{U}_r$ (right state), $\mathbf{n}$ (unit normal), ξ (self-similar variable) and $\mathbf{v}_p$ (vector parameter), we assume that the Riemann solver satisfies the following classical properties

- $\mathbf{W}_{pcf}(\mathbf{U}_l, \mathbf{U}_r, \mathbf{n}, \xi, \mathbf{v}_p) = \mathbf{U}_l$ for $-\xi$ large enough;
- $\mathbf{W}_{pcf}(\mathbf{U}_l, \mathbf{U}_r, \mathbf{n}, \xi, \mathbf{v}_p) = \mathbf{U}_r$ for ξ large enough;
- $\mathbf{W}_{pcf}(\mathbf{U}, \mathbf{U}, \mathbf{n}, \xi, \mathbf{v}_p) = \mathbf{U}$.

We also make the assumption that the approximate Riemann solver is symmetric with respect to the interface, that is

$$\mathbf{W}_{pcf}(\mathbf{U}_c, \mathbf{U}_{d(c,f)}, \mathbf{n}_{pcf}, \xi, \mathbf{v}_p) = \mathbf{W}_{pdf}(\mathbf{U}_{d(c,f)}, \mathbf{U}_c, \mathbf{n}_{pcf}, -\xi, \mathbf{v}_p). \tag{3.2.7}$$

Finally, we express the subface flux $\overline{\mathbf{F}}_{pcf}$ in terms of the approximate Riemann solver $\mathbf{W}_{pcf}$ as follows

$$\overline{\mathbf{F}}_{pcf} = \mathbb{F}(\mathbf{U}_c)\mathbf{n}_{pcf} - \int_{-\infty}^{0} \left[\mathbf{W}_{pcf}(\mathbf{U}_c, \mathbf{U}_{d(c,f)}, \mathbf{n}_{pcf}, \xi, \mathbf{v}_p) - \mathbf{U}_c \right] \mathrm{d}\xi. \tag{3.2.8}$$

The subface flux approximation is obtained by integrating the conservation law of $(\mathcal{RP})$ over the space-time domain $[-\Delta x_l, 0] \times [0, \Delta t]$ where Δx_l and Δt are respectively space and time increments and the formula could be found in Harten *et al.* (1983).

Remark 3.2.1. *This choice to define the subface flux from the approximate Riemann solver is original and crucial. We note in passing that the aforementioned works employing subface flux-based discretization do not rely on such a definition. Here, the subface flux* (3.2.8) *might be viewed as a left-sided flux and there is absolutely no reason why* $\overline{\mathbf{F}}_{pcf} = \overline{\mathbf{F}}_{pdf}$*, where* $\overline{\mathbf{F}}_{pdf}$ *is the subface flux attached to subface* f *viewed from cell*

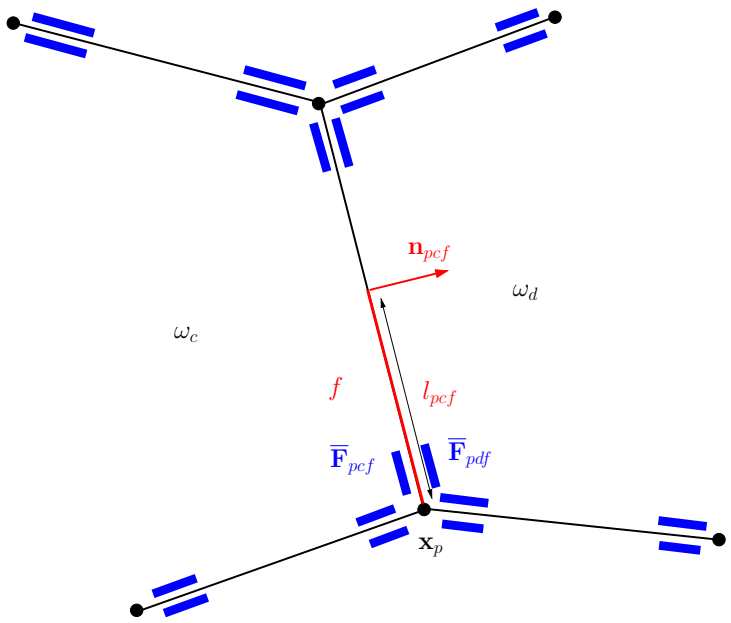


Figure 3.2: Interface between cell ω_c and cell ω_d.

$d = d(c, f)$, refer to figure 3.2. Indeed, in general we should have $\overline{\mathbf{F}}_{pcf} \neq \overline{\mathbf{F}}_{pdf}$ this implies that the Finite Volume scheme (3.2.9) characterized by subface flux (3.2.8) is not conservative in the classical sense. We shall present in section 3.2.6 a novel framework to study the conservativity of this subface-based Finite Volume method.

3.2.3 Preservation of the definition domain

The notion of invariant domain is classical in the context of hyperbolic systems of conservation laws, refer for instance to Bouchut (2004). A domain is invariant if for any initial condition $\mathbf{U}^0$ belonging to the domain, the solution of the hyperbolic system under consideration remains in the domain for all time $t > 0$. The most practical situation corresponds to the case for which the invariant domain is convex. For instance, in the case of Lagrangian gas dynamics $\mathbf{U} = (\tau, \mathbf{v}, e)^t$, where τ is the specific volume, $\mathbf{v}$ is the velocity vector and e the total energy, and the definition domain (admissible set) $\mathcal{D} = \{\mathbf{U} \text{ such that } \tau \geq 0 \text{ and } e - \frac{1}{2}\mathbf{v}^2 \geq 0\}$ is convex. In what follows, we assume that the definition domain $\mathcal{D}$ of $\mathbb{F}(\mathbf{U})$, $\Sigma(\mathbf{U})$ and $\mathbf{Q}(\mathbf{U})$ is convex and we shall study under which condition our Finite Volume discretization preserves the definition domain.

The studied Finite Volume scheme (3.2.4) writes under the form

$$\mathbf{U}_c^{n+1} - \mathbf{U}_c^n + \frac{\Delta t}{|\omega_c|} \sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{SF}(pc)} l_{pcf} \left[\overline{\mathbf{F}}_{pcf} - \mathbb{F}(\mathbf{U}_c^n)\mathbf{n}_{pcf} \right] = \mathbf{0}, \tag{3.2.9}$$

where we have made use of the geometric identity

$$\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} l_{pcf}\mathbf{n}_{pcf} = \mathbf{0}, \tag{3.2.10}$$

to make appear the fluctuations related to each subface.

Assuming that the approximate Riemann solver is $\mathcal{D}$-preserving, that is, if $\mathbf{U}_c^n \in \mathcal{D}$ then $\mathbf{W}_{pcf}(\xi) \in \mathcal{D}$ for all $\xi \in \mathbb{R}$, we shall determine the time step condition ensuring that the foregoing Finite Volume scheme is itself $\mathcal{D}$-preserving, that is, $\mathbf{U}_c^{n+1} \in \mathcal{D}$. To this end, we introduce $\xi_{pcf}^{\min} \geq 0$ such that

$$\mathbf{W}_{pcf}(\xi) = \mathbf{U}_c^n, \quad for \quad \xi < -\xi_{pcf}^{\min}.$$

This allows us to develop the subface flux expression as follows

$$\overline{\mathbf{F}}_{pcf} = \mathbb{F}(\mathbf{U}_c^n)\mathbf{n}_{pcf} - \int_{-\xi_{pcf}^{\min}}^{0} (\mathbf{W}_{pcf}(\xi) - \mathbf{U}_c^n)\,\mathrm{d}\xi = \mathbb{F}(\mathbf{U}_c^n)\mathbf{n}_{pcf} + \xi_{pcf}^{\min}\mathbf{U}_c^n - \int_{-\xi_{pcf}^{\min}}^{0} \mathbf{W}_{pcf}(\xi)\,\mathrm{d}\xi.$$

Substituting the subface flux into the foregoing Finite Volume scheme yields

$$\mathbf{U}_c^{n+1} = \mathbf{U}_c^n - \frac{\Delta t}{|\omega_c|}\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} l_{pcf}\left[\xi_{pcf}^{\min}\mathbf{U}_c^n - \int_{-\xi_{pcf}^{\min}}^{0} \mathbf{W}_{pcf}(\xi)\,\mathrm{d}\xi\right].$$

Now, collecting the terms in factor of $\mathbf{U}_c^n$ we arrive at

$$\mathbf{U}_c^{n+1} = \left(1 - \frac{\Delta t}{|\omega_c|}\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} l_{pcf}\xi_{pcf}^{\min}\right)\mathbf{U}_c^n + \frac{\Delta t}{|\omega_c|}\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} l_{pcf}\int_{-\xi_{pcf}^{\min}}^{0} \mathbf{W}_{pcf}(\xi)\,\mathrm{d}\xi. \tag{3.2.11}$$

Assuming that $\mathbf{U}_c^n \in \mathcal{D}$ and $\mathbf{W}_{pcf}(\xi) \in \mathcal{D}$ then $\mathbf{U}_c^{n+1} \in \mathcal{D}$ provided that the time step satisfies the condition

$$\Delta t \leq \frac{|\omega_c|}{\displaystyle\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} l_{pcf}\xi_{pcf}^{\min}}. \tag{3.2.12}$$

In this case, we observe that $\mathbf{U}_c^{n+1}$ is nothing but a convex combination of $\mathbf{U}_c^n$ and the intermediate states of the subface-based approximate Riemann solvers. Introducing

$$\Delta t_c = \frac{|\omega_c|}{\displaystyle\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} l_{pcf}\xi_{pcf}^{\min}},$$

and assuming that the Riemann solver preserves the domain of definition, we claim that the Finite Volume scheme is $\mathcal{D}$-preserving under the global time-step condition

$$\Delta t \leq \min_c \Delta t_c. \tag{3.2.13}$$

3.2.4 Entropy inequality

Here, we derive the entropy flux approximation attached to our Finite Volume scheme. Assuming that $\mathbf{U}_c^n \in \mathcal{D}$, $\mathbf{W}_{pcf}(\xi) \in \mathcal{D}$ and Δt satisfies the time step condition (3.2.12), then by virtue of (3.2.11), $\mathbf{U}_c^{n+1}$ appears to be a convex combination of $\mathbf{U}_c^n$ and $\frac{1}{\xi_{pcf}^{\min}} \int_{-\xi_{pcf}^{\min}}^{0} \mathbf{W}_{pcf}(\xi)\, \mathrm{d}\xi$, which thus belongs to $\mathcal{D}$. Under the foregoing assumptions and thanks to the convexity of the entropy, $\Sigma_c^{n+1} = \Sigma(\mathbf{U}_c^{n+1})$ satisfies

$$\begin{aligned} \Sigma_c^{n+1} \leq & \left(1 - \frac{\Delta t}{|\omega_c|} \sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{SF}(pc)} l_{pcf} \xi_{pcf}^{\min}\right) \Sigma_c^n \\ & + \frac{\Delta t}{|\omega_c|} \sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{SF}(pc)} l_{pcf} \xi_{pcf}^{\min} \Sigma \left(\frac{1}{\xi_{pcf}^{\min}} \int_{-\xi_{pcf}^{\min}}^{0} \mathbf{W}_{pcf}(\xi)\, \mathrm{d}\xi\right). \end{aligned} \tag{3.2.14}$$

By virtue of Jensen inequality

$$\Sigma \left(\frac{1}{\xi_{pcf}^{\min}} \int_{-\xi_{pcf}^{\min}}^{0} \mathbf{W}_{pcf}(\xi)\, \mathrm{d}\xi\right) \leq \frac{1}{\xi_{pcf}^{\min}} \int_{-\xi_{pcf}^{\min}}^{0} \Sigma(\mathbf{W}_{pcf}(\xi))\, \mathrm{d}\xi.$$

Substituting the foregoing result into (3.2.14), we arrive at

$$\Sigma_c^{n+1} - \Sigma_c^n \leq \frac{\Delta t}{|\omega_c|} \sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{SF}(pc)} l_{pcf} \int_{-\infty}^{0} (\Sigma(\mathbf{W}_{pcf}(\xi)) - \Sigma_c^n)\, \mathrm{d}\xi. \tag{3.2.15}$$

Here, we have used the fact that if $\xi \leq \xi_{pcf}^{\min}$ then $\mathbf{W}_{pcf}(\xi) = \mathbf{U}_c^n$. Finally, utilizing the geometric identity (3.2.10) we get

$$\sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{SF}(pc)} l_{pcf} \mathbf{Q}(\mathbf{U}_c^n) \cdot \mathbf{n}_{pcf} = 0.$$

Introducing the foregoing expression into (3.2.15) leads to

$$\Sigma_c^{n+1} - \Sigma_c^n + \frac{\Delta t}{|\omega_c|} \sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{SF}(pc)} l_{pcf} \left[\mathbf{Q}(\mathbf{U}_c^n) \cdot \mathbf{n}_{pcf} - \int_{-\infty}^{0} (\Sigma(\mathbf{W}_{pcf}(\xi)) - \Sigma(\mathbf{U}_c^n))\, \mathrm{d}\xi\right] \leq 0. \tag{3.2.16}$$

This is formally the discrete counterpart of the continuous entropy inequality (3.1.2). The comparison between the foregoing inequality and (3.1.2) incites us to define the subface entropy flux

$$\overline{Q}_{pcf} = \mathbf{Q}(\mathbf{U}_c^n) \cdot \mathbf{n}_{pcf} - \int_{-\infty}^{0} (\Sigma(\mathbf{W}_{pcf}(\xi)) - \Sigma(\mathbf{U}_c^n))\, \mathrm{d}\xi. \tag{3.2.17}$$

With this notation, we rewrite inequality (3.2.16) under the compact form

$$\Sigma_c^{n+1} - \Sigma_c^n + \frac{\Delta t}{|\omega_c|} \sum_{p\in\mathcal{P}(c)} \sum_{f\in\mathcal{SF}(pc)} l_{pcf} \overline{Q}_{pcf} \leq 0. \tag{3.2.18}$$

Remark 3.2.2. *At this stage nothing can be said regarding the entropy stability of the Finite Volume scheme under consideration. This crucial point shall be investigated in section 3.2.7.*

3.2.5 Summary about the subface-based Finite Volume scheme

We have designed a subface-based Finite Volume scheme for which the subface flux $\overline{\mathbf{F}}_{pcf}$ is defined from the approximate Riemann solver $\mathbf{W}_{pcf}$. Under an explicit time step condition this Finite Volume scheme is $\mathcal{D}$-preserving and satisfies a formal entropy inequality. We summarize hereafter the main characteristics of this Finite Volume scheme.

- The generic Finite Volume scheme reads

$$\mathbf{U}_c^{n+1} - \mathbf{U}_c^n + \frac{\Delta t}{|\omega_c|} \sum_{p\in\mathcal{P}(c)} \sum_{f\in\mathcal{SF}(pc)} l_{pcf} \overline{\mathbf{F}}_{pcf} = \mathbf{0},$$

  where the subface flux expression in terms of the approximate Riemann solver $\mathbf{W}_{pcf}$ writes

$$\overline{\mathbf{F}}_{pcf} = \mathbb{F}(\mathbf{U}_c^n)\mathbf{n}_{pcf} - \int_{-\infty}^{0} (\mathbf{W}_{pcf}(\xi) - \mathbf{U}_c^n)\, \mathrm{d}\xi.$$

- The time step condition to ensure that the FV scheme is $\mathcal{D}$-preserving reads

$$\Delta t \leq \Delta t_c = \frac{|\omega_c|}{\displaystyle\sum_{p\in\mathcal{P}(c)} \sum_{f\in\mathcal{SF}(pc)} l_{pcf} \xi_{pcf}^{\min}}.$$

- The formal entropy inequality attached to the FV scheme under the foregoing time step condition reads

$$\Sigma_c^{n+1} - \Sigma_c^n + \frac{\Delta t}{|\omega_c|} \sum_{p\in\mathcal{P}(c)} \sum_{f\in\mathcal{SF}(pc)} l_{pcf} \overline{Q}_{pcf} \leq 0,$$

  where the corner entropy flux writes

$$\overline{Q}_{pcf} = \mathbf{Q}(\mathbf{U}_c^n) \cdot \mathbf{n}_{pcf} - \int_{-\infty}^{0} (\Sigma(\mathbf{W}_{pcf}(\xi)) - \Sigma(\mathbf{U}_c^n))\, \mathrm{d}\xi.$$

Now, it remains to investigate not only the conservation property of the subface-based Finite Volume scheme but also its entropy stability.

3.2.6 Conservation property

This section aims at determining under which conditions the studied subface-based Finite Volume scheme is conservative. Assuming that the computational domain is the whole space $\mathbb{R}^2$, the subface-based Finite Volume scheme,

$$|\omega_c|(\mathbf{U}_c^{n+1} - \mathbf{U}_c^n) + \Delta t \sum_{p\in\mathcal{P}(c)} \sum_{f\in\mathcal{SF}(pc)} l_{pcf}\overline{\mathbf{F}}_{pcf} = \mathbf{0},$$

is conservative if and only if

$$\sum_c |\omega_c|\mathbf{U}_c^{n+1} = \sum_c |\omega_c|\mathbf{U}_c^n \iff \sum_c \sum_{p\in\mathcal{P}(c)} \sum_{f\in\mathcal{SF}(pc)} l_{pcf}\overline{\mathbf{F}}_{pcf} = \mathbf{0}.$$

Now, exchanging the summation over the cells with the summation over the nodes, the right-sided condition turns into

$$\sum_p \sum_{c\in\mathcal{C}(p)} \sum_{f\in\mathcal{SF}(pc)} l_{pcf}\overline{\mathbf{F}}_{pcf} = \mathbf{0},$$

where $\mathcal{C}(p)$ is the set of cells sharing the point p.

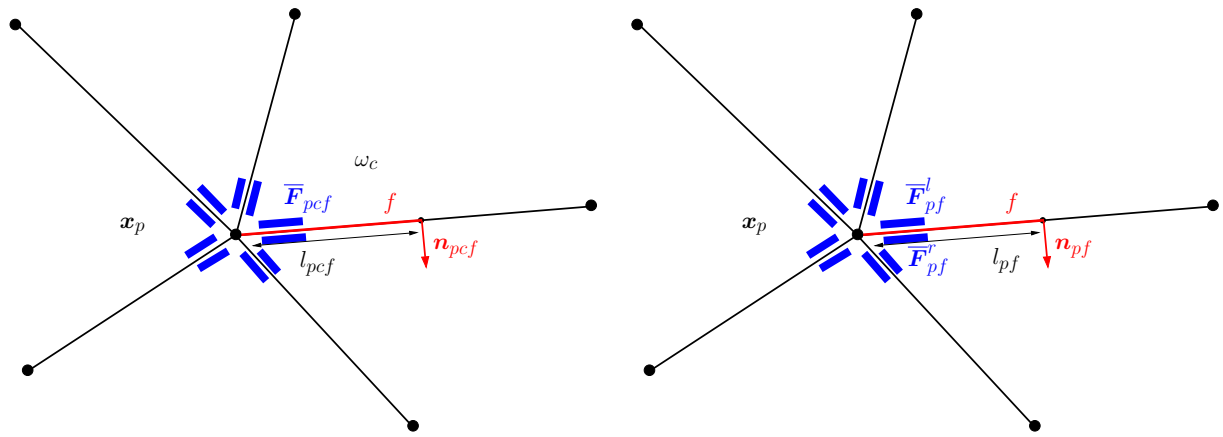
We claim that a sufficient condition to ensure the conservativity of the subface-based Finite Volume scheme writes

$$\sum_{c\in\mathcal{C}(p)} \sum_{f\in\mathcal{SF}(pc)} l_{pcf}\overline{\mathbf{F}}_{pcf} = \mathbf{0}. \tag{3.2.19}$$

This means that the summation over the cells c sharing point p of the fluxes attached to the subfaces impinging at p is equal to zero, refer to figure 3.3(a), where the subface fluxes have been displayed by means of blue patches on both sides of each subface emanating from point p. Indeed our numerical method is provably positivity preserving under the explicit CFL condition (3.2.13). Noticing that the sum over the cells c sharing p of the fluxes attached to the subfaces impinging at p is rigorously equal to the sum over the left-sided and the right-sided fluxes attached to the subfaces impinging at p leads to reformulate the sufficient condition (3.2.19) into

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}(\overline{\mathbf{F}}_{pf}^l + \overline{\mathbf{F}}_{pf}^r) = \mathbf{0}. \tag{3.2.20}$$

Here, $\mathcal{SF}(p)$ denotes the set of subfaces impinging at point p. For $f \in \mathcal{SF}(p)$, l_{pf} is the length of subface f and $\mathbf{n}_{pf}$ is its unit normal pointing towards the right state. In the foregoing equation, $\overline{\mathbf{F}}_{pf}^l$ (resp. $\overline{\mathbf{F}}_{pf}^r$) denotes respectively the left-sided (resp. right-sided) flux attached to the subface f, refer to figure 3.3(b). By virtue of (3.2.5), with an obvious notation adaptation, the left and right-sided fluxes expressions in terms of the left state



(a) Cell-based notation. (b) Face-based notation.

Figure 3.3: Fragment of the computational grid in the vicinity of point p.

$\mathbf{U}_{lf}$, right state $\mathbf{U}_{rf}$ and the unit normal $\mathbf{n}_{pf}$ and vector nodal parameter $\mathbf{v}_p$ read

$$\overline{\mathbf{F}}^l_{pf} = \overline{\mathbf{F}}^l_{pf}(\mathbf{U}_{lf}, \mathbf{U}_{rf}, \mathbf{n}_{pf}, \mathbf{v}_p), \text{ and } \overline{\mathbf{F}}^r_{pf} = \overline{\mathbf{F}}^r_{pf}(\mathbf{U}_{rf}, \mathbf{U}_{lf}, -\mathbf{n}_{pf}, \mathbf{v}_p).$$

Substituting this into the conservation condition (3.2.20) leads to

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}\left[\overline{\mathbf{F}}^l_{pf}(\mathbf{U}_{lf}, \mathbf{U}_{rf}, \mathbf{n}_{pf}, \mathbf{v}_p) + \overline{\mathbf{F}}^r_{pf}(\mathbf{U}_{rf}, \mathbf{U}_{lf}, -\mathbf{n}_{pf}, \mathbf{v}_p)\right] = \mathbf{0}. \tag{3.2.21}$$

Thanks to (3.2.8) and with an obvious notation adaptation, we express the subface fluxes in (3.2.21) in terms of the approximate Riemann solver

$$\overline{\mathbf{F}}^l_{pf}(\mathbf{U}_{lf}, \mathbf{U}_{rf}, \mathbf{n}_{pf}, \mathbf{v}_p) = \mathbb{F}(\mathbf{U}_{lf})\mathbf{n}_{pf} - \int_{-\infty}^{0} (\mathbf{W}_{pf}(\mathbf{U}_{lf}, \mathbf{U}_{rf}, \xi, \mathbf{v}_p) - \mathbf{U}_{lf})\,\mathrm{d}\xi,$$
$$\overline{\mathbf{F}}^r_{pf}(\mathbf{U}_{rf}, \mathbf{U}_{lf}, -\mathbf{n}_{pf}, \mathbf{v}_p) = \mathbb{F}(\mathbf{U}_{rf})(-\mathbf{n}_{pf}) - \int_{-\infty}^{0} (\mathbf{W}_{pf}(\mathbf{U}_{rf}, \mathbf{U}_{lf}, \xi, \mathbf{v}_p) - \mathbf{U}_{rf})\,\mathrm{d}\xi.$$

Substituting these expressions of the subface fluxes into the nodal conservation condition (3.2.21) and by virtue of the symmetry assumption of the approximate Riemann solver (3.2.7), *i.e.*, $\mathbf{W}_{pf}(\mathbf{U}_{lf}, \mathbf{U}_{rf}, \xi, \mathbf{v}_p) = \mathbf{W}_{pf}(\mathbf{U}_{rf}, \mathbf{U}_{lf}, -\xi, \mathbf{v}_p)$ we arrive at the node based condition:

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}\left[\int_{-\infty}^{0} (\mathbf{W}_{pf}(\mathbf{U}_{lf}, \mathbf{U}_{rf}, \xi, \mathbf{v}_p) - \mathbf{U}_{lf})\,\mathrm{d}\xi + \int_{0}^{\infty} (\mathbf{W}_{pf}(\mathbf{U}_{lf}, \mathbf{U}_{rf}, \xi, \mathbf{v}_p) - \mathbf{U}_{rf})\,\mathrm{d}\xi\right. \tag{3.2.22}$$
$$\left. + (\mathbb{F}(\mathbf{U}_{rf}) - \mathbb{F}(\mathbf{U}_{lf}))\,\mathbf{n}_{pf}\right] = \mathbf{0}.$$

The Finite Volume scheme under consideration is conservative provided that the foregoing

node-based condition is fulfilled for each node.

Remark 3.2.3. *The node-based condition* (3.2.21) *is not only a sufficient condition for the Finite Volume scheme to be conservative but also a necessary one. This shall be shown in a forthcoming work that constructs a unique half-flux attached to each half-face impinging at node p. The construction of this unique half-flux per half-edge is performed employing the methodology introduced in Abgrall (2018) to demonstrate that the residual distribution schemes possess a flux formulation and thus are locally conservative.*

3.2.7 Entropy stability

We aim at exhibiting conditions which ensure that the total entropy over the whole space $\mathbb{R}^2$ is non increasing that is

$$\sum_c |\omega_c| \left(\Sigma_c^{n+1} - \Sigma_c^n\right) \leq 0. \tag{3.2.23}$$

Knowing that the local in-cell inequality (3.2.18) holds true, *i.e.*,

$$\Sigma_c^{n+1} - \Sigma_c^n + \frac{\Delta t}{|\omega_c|} \sum_{p\in\mathcal{P}(c)} \sum_{f\in\mathcal{SF}(pc)} l_{pcf}\overline{Q}_{pcf} \leq 0,$$

implies that the global entropy inequality (3.2.23) holds true provided that

$$\sum_c \sum_{p\in\mathcal{P}(c)} \sum_{f\in\mathcal{SF}(p)} l_{pcf}\overline{Q}_{pcf} \geq 0,$$

where $\overline{Q}_{pcf}$ denotes the subface entropy flux attached to the subface f of corner pc. Similarly to the study of the Finite Volume scheme conservation in section 3.2.6, we exchange the summation over the cells with the summation over the nodes in the foregoing inequality to arrive at

$$\sum_p \sum_{c\in\mathcal{C}(p)} \sum_{f\in\mathcal{SF}(p)} l_{pcf}\overline{Q}_{pcf} \geq 0,$$

where $\mathcal{C}(p)$ is the set of cells sharing the point p. Therefore, a sufficient condition to ensure that the Finite Volume scheme satisfies the global entropy inequality (3.2.23) writes

$$\sum_{c\in\mathcal{C}(p)} \sum_{f\in\mathcal{SF}(p)} l_{pcf}\overline{Q}_{pcf} \geq 0. \tag{3.2.24}$$

Once more, observing that the sum over the cells c sharing p of the entropy fluxes attached to the subfaces impinging at p is rigorously equal to the sum over the left-sided and the right-sided entropy fluxes attached to the subfaces impinging at p leads to reformulate

sufficient condition (3.2.24) into

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}\left(\overline{Q}^{l}_{pf}+\overline{Q}^{r}_{pf}\right)\geq 0, \tag{3.2.25}$$

where $\mathcal{SF}(p)$ is the set of subfaces impinging at point p. Developing the expression of the left-sided and the right-sided subface entropy fluxes into the sufficient condition (3.2.25) yields

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}\left[\overline{Q}^{l}_{pf}(\mathbf{U}_{lf},\mathbf{U}_{rf},\mathbf{n}_{pf},\mathbf{v}_p)+\overline{Q}^{r}_{pf}(\mathbf{U}_{rf},\mathbf{U}_{lf},-\mathbf{n}_{pf},\mathbf{v}_p)\right]\geq 0. \tag{3.2.26}$$

Substituting the expression of the subface entropy flux (3.2.17) in terms of the approximate Riemann solver into the nodal entropy condition (3.2.26) and by virtue of the symmetry assumption of the approximate Riemann solver (3.2.7), *i.e.*, $\mathbf{W}_{pf}(\mathbf{U}_{lf},\mathbf{U}_{rf},\xi,\mathbf{v}_p)=\mathbf{W}_{pf}(\mathbf{U}_{rf},\mathbf{U}_{lf},-\xi,\mathbf{v}_p)$ we arrive at

$$\begin{aligned}\sum_{f\in\mathcal{SF}(p)} l_{pf}\Bigg[&\int_{-\infty}^{0}\left(\Sigma(\mathbf{W}_{pf}(\xi,\mathbf{v}_p))-\Sigma_{lf}\right)\mathrm{d}\xi+\int_{0}^{\infty}\left(\Sigma(\mathbf{W}_{pf}(\xi,\mathbf{v}_p))-\Sigma_{rf}\right)\mathrm{d}\xi\\&+\left(\mathbf{Q}(\mathbf{U}_{rf})-\mathbf{Q}(\mathbf{U}_{lf})\right)\cdot\mathbf{n}_{pf}\Bigg]\leq 0.\end{aligned} \tag{3.2.27}$$

3.3 Fundamental relation between Lagrangian and Eulerian frameworks

This section provides a general and systematic framework for constructing simple approximate Riemann solvers for one-dimensional hyperbolic systems of conservation laws written under Eulerian form. The underlying methodology stems from the transformation relating the Lagrangian and the Eulerian representations of conservation laws which is further applied to the Lagrangian Riemann solvers to deduce their Eulerian counterparts. This manner of proceeding ensures the direct transfer of the properties (conservation, positivity and entropy control) satisfied by the Lagrangian solver to its Eulerian counterpart. This transformation has been initially introduced in Gallice (2000, 2003) for designing conservative, positive and entropic simple approximate Riemann solvers, and, recently reused for the gas dynamics equations in Chan *et al.* (2021). It is worth noticing that this approach has been extended not only to the magnetohydrodynamics equations written under Powell's form but also to hyperbolic systems with source terms leading to well-balanced numerical discretizations, refer to Gallice (2002*a*).

3.3.1 One-dimensional Lagrange-to-Euler mapping

This section studies the one-dimensional Riemann problem located at the subface interface which is required to construct the numerical flux approximation by means of the generic expression (3.2.8). Let us consider the subface characterized by the unit outward $\mathbf{n}$, the corresponding Riemann problem reads

$$(\mathcal{RP}_E)\begin{cases} \dfrac{\partial \mathbf{U}}{\partial t} + \dfrac{\partial \mathbf{F}_{\mathbf{n}}(\mathbf{U})}{\partial x_{\mathbf{n}}} & = \mathbf{0}, \quad \text{where } \mathbf{F}_{\mathbf{n}}(\mathbf{U}) = \mathbb{F}(\mathbf{U})\mathbf{n}, \\ \mathbf{U}(x_{\mathbf{n}}, 0) & = \begin{cases} \mathbf{U}_l & \text{if} \quad x_{\mathbf{n}} < 0, \\ \mathbf{U}_r & \text{if} \quad x_{\mathbf{n}} \geq 0. \end{cases} \end{cases}$$

Here, $x_{\mathbf{n}} = \mathbf{x} \cdot \mathbf{n}$ denotes the space variable in the direction normal to the interface. This Riemann problem is also equipped with the the entropy inequality

$$\frac{\partial \Sigma}{\partial t} + \frac{\partial Q_{\mathbf{n}}}{\partial x_{\mathbf{n}}} \leq 0, \tag{3.3.1}$$

where $Q_{\mathbf{n}} = \mathbf{Q} \cdot \mathbf{n}$ denotes the entropy flux.

From now on, we focus on systems of conservation laws describing physical phenomena in the domain of continuum mechanics, for instance gas dynamics, shallow water equations, MagnetoHdydroDynamics (MHD), hyperelasticity... The interested reader might refer for instance to Kulikowskii *et al.* (2001) wherein several systems of conservation laws originated from continuum mechanics are described and studied. In this framework, we assume that the first component of the general system of conservation laws (3.1.1) is the mass conservation equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{3.3.2}$$

where $\rho > 0$ is the mass density and $\mathbf{v}$ the material velocity. Thus, the first component of the Riemann problem $(\mathcal{RP}_E)$ reads

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v_{\mathbf{n}}) = 0, \tag{3.3.3}$$

where $v_{\mathbf{n}} = \mathbf{v} \cdot \mathbf{n}$ is the projection of the material velocity onto the unit normal $\mathbf{n}$.

We construct the Lagrangian counterpart of $(\mathcal{RP}_{\mathcal{E}})$ introducing the Lagrange-to-Euler mapping $m \longmapsto x_{\mathbf{n}}(m, t)$ such that

$$\mathrm{d}x_{\mathbf{n}} = \frac{1}{\rho}\mathrm{d}m + v_{\mathbf{n}}\mathrm{d}t,$$

is an exact differential and m denotes the Lagrangian mass coordinate. By construction

$$\frac{\partial x_{\mathbf{n}}}{\partial m}(m, t) = \frac{1}{\rho}, \text{ and } \frac{\partial x_{\mathbf{n}}}{\partial t}(m, t) = v_{\mathbf{n}},$$

and since $\mathrm{d}x_{\mathbf{n}}$ is an exact differential the following compatibility condition holds true

$$\frac{\partial \tau}{\partial t} - \frac{\partial v_{\mathbf{n}}}{\partial m} = 0, \tag{3.3.4}$$

where $\tau = \frac{1}{\rho}$ is the specific volume. This is nothing but the Lagrangian mass/volume equation. Let us point out that we utilize the same notation for the Lagrangian and the Eulerian time. Moreover, the same physical quantity might be indifferently expressed either in terms of the Lagrangian coordinates (m,t) or in terms of the Eulerian ones $(x_{\mathbf{n}},t)$ knowing that $x_{\mathbf{n}} = x_{\mathbf{n}}(m,t)$. This amounts to write formally

$$\mathbf{U}(m,t) = \mathbf{U}(x_{\mathbf{n}}(m,t),t).$$

Taking the time derivative of the foregoing identity holding m fixed, *i.e.*, the Lagrangian time derivative, and applying the chain rule leads to

$$\frac{\partial \mathbf{U}}{\partial t}(m,t)\Big|_m = \frac{\partial \mathbf{U}}{\partial t}(m,t)\Big|_{x_{\mathbf{n}}} + v_{\mathbf{n}} \frac{\partial \mathbf{U}}{\partial x_{\mathbf{n}}}(x_{\mathbf{n}},t)\Big|_t.$$

Therefore, the following identity holds true

$$\rho \frac{\partial}{\partial t}(\tau \mathbf{U})(m,t)\Big|_m = \frac{\partial \mathbf{U}}{\partial t}(m,t)\Big|_{x_{\mathbf{n}}} + \frac{\partial (v_{\mathbf{n}}\mathbf{U})}{\partial x_{\mathbf{n}}}(x_{\mathbf{n}},t)\Big|_t.$$

Substituting the Eulerian time derivative thanks to $(\mathcal{RP}_E)$ in the foregoing equation leads to

$$\frac{\partial (\tau \mathbf{U})}{\partial t} + \frac{\partial}{\partial m}(\mathbf{F}_{\mathbf{n}} - v_{\mathbf{n}}\mathbf{U}) = \mathbf{0}. \tag{3.3.5}$$

This system is the Lagrangian counterpart of $(\mathcal{RP}_{\mathcal{E}})$. However, one notices that its first component is trivial, we thus replace it by (3.3.4). Finally, to complete the definition of the Lagrange-to-Euler mapping let us introduce the expressions of the Lagrangian variable and flux in terms of their Eulerian counterparts

$$\begin{aligned} \mathbf{V} &= \tau(\mathbf{U} - \rho \mathbf{e}_1) + \tau \mathbf{e}_1, \text{ where } \mathbf{e}_1 = (1,0,\ldots,0)^t, \\ \mathbf{G}_{\mathbf{n}} &= \mathbf{F}_{\mathbf{n}} - v_{\mathbf{n}}\mathbf{U} - v_{\mathbf{n}}\mathbf{e}_1. \end{aligned}$$

The foregoing formulas define the Lagrange-to-Euler transformation which allows us to deduce the Lagrangian Riemann problem

$$(\mathcal{RP}_{\mathcal{E}}) \left\{ \begin{array}{lcl} \dfrac{\partial \mathbf{V}}{\partial t} + \dfrac{\partial \mathbf{G}_{\mathbf{n}}(\mathbf{V})}{\partial m} & = & \mathbf{0} \\ \mathbf{V}(x_{\mathbf{n}},0) & = & \left\{ \begin{array}{lll} \mathbf{V}_l & \text{if} & m < 0, \\ \mathbf{V}_r & \text{if} & m \geq 0, \end{array} \right. \end{array} \right.$$

which is the Lagrangian counterpart of $(\mathcal{RP}_{\mathcal{E}})$. Here, $\mathbf{V} = \mathbf{V}(m,t)$ and $\mathbf{G}_{\mathbf{n}}$ are respectively the vector of conservative variables and the flux vector written under Lagrangian

representation. The foregoing Lagrangian Riemann problem is also equipped with the Lagrangian entropy inequality

$$\frac{\partial \sigma}{\partial t} + \frac{\partial q_{\mathbf{n}}}{\partial m} \leq 0, \tag{3.3.6}$$

where the Lagrangian entropy and entropy flux pair are written in terms of their Eulerian counterparts $\sigma = \tau\Sigma$ and $q_{\mathbf{n}} = Q_{\mathbf{n}} - v_{\mathbf{n}}\Sigma$.

Let $\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E)$, where $\xi_E = \frac{x_{\mathbf{n}}}{t}$, denotes the Eulerian Riemann solver which consists of an approximate solution to $(\mathcal{RP}_{\mathcal{E}})$. Similarly, we define $\mathbf{W}_L(\mathbf{V}_l, \mathbf{V}_r, \xi_L)$, where $\xi_L = \frac{m}{t}$, the Lagrangian Riemann solver which also consists of an approximate solution to $(\mathcal{RP}_L)$. We assume that the Eulerian approximate Riemann solver fulfills the basic requirements

- $\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E) = \mathbf{U}_l$ for $-\xi_E$ large enough,
- $\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E) = \mathbf{U}_r$ for ξ_E large enough,
- $\mathbf{W}_E(\mathbf{U}, \mathbf{U}, \xi_E) = \mathbf{U}$.

We prescribe similar assumptions for the Lagrangian Riemann solver.

3.3.2 Consistency of approximate Riemann solvers

Here, we recall the fundamental notion of consistency that has been initially introduced in the seminal works of Harten & Lax (1981); Harten *et al.* (1983). Integrating the conservation law $(\mathcal{RP})_E$ over $[-\Delta x_l, 0] \times [0, \Delta t]$ and replacing $\mathbf{U}(x_{\mathbf{n}}, t)$ by its approximation $\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E)$ leads to

$$\int_{-\Delta x_l}^{0} \Big(\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \frac{x_{\mathbf{n}}}{\Delta t}) - \mathbf{U}_l \Big) \, \mathrm{d}x_{\mathbf{n}} + \int_0^{\Delta t} \Big(\mathbf{F}_{\mathbf{n}}(\mathbf{U}(0^-, t)) - \mathbf{F}_{\mathbf{n}}(\mathbf{U}_l) \Big) \, \mathrm{d}t = \mathbf{0}.$$

This incites us to define the left-sided flux $\overline{\mathbf{F}}_{\mathbf{n}}^{-} = \dfrac{1}{\Delta t} \displaystyle\int_0^{\Delta t} \mathbf{F}_{\mathbf{n}}(\mathbf{U}(0^-, t))$ as follows

$$\overline{\mathbf{F}}_{\mathbf{n}}^{-} = \mathbf{F}_{\mathbf{n}}(\mathbf{U}_l) - \frac{1}{\Delta t} \int_{-\Delta x_l}^{0} \Big(\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \frac{x_{\mathbf{n}}}{\Delta t}) - \mathbf{U}_l \Big) \, \mathrm{d}x_{\mathbf{n}}.$$

Now, making the change of variable $\xi_E = \dfrac{x_{\mathbf{n}}}{\Delta t}$ in the foregoing integral and noticing that for $-\xi_E$ large enough $\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E) = \mathbf{U}_l$, we rewrite the left-sided flux employing the compact formula

$$\overline{\mathbf{F}}_{\mathbf{n}}^{-} = \mathbf{F}_{\mathbf{n}}(\mathbf{U}_l) - \int_{-\infty}^{0} (\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E) - \mathbf{U}_l) \, \mathrm{d}\xi_E. \tag{3.3.7}$$

Similarly, the right-sided flux writes

$$\overline{\mathbf{F}}_{\mathbf{n}}^{+} = \mathbf{F}_{\mathbf{n}}(\mathbf{U}_r) + \int_{0}^{+\infty} (\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E) - \mathbf{U}_r) \, \mathrm{d}\xi_E. \tag{3.3.8}$$

Subtracting (3.3.7) to (3.3.8) yields

$$\overline{\mathbf{F}}_{\mathbf{n}}^{+} - \overline{\mathbf{F}}_{\mathbf{n}}^{-} = \tag{3.3.9}$$
$$\int_{-\infty}^{0} (\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E) - \mathbf{U}_l)\, \mathrm{d}\xi_E + \int_{0}^{+\infty} (\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E) - \mathbf{U}_r)\, \mathrm{d}\xi_E + \mathbf{F}_{\mathbf{n}}(\mathbf{U}_r) - \mathbf{F}_{\mathbf{n}}(\mathbf{U}_l).$$

The approximate Riemann solver, $\mathbf{W}_E$, is consistent with the integral form of the Riemann problem, $(\mathcal{RP}_E)$, if and only if the left and right-sided fluxes are equal that is

$$\int_{-\infty}^{0} (\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E) - \mathbf{U}_l)\, \mathrm{d}\xi_E + \int_{0}^{+\infty} (\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E) - \mathbf{U}_r)\, \mathrm{d}\xi_E \tag{3.3.10}$$
$$+\mathbf{F}_{\mathbf{n}}(\mathbf{U}_r) - \mathbf{F}_{\mathbf{n}}(\mathbf{U}_l) = \mathbf{0}.$$

Then, the approximate Riemann solver induces a Finite Volume Godunov-type scheme which is conservative by construction.

Similarly to what has been done for the flux, we also introduce the left-sided and the right-sided entropy fluxes

$$\overline{Q}_{\mathbf{n}}^{-} = Q_{\mathbf{n}}(\mathbf{U}_l) - \int_{-\infty}^{0} [\Sigma(\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E)) - \Sigma(\mathbf{U}_l)]\, \mathrm{d}\xi_E, \tag{3.3.11a}$$
$$\overline{Q}_{\mathbf{n}}^{+} = Q_{\mathbf{n}}(\mathbf{U}_r) + \int_{0}^{+\infty} [\Sigma(\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E)) - \Sigma(\mathbf{U}_r)]\, \mathrm{d}\xi_E. \tag{3.3.11b}$$

Then, the approximate Riemann solver, $\mathbf{W}_E$, is consistent with the integral form of the entropy inequality (3.3.1) if and only if $\overline{Q}_{\mathbf{n}}^{+} - \overline{Q}_{\mathbf{n}}^{-} \leq 0$. Namely, substituting (3.3.11a) and (3.3.11b) into the foregoing difference yields the inequality

$$\int_{-\infty}^{0} [\Sigma(\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E)) - \Sigma(\mathbf{U}_l)]\, \mathrm{d}\xi_E + \int_{0}^{+\infty} [\Sigma(\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E)) - \Sigma(\mathbf{U}_r)]\, \mathrm{d}\xi_E \tag{3.3.12}$$
$$+ Q_{\mathbf{n}}(\mathbf{U}_r) - Q_{\mathbf{n}}(\mathbf{U}_l) \leq 0.$$

If this inequality holds true, the Riemann solver induces an entropic Finite Volume scheme. Obviously, similar definitions might be introduced for the Lagrangian Riemann solver $\mathbf{W}_L(\mathbf{V}_l, \mathbf{V}_r, \xi_L)$ and we omit it for the sake of conciseness.

3.3.3 Lagrangian simple Riemann solvers

3.3.3.a Definition of the simple Lagrangian Riemann solver

In this section, we focus on a particular class of approximate Riemann solvers initially introduced in Gallice (2002*a*, 2003) and named simple Riemann solvers. The Lagrangian Riemann solver $\mathbf{W}_L(\mathbf{V}_l, \mathbf{V}_r, \xi_L)$, where $\xi_L = \frac{m}{t}$, represents an approximate solution of the Lagrangian Riemann problem $(\mathcal{RP}_L)$. It is a simple Riemann solver if and only if it consists of $m+1$ constant states $\mathbf{V}_k$, $k = 1 \ldots m+1$ separated by m discontinuities of

slopes $\lambda_k,\ k=1,\dots,m$ in the (m,t) plane. More precisely,

$$\mathbf{W}_L(\mathbf{V}_l,\mathbf{V}_r,\frac{m}{t}) = \begin{cases} \mathbf{V}_1=\mathbf{V}_l \quad \text{if} \quad \frac{m}{t}<\lambda_1, \\ \mathbf{V}_k \quad \text{if} \quad \lambda_{k-1}\le\frac{m}{t}<\lambda_k,\ k=2,\dots,m, \\ \mathbf{V}_{m+1}=\mathbf{V}_r \quad \text{if} \quad \lambda_m\le\frac{m}{t}. \end{cases}$$

Here, the λ_k for $k=1,\dots,m$ are the Lagrangian wave speeds in ascending order in the (m,t) plane and thus homogeneous to $\frac{m}{t}$. As its name suggests, the simple solver represents the simplest form of approximate Riemann solver.

3.3.3.b Consistency of the simple Lagrangian solver

Similarly to what has been presented in section 3.3.2 the general expression of the left and right-sided Lagrangian fluxes write

$$\overline{\mathbf{G}}_{\mathbf{n}}^{-} = \mathbf{G}_{\mathbf{n}}(\mathbf{V}_l) - \int_{-\infty}^{0} (\mathbf{W}_L(\mathbf{V}_l,\mathbf{V}_r,\xi_L)-\mathbf{V}_l)\,\mathrm{d}\xi_L, \tag{3.3.13a}$$

$$\overline{\mathbf{G}}_{\mathbf{n}}^{+} = \mathbf{G}_{\mathbf{n}}(\mathbf{V}_r) + \int_{0}^{+\infty} (\mathbf{W}_L(\mathbf{V}_l,\mathbf{V}_r,\xi_L)-\mathbf{V}_r)\,\mathrm{d}\xi_L. \tag{3.3.13b}$$

Replacing $\mathbf{W}_L$ by its expression in the foregoing formulas leads to the explicit expressions of the left and right-sided fluxes in terms of the intermediate states and the wave speeds

$$\overline{\mathbf{G}}_{\mathbf{n}}^{-} = \mathbf{G}_{\mathbf{n}}(\mathbf{V}_l) - \sum_{k=1}^{m} \lambda_k^{(-)}(\mathbf{V}_{k+1}-\mathbf{V}_k), \tag{3.3.14a}$$

$$\overline{\mathbf{G}}_{\mathbf{n}}^{+} = \mathbf{G}_{\mathbf{n}}(\mathbf{V}_r) - \sum_{k=1}^{m} \lambda_k^{(+)}(\mathbf{V}_{k+1}-\mathbf{V}_k), \tag{3.3.14b}$$

where for any real, x, we denote by $x^{(+)}=\frac{1}{2}(|x|+x)$ and $x^{(-)}=\frac{1}{2}(|x|-x)$ respectively its positive and negative part. We claim that the simple Lagrangian Riemann solver $\mathbf{W}_L$ is consistent with the integral form of the conservation law $(\mathcal{RP}_L)$ if and only if $\overline{\mathbf{G}}_{\mathbf{n}}^{-}=\overline{\mathbf{G}}_{\mathbf{n}}^{+}$. Indeed subtracting (3.3.14a) to (3.3.14b) this amounts to write

$$-\sum_{k=1}^{m}\lambda_k(\mathbf{V}_{k+1}-\mathbf{V}_k)+\mathbf{G}_{\mathbf{n}}(\mathbf{V}_r)-\mathbf{G}_{\mathbf{n}}(\mathbf{V}_l)=\mathbf{0}. \tag{3.3.15}$$

Therefore, the numerical flux at the interface writes

$$\overline{\mathbf{G}}_{\mathbf{n}} = \frac{1}{2}\left[\mathbf{G}_{\mathbf{n}}(\mathbf{V}_l)+\mathbf{G}_{\mathbf{n}}(\mathbf{V}_r)\right] - \frac{1}{2}\sum_{k=1}^{m}|\lambda_k|(\mathbf{V}_{k+1}-\mathbf{V}_k).$$

As such, we have recovered the classical expression of the numerical flux decomposed into a centered part plus a viscous part, refer for instance to Toro (1999).

It remains to investigate the consistency of the simple Lagrangian Riemann solver

with the integral form of the Lagrangian entropy inequality (3.3.6). First, we compute the expression of the left and the right-sided Lagrangian entropy fluxes defined respectively by

$$\begin{aligned}\overline{q}_{\mathbf{n}}^{-} &= q_{\mathbf{n}}(\mathbf{V}_l) - \int_{-\infty}^{0} \left[\sigma(\mathbf{W}_L(\mathbf{V}_l, \mathbf{V}_r, \xi_L)) - \sigma(\mathbf{V}_l)\right] \mathrm{d}\xi_L, \\ \overline{q}_{\mathbf{n}}^{+} &= q_{\mathbf{n}}(\mathbf{V}_r) + \int_{0}^{+\infty} \left[\sigma(\mathbf{W}_L(\mathbf{V}_l, \mathbf{V}_r, \xi_L)) - \sigma(\mathbf{V}_r)\right] \mathrm{d}\xi_L.\end{aligned}$$

Replacing $\mathbf{W}_L$ by its expression in terms of the intermediate states and the waves speeds yields

$$\begin{aligned}\overline{q}_{\mathbf{n}}^{-} &= q_{\mathbf{n}}(\mathbf{V}_l) - \sum_{k=1}^{m} \lambda_k^{(-)}(\sigma_{k+1} - \sigma_k), \\ \overline{q}_{\mathbf{n}}^{+} &= q_{\mathbf{n}}(\mathbf{V}_r) - \sum_{k=1}^{m} \lambda_k^{(+)}(\sigma_{k+1} - \sigma_k),\end{aligned}$$

where $\sigma_k = \sigma(\mathbf{V}_k)$. We recall that the Riemann solver $\mathbf{W}_L$ is consistent with the entropy inequality (3.3.6) if and only if $\overline{q}_{\mathbf{n}}^{+} - \overline{q}_{\mathbf{n}}^{-} \leq 0$. Substituting the foregoing explicit expressions of the left and right-sided entropy fluxes in the aforementioned inequality, we thus claim that the simple Lagrangian Riemann solver $\mathbf{W}_L$ is consistent with the entropy inequality if and only if

$$-\sum_{k=1}^{m} \lambda_k(\sigma_{k+1} - \sigma_k) + q_{\mathbf{n}}(\mathbf{V}_r) - q_{\mathbf{n}}(\mathbf{V}_l) \leq 0. \tag{3.3.16}$$

Remark 3.3.1. *It is worth pointing out that very often the Lagrangian systems of conservation laws governing physical phenomena in the domain of continuum mechanics, e.g., gas dynamics, MHD, hyperelasticity, are characterized by a zero entropy flux, i.e.,* $q_{\mathbf{n}} = 0$. *This remarkable property which has been characterized in Després (2001) considerably simplifies the study of Lagrangian simple approximate Riemann solvers from the point of view of entropy stability.*

3.3.4 Construction of the simple Eulerian Riemann solver from its Lagrangian counterpart

Now, we construct the simple Eulerian Riemann solver $\mathbf{W}_E$ from the simple Lagrangian Riemann one $\mathbf{W}_L$ employing the Lagrange-to-Euler transformation introduced in section 3.3.1. This methodology has been initially introduced in Gallice (2002*a*, 2003) to deduce the Eulerian Riemann solver from its Lagrangian counterpart. Following Gallice (2002*a*) we suppose that the Riemann solver $\mathbf{W}_L$ satisfies the assumptions

- (H_1) $\lambda_k(\tau_{k+1} - \tau_k) + v_{\mathbf{n},k+1} - v_{\mathbf{n},k} = 0$, for $k = 1, \ldots, m$.
- (H_2) $\tau_k \geq 0$, for $k = 1, \ldots, m$.

Hypothesis (H_1) is nothing but the weak form of the volume/mass conservation equation (3.3.4) written across each discontinuity of speed λ_k for $k = 1, \dots, m$. By virtue of (H_1) for any $k = 1, \dots, m$

$$v_{\mathbf{n},k} + \lambda_k \tau_k = v_{\mathbf{n},k+1} + \lambda_k \tau_{k+1}.$$

This in turn allows to define the Eulerian wave speeds

$$\Lambda_k = v_{\mathbf{n},k} + \lambda_k \tau_k = v_{\mathbf{n},k+1} + \lambda_k \tau_{k+1}, \quad \text{for } k = 1 \dots m. \tag{3.3.17}$$

Now, observing that $\Lambda_{k+1} - \Lambda_k = v_{\mathbf{n},k+1} + \lambda_{k+1}\tau_{k+1} - v_{\mathbf{n},k+1} - \lambda_k \tau_{k+1} = \tau_{k+1}(\lambda_{k+1} - \lambda_k)$ and by virtue of (H_2), we deduce that the ordering of the Eulerian wave speeds is similar to that of the Lagrangian ones. Bearing this in mind, the Eulerian simple approximate Riemann solver $\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E)$, where $\xi_E = \frac{x_\mathbf{n}}{t}$, is deduced from its Lagrangian counterpart $\mathbf{W}_L(\mathbf{V}_l, \mathbf{V}_r, \xi_L)$, where $\xi_L = \frac{m}{t}$, employing the Euler-to-Lagrange transformation which maps the Lagrangian vector of variables $\mathbf{V}$ onto its Eulerian counterpart $\mathbf{U}(\mathbf{V}) = \rho(\mathbf{V} - \tau \mathbf{e}_1) + \rho \mathbf{e}_1$. Applying this transformation to the intermediate states of $\mathbf{W}_L$ yields

$$\mathbf{W}_E(\mathbf{U}_l, \mathbf{U}_r, \xi_E) = \begin{cases} \mathbf{U}_1 = \mathbf{U}_l = \mathbf{U}(\mathbf{V}_l) \quad \text{if} \quad \xi_E < \Lambda_1, \\ \mathbf{U}_k = \mathbf{U}(\mathbf{V}_k) \quad \text{if} \quad \Lambda_{k-1} \leq \xi_E < \Lambda_k \quad \text{for} \quad k = 2, \dots, m, \\ \mathbf{U}_{m+1} = \mathbf{U}_r = \mathbf{U}(\mathbf{V}_r) \quad \text{if} \quad \Lambda_m \leq \xi_E. \end{cases}$$

Here, the Eulerian wave speeds are deduced from the Lagrangian ones by means of (3.3.17). Thanks to (H_1), one can easily demonstrate that the weak form of the Eulerian mass conservation (3.3.3) holds true, that is

$$-\Lambda_k(\rho_{k+1} - \rho_k) + \rho_{k+1} v_{\mathbf{n},k+1} - \rho_k v_{\mathbf{n},k} = 0, \quad \text{for } k = 1 \cdots m. \tag{3.3.18}$$

This is nothing but the Eulerian version of (H_1).

3.3.5 Fundamental property relating the Eulerian and the Lagrangian fluxes

We finish this section by presenting the fundamental property of the simple Riemann solvers which allows to show that the difference between the left and the right-sided Eulerian fluxes is rigorously equal to the difference between the left and the right-sided Lagrangian ones provided that the underlying Eulerian and Lagrangian Riemann solvers are deduced one from the other by means of the Lagrange-to-Euler transformation. Substituting the expression of the simple Eulerian Riemann solver into (3.3.9) then the difference of the fluxes becomes

$$\overline{\mathbf{F}}_\mathbf{n}^+ - \overline{\mathbf{F}}_\mathbf{n}^- = -\sum_{k=1}^{m} \Lambda_k(\mathbf{U}_{k+1} - \mathbf{U}_k) + \mathbf{F}_\mathbf{n}(\mathbf{U}_r) - \mathbf{F}_\mathbf{n}(\mathbf{U}_l). \tag{3.3.19}$$

Now, substituting the expression of the Eulerian wave speeds (3.3.17) as functions of the Lagrangian ones and invoking the definition of the Eulerian intermediate states and fluxes allows to form their Lagrangian counterparts by means of the Lagrange-to-Euler transformation

$$\begin{aligned}\overline{\mathbf{F}}_{\mathbf{n}}^{+}-\overline{\mathbf{F}}_{\mathbf{n}}^{-} &= -\sum_{k=1}^{m}\lambda_k(\tau_{k+1}\mathbf{U}_{k+1}-\tau_k\mathbf{U}_k)+\mathbf{G}_{\mathbf{n},r}-\mathbf{G}_{\mathbf{n},l}+(v_{\mathbf{n},r}-v_{\mathbf{n},l})\mathbf{e}_1\\ &= -\sum_{k=1}^{m}\lambda_k(\mathbf{V}_{k+1}-\mathbf{V}_k)+\mathbf{G}_{\mathbf{n},r}-\mathbf{G}_{\mathbf{n},l}+\sum_{k=1}^{m}\Big[\underbrace{\lambda_k(\tau_{k+1}-\tau_k)+v_{\mathbf{n},k+1}-v_{\mathbf{n},k}}_{=0 \text{ thanks to } (\mathrm{H}_1)}\Big]\mathbf{e}_1\\ &= -\sum_{k=1}^{m}\lambda_k(\mathbf{V}_{k+1}-\mathbf{V}_k)+\mathbf{G}_{\mathbf{n},r}-\mathbf{G}_{\mathbf{n},l}.\end{aligned}$$

Finally, we arrive at the fundamental formula

$$-\sum_{k=1}^{m}\Lambda_k(\mathbf{U}_{k+1}-\mathbf{U}_k)+\mathbf{F}_{\mathbf{n}}(\mathbf{U}_r)-\mathbf{F}_{\mathbf{n}}(\mathbf{U}_l)=-\sum_{k=1}^{m}\lambda_k(\mathbf{V}_{k+1}-\mathbf{V}_k)+\mathbf{G}_{\mathbf{n}}(\mathbf{V}_r)-\mathbf{G}_{\mathbf{n}}(\mathbf{V}_l). \tag{3.3.20}$$

This amounts to write that

$$\overline{\mathbf{F}}_{\mathbf{n}}^{+}-\overline{\mathbf{F}}_{\mathbf{n}}^{-}=\overline{\mathbf{G}}_{\mathbf{n}}^{+}-\overline{\mathbf{G}}_{\mathbf{n}}^{-}. \tag{3.3.21}$$

This result is a consequence of the construction of the Eulerian simple solver from the Lagrangian one utilizing the Lagrange-to-Euler mapping and (H_1) assumption and is the fundamental formula that will be used in the following section together with the purely Lagrangian node-based conservation condition. We point out that formula (3.3.20) has been already introduced in Gallice (2002*a*, 2003) to demonstrate the equivalence of the Lagrangian and Eulerian Riemann solvers regarding the consistency properties provided that assumption (H_1) holds true. Following the same methodology, we compute the difference between the Eulerian left and right-sided entropy fluxes by subtracting (3.3.11a) to (3.3.11b) for the simple Eulerian Riemann solver $\mathbf{W}_E$

$$\overline{Q}_{\mathbf{n}}^{+}-\overline{Q}_{\mathbf{n}}^{-}=-\sum_{k=1}^{m}\Lambda_l(\Sigma_{k+1}-\Sigma_k)+Q_{\mathbf{n}}(\mathbf{U}_r)-Q_{\mathbf{n}}(\mathbf{U}_l).$$

Recalling that Eulerian wave speed satisfies $\Lambda_k=\lambda_k\tau_k+v_{\mathbf{n},k}=\lambda_k\tau_{k+1}+v_{\mathbf{n},k+1}$, and, the Eulerian entropy and entropy flux are expressed in terms of their Lagrangian counterparts by $\Sigma=\rho\sigma$ and $Q_{\mathbf{n}}=q_{\mathbf{n}}+\rho\sigma v_{\mathbf{n}}$ yields

$$-\sum_{k=1}^{m}\Lambda_k(\Sigma_{k+1}-\Sigma_k)+Q_{\mathbf{n}}(\mathbf{U}_r)-Q_{\mathbf{n}}(\mathbf{U}_l)=-\sum_{k=1}^{m}\lambda_k(\sigma_{k+1}-\sigma_k)+q_{\mathbf{n}}(\mathbf{V}_r)-q_{\mathbf{n}}(\mathbf{V}_l). \tag{3.3.22}$$

This amounts to write $\overline{Q}_{\mathbf{n}}^{+}-\overline{Q}_{\mathbf{n}}^{-}=\overline{q}_{\mathbf{n}}^{+}-\overline{q}_{\mathbf{n}}^{-}$.

3.4 Revisiting the node-based conservation condition and entropy condition in the case of approximate Riemann solvers

The conservation and entropy properties of the generic Finite Volume scheme (3.2.4) developed in section 3.2.5 rely respectively on the node-based conservation condition (3.2.22) and the node-based entropy condition (3.2.27). We have shown that these node-based conditions are sufficient to ensure that the aforementioned Finite Volume scheme is conservative and satisfies an entropy inequality under the time step condition (3.2.12). Let us recall that for a generic node p these node-based conditions are written in terms of the left and right-sided subface fluxes, $\overline{\mathbf{F}}^l_{pf}$ and $\overline{\mathbf{F}}^r_{pf}$ for the conservation condition and also in terms of the left and right-sided subface entropy fluxes $\overline{Q}^l_{pf}$ and $\overline{Q}^r_{pf}$ for the entropy condition, refer respectively to sections 3.2.6 and 3.2.7. The aforementioned left and right sided subface fluxes and subface entropy fluxes are attached to subface f impinging at p, and defined through the approximate Riemann solver, $\mathbf{W}_{pf}$, refer to (3.2.8) and (3.2.17).

3.4.1 Expression of the node-based conditions for a simple Riemann solver

Here, we shall further develop these node-based conservation and entropy conditions in the particular cases for which $\mathbf{W}_{pf}$ is a simple approximate Riemann solver. Consequently, $\mathbf{W}_{pf}$ consists of $m+1$ constant states $\mathbf{U}_k$ for $k = 1 \dots m+1$ separated by m waves of speeds Λ_k for $k = 1 \dots m$

$$\mathbf{W}_{pf}(\mathbf{U}_{lf}, \mathbf{U}_{rf}, \mathbf{n}_{pf}, \xi, \mathbf{v}_p) = \begin{cases} \mathbf{U}_1 = \mathbf{U}_{lf} & \text{if} \quad \xi < \Lambda_1, \\ \mathbf{U}_k & \text{if} \quad \Lambda_{k-1} < \xi \leq \Lambda_k, \ k = 2, \dots, m, \\ \mathbf{U}_{m+1} = \mathbf{U}_{rf} & \text{if} \quad \Lambda_m \leq \xi. \end{cases}$$

Here, the approximate Riemann solver is attached to the subface f whose unit normal $\mathbf{n}_{pf}$ points towards the cell characterized by the right state $\mathbf{U}_{rf}$. The self-similar variable reads $\xi = \frac{\mathbf{x}\cdot\mathbf{n}_{pf}}{t}$. In what follows, for the sake of simplicity, we shorten the notation of the Riemann solver into $\mathbf{W}_{pf}(\mathbf{U}_{lf}, \mathbf{U}_{rf}, \xi)$.

Before proceeding any further, let us introduce the following identity which shall be useful and holds true for any real valued function f

$$\int_{-\infty}^{0} \left[f(\mathbf{W}_{pf}(\mathbf{U}_{lf}, \mathbf{U}_{rf}, \xi)) - f(\mathbf{U}_{lf})\right] \mathrm{d}\xi + \int_{0}^{\infty} \left[f(\mathbf{W}_{pf}(\mathbf{U}_{lf}, \mathbf{U}_{rf}, \xi)) - f(\mathbf{U}_{rf})\right] \mathrm{d}\xi = \tag{3.4.1}$$
$$- \sum_{k=1}^{m} \Lambda_k \left[f(\mathbf{U}_{k+1}) - f(\mathbf{U}_k)\right].$$

Its straightforward application to the node-based conservation condition (3.2.22) turns it into

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}\left\{-\left[\sum_{k=1}^{m}\Lambda_k\left(\mathbf{U}_{k+1}-\mathbf{U}_k\right)\right]_{l,r}+\left[\mathbb{F}(\mathbf{U}_{rf})-\mathbb{F}(\mathbf{U}_{lf})\right]\mathbf{n}_{pf}\right\}=\mathbf{0}, \tag{3.4.2}$$

Similarly, by virtue of identity (3.4.1) the node-based entropy condition (3.2.27) becomes

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}\left\{-\left[\sum_{k=1}^{m}\Lambda_k\left(\Sigma(\mathbf{U}_{k+1})-\Sigma(\mathbf{U}_k)\right)\right]_{l,r}+\left[\mathbf{Q}(\mathbf{U}_{rf})-\mathbf{Q}(\mathbf{U}_{lf})\right]\cdot\mathbf{n}_{pf}\right\}\leq 0. \tag{3.4.3}$$

3.4.2 Lagrangian equivalence of the node-based conditions

So far we have written the expressions of the node-based conservation and entropy conditions obtained for a simple Eulerian Riemann solver. These expressions might be further simplified employing the fundamental property attached to simple Eulerian and Lagrangian Riemann solvers put in relation through the Lagrange-to-Euler mapping and the (H_1) hypothesis, refer to section 3.3.3. To this end, let us introduce, $\mathbf{W}_{pf}^{Lagr}(\mathbf{V}_{lf},\mathbf{V}_{rf},\xi_L)$, the Lagrangian counterpart of the Eulerian simple solver $\mathbf{W}_{pf}(\mathbf{U}_{lf},\mathbf{U}_{rf},\xi)$. The Lagrangian simple Riemann solver writes

$$\mathbf{W}_{pf}^{Lagr}(\mathbf{V}_{lf},\mathbf{V}_{rf},\xi_L)=\begin{cases}\mathbf{V}_1=\mathbf{V}_{lf} & \text{if} \quad \xi_L<\lambda_1,\\ \mathbf{V}_k \quad \text{if} \quad \Lambda_{k-1}\leq\xi_L<\lambda_k,\ k=2,\ldots,m,\\ \mathbf{V}_{m+1}=\mathbf{V}_{rf} \quad \text{if} \quad \lambda_m\leq\xi_L.\end{cases}$$

Here, the Lagrangian self-similar variable is defined by $\xi_L=\frac{m}{t}$ where m is the mass Lagrangian coordinate defined from the Eulerian coordinate $x_{\mathbf{n}_{pf}}$. The Lagrangian intermediate states $\mathbf{V}_k$ are connected to the Eulerian intermediate states through the Lagrange-to-Euler mapping and the Lagrangian wave speeds λ_k are related to their Eulerian counterpart thanks to (H_1) hypothesis. Now, with an obvious notation adaptation, the fundamental relations (3.3.20) for the flux and the entropy flux (3.3.22) turn into

$$\begin{aligned}-\sum_{k=1}^{m}\Lambda_k(\mathbf{U}_{k+1}-\mathbf{U}_k)+\mathbf{F}_{\mathbf{n},r}-\mathbf{F}_{\mathbf{n},l}&=-\sum_{k=1}^{m}\lambda_k(\mathbf{V}_{k+1}-\mathbf{V}_k)+\mathbf{G}_{\mathbf{n},r}-\mathbf{G}_{\mathbf{n},l},\\ -\sum_{k=1}^{m}\Lambda_k(\Sigma_{k+1}-\Sigma_k)+Q_{\mathbf{n},r}-Q_{\mathbf{n},l}&=-\sum_{k=1}^{m}\lambda_k(\sigma_{k+1}-\sigma_k)+q_{\mathbf{n},r}-q_{\mathbf{n},l}.\end{aligned}$$

Finally, utilizing these identities, the node-based Eulerian conservation and entropy conditions for the Finite Volume scheme boil down to a Lagrangian expression. More precisely, the Eulerian node-based conservation condition (3.4.2) is equivalent to the Lagrangian

one

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}\left\{-\left[\sum_{k=1}^{m}\lambda_k\left(\mathbf{V}_{k+1}-\mathbf{V}_k\right)\right]_{l,r}+\mathbf{G}_{\mathbf{n}_{pf},r}-\mathbf{G}_{\mathbf{n}_{pf},l}\right\}=\mathbf{0}. \tag{3.4.4}$$

Similarly the Eulerian node-based entropy condition (3.4.3) is equivalent to the Lagrangian one

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}\left\{-\left[\sum_{k=1}^{m}\lambda_k\left(\sigma(\mathbf{V}_{k+1})-\sigma(\mathbf{V}_k)\right)\right]_{l,r}+\left[\mathbf{q}(\mathbf{V}_r)-\mathbf{q}(\mathbf{V}_l)\right]\cdot\mathbf{n}_{pf}\right\}\leq 0. \tag{3.4.5}$$

The interest of the above formulations is their simplicity as they are expressed only in terms of Lagrangian fluxes and unknowns. Particularly, for fluid dynamics systems having a zero entropy flux Lagrangian formulation, refer to remark 3.3.1, the entropy condition boils down to the node-based condition

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}\left[\sum_{k=1}^{m}\lambda_k\left(\sigma(\mathbf{V}_{k+1})-\sigma(\mathbf{V}_k)\right)\right]_{l,r}\geq 0. \tag{3.4.6}$$

These Lagrangian formulations of the node-based conservation and entropy conditions shall be investigated further in the next chapter presenting the application of this original framework to the gas dynamics system.

3.5 Chapter summary

In this chapter, a general multidimensional Godunov-type Finite Volume scheme for systems of conservation laws on unstructured grids is constructed. This scheme is developed in terms of subface flux, $\mathbf{F}^-$, and its approximation is accomplished calling upon simple approximate Riemann solver, entropy stability, conservation and preservation of domain of definition with an appropriate time step condition.

Following the seminal work of Gallice (2003) and similarly to the previous chapter, we built an approximate Riemann solver in the Lagrangian framework in the first place. This strategy is pertinent since the discretization based on subcells have already been developed in Lagrangian frameworks, such as the works of Maire (2009, 2011); Loubère *et al.* (2010). The consistency of the simple Lagrangian solver allows us to recover the classical expression of the Lagrangian numerical flux. The Lagrangian solver will then serve its purpose as a building block to build the Eulerian counterpart. The key ingredients are

1. the Lagrange-to-Euler mapping, that allows to express the Lagrangian variable and flux vectors in terms of the Eulerian ones, and vice versa;

2. hypothesis (H_1) and (H_2), that allows to relate the discontinuity velocities in both the Lagrangian and Eulerian framework;

3. the fundamental relation that demonstrates the equivalence of the Lagrangian and Eulerian numerical fluxes, such that $\mathbf{F}^+-\mathbf{F}^-=\mathbf{G}^+-\mathbf{G}^-$

The conservation condition of this subface-based scheme differs from classical scheme since the fluxes on a face is not unique. Therefore, a node-based sufficient condition is imposed on each node to ensure conservation of the scheme. This condition requires that the sum of all the subface flux around a node to sum to zero, *i.e.* $(\mathbf{F}^+ - \mathbf{F}^- = 0)$, in order to retrieve the conservativity and entropy condition of the scheme. This node-based condition provides a vectorial relation that allows to determine the nodal parameter $\mathbf{v}_p$ that is an unknown in the Riemann solver and the numerical fluxes. The node-based conservation condition also implies that the stencil onto which the scheme operates covers all cells in contact with the current one leading to a genuinely multidimensional scheme, hence the name **multi-point** scheme. Classically, the notion of conservation is obtained via a **two-point** flux, that is based upon a fundamental one-dimensional nature.

Now that the theoretical framework of this novel method is presented, in the following chapter, we will implement this scheme on the gas dynamics equations for a concrete demonstration.

4 Multidimensional Godunov-type Finite Volume scheme: Applications

Contents

This chapter will be a continuity of the previous theoretical one. For a thorough demonstration, the subface-based Finite Volume method presented in chapter 3 is applied to the gas dynamics equation, firstly in two-dimensional, followed by the three-dimensional case.

The node-based conservation condition that takes into account the states and geometry of the cells surrounding a node then boils down to the Lagrangian nodal solver in Maire (2009) that allows to compute the nodal velocity. An additional term that involves this nodal velocity appears in the expression of the numerical flux. The application of this formalism to the case of gas dynamics provides a multidimensional Finite Volume scheme which is positive and entropic under an explicit condition on the time step. An associated Finite Volume simulation code has been built in multi-dimensions for unstructured meshes. Parallelization has been accomplished using the MPI library embedded in PETSc. A large set of 2D/3D numerical experiments show that the proposed solver is less sensitive to spurious instabilities such as the infamous carbuncle, compared to the classical one. To further improve accuracy, the current scheme has been extended to second-order in time and space. Lastly, an implicit time-stepping scheme is also implemented mainly to study steady flows.

Therefore, the aim of this chapter is to:

- Incorporate this original subface-based Finite Volume scheme to the gas dynamics equations in two- and three-dimensional;
- Develop an implicit time integration scheme for steady flows;
- Validate the numerical scheme on demanding test cases.

4.1 Application to two-dimensional gas dynamics equations

In this section, we aim at describing the main building blocks of the original Finite Volume scheme mentioned in Chapter 3 for discretizing the gas dynamics system onto unstructured general grids.

4.1.1 Governing equations

The gas dynamics system of conservation laws expresses the conservation of mass, momentum and total energy and is written under Eulerian representation as follows

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbb{F}(\mathbf{U}) = \mathbf{0}.$$

Here, $\mathbf{U} = \mathbf{U}(\mathbf{x}, t)$, where $\mathbf{x} \in \mathbb{R}^{\mathrm{d}}$, is the vector of conservative variables. This vector writes $\mathbf{U} = (\rho, \rho\mathbf{v}, \rho e)^t \in \mathbb{R}^{\mathrm{d}+2}$ where ρ is the mass density, $\mathbf{v}$ the velocity vector and e the specific total energy. The physical flux is described by the $(\mathrm{d}+2)\times^{\mathrm{d}}$ tensor

$$\mathbb{F}(\mathbf{U}) = \begin{pmatrix} \rho\mathbf{v}^t \\ \rho\mathbf{v}\otimes\mathbf{v} + p\mathbb{I}_d \\ \rho e\mathbf{v}^t + p\mathbf{v}^t \end{pmatrix},$$

where p denotes the thermodynamic pressure. The specific internal energy is given by $\varepsilon = e - \frac{1}{2}\mathbf{v}^2$. Let η be the specific physical entropy, and $\tau = \frac{1}{\rho}$ the specific volume, we make the fundamental assumption that $(\tau, \eta) \mapsto \varepsilon(\tau, \eta)$ is strictly convex which is equivalent to assume that $(\tau, \varepsilon) \mapsto \eta(\tau, \varepsilon)$ is strictly concave, refer to Godlewski & Raviart (1996). We work with the particular entropy, entropy flux pair $(\Sigma, \mathbf{Q}) = (-\rho\eta, -\rho\mathbf{v}\eta)$ and thus the gas dynamics system of conservation laws is equipped with the entropy inequality

$$\frac{\partial \rho\eta}{\partial t} + \nabla \cdot (\rho\eta\mathbf{v}) \geq 0. \tag{4.1.1}$$

The thermodynamic closure of this system of conservation laws is ensured by means of the complete equation of state

$$p(\tau, \eta) = -\frac{\partial \varepsilon}{\partial \tau}, \quad \theta(\tau, \eta) = \frac{\partial \varepsilon}{\partial \eta}. \tag{4.1.2}$$

We make the classical assumption that the absolute temperature is strictly positive: $\theta > 0$. By virtue of (4.1.2), writing the differential of $\varepsilon(\tau, \eta)$ leads to the fundamental Gibbs relation

$$\theta \, \mathrm{d}\eta = p \, \mathrm{d}\tau + \mathrm{d}\varepsilon. \tag{4.1.3}$$

The convexity of the specific internal energy with respect to the specific volume allows us to define the isentropic sound speed

$$\frac{a^2}{\tau^2} = -\frac{\partial p}{\partial \tau} = \frac{\partial^2 \varepsilon}{\partial \tau^2}. \tag{4.1.4}$$

4.1.2 Description of the one-dimensional Eulerian and Lagrangian systems of conservation laws

Let $\mathbf{n}$ be the unit normal to a generic interface along which we shall define Eulerian and Lagrangian one-dimensional problems in the normal direction and their associated Riemann solvers and finally construct the numerical approximation of the subfluxes of our Finite Volume scheme. In what follows, we shall restrict our developments to the bidimensional space, *i.e.*, $d = 2$. In this framework, $\mathbf{t}$ is the unit vector such that $(\mathbf{t}, \mathbf{n})$ is a direct orthonormal basis attached to the generic interface. The normal and tangential components of the velocity write respectively $v_\mathbf{n} = \mathbf{v} \cdot \mathbf{n}$ and $v_\mathbf{t} = \mathbf{v} \cdot \mathbf{t}$, and, obviously, $\mathbf{v} = v_\mathbf{n}\mathbf{n} + v_\mathbf{t}\mathbf{t}$. The vector of conservative variables and the flux projected onto the normal direction $\mathbf{n}$ write

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho v_\mathbf{n} \\ \rho v_\mathbf{t} \\ \rho e \end{pmatrix}, \quad \mathbf{F}_\mathbf{n} = \mathbb{F}\mathbf{n} = \begin{pmatrix} \rho v_\mathbf{n} \\ \rho v_\mathbf{n}^2 + p \\ \rho v_\mathbf{n} v_\mathbf{t} \\ \rho v_\mathbf{n} e + p v_\mathbf{n} \end{pmatrix}.$$

Therefore, the one-dimensional Eulerian system associated to the gas dynamics system in the $\mathbf{n}$ direction, where $x_\mathbf{n} = \mathbf{x} \cdot \mathbf{n}$, reads

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_\mathbf{n}(\mathbf{U})}{\partial x_\mathbf{n}} = \mathbf{0}.$$

This system is hyperbolic and admits the four following eigenvalues $v_\mathbf{n} - a$, $v_\mathbf{n}$ with multiplicity 2 and $v_\mathbf{n} + a$, and is equipped with the entropy inequality

$$\frac{\partial \rho\eta}{\partial t} + \frac{\partial}{\partial x_\mathbf{n}}(\rho\eta v_\mathbf{n}) \geq 0. \tag{4.1.5}$$

Employing the Lagrange-to-Euler mapping introduced in section 3.3.1 we can derive the corresponding one-dimensional Lagrangian system

$$\frac{\partial \mathbf{V}}{\partial t} + \frac{\partial \mathbf{G}_\mathbf{n}(\mathbf{V})}{\partial m} = \mathbf{0}, \tag{4.1.6}$$

where m is the Lagrangian mass coordinate related to the Eulerian coordinate $x_\mathbf{n}$. The

Lagrangian vector of conservative variables and the Lagrangian flux write

$$\mathbf{V} = \begin{pmatrix} \tau \\ v_{\mathbf{n}} \\ v_{\mathbf{t}} \\ e \end{pmatrix}, \quad \mathbf{G}_{\mathbf{n}} = \begin{pmatrix} -v_{\mathbf{n}} \\ p \\ 0 \\ pv_{\mathbf{n}} \end{pmatrix}.$$

The Lagrangian system is also hyperbolic and admits the four eigenvalues $-\frac{a}{\tau}$, 0 with multiplicity 2 and $\frac{a}{\tau}$. The selection of physically admissible weak solutions is ensured supplementing this system of conservation laws by the entropy inequality

$$\frac{\partial \eta}{\partial t} \geq 0. \tag{4.1.7}$$

4.1.3 Lagrangian simple Riemann solver

We consider the Lagrangian simple approximate Riemann solver corresponding to the foregoing one-dimensional Lagrangian gas dynamics system. This approximate Riemann solver structure mimics the continuous structure of the one-dimensional Lagrangian system of conservation laws. It is naturally composed of four states $\mathbf{V}_l$, $\mathbf{V}_l^\star$, $\mathbf{V}_r^\star$ and $\mathbf{V}_r$ separated respectively by discontinuities of speeds $-\lambda_l$, 0 and λ_r in the (m,t) plane. Here, λ_l and λ_r are positive real parameters which shall be constrained to ensure the positivity and entropy stability properties of the Riemann solver following the methodology introduced initially in Gallice (2002*a*, 2003) and revised recently in Chan *et al.* (2021). The states components write $\mathbf{V}_s = (\tau_s, v_{\mathbf{n},s}, v_{\mathbf{t},s}, e_s)^t$ and the intermediate states components $\mathbf{V}_s^\star = (\tau_s^\star, v_{\mathbf{n},s}^\star, v_{\mathbf{t},s}^\star, e_s^\star)^t$ for $s = l, r$. Assuming (H_1) hypothesis is satisfied yields

$$\begin{aligned} \lambda_l(\tau_l^\star - \tau_l) - (v_{\mathbf{n},l}^\star - v_{\mathbf{n},l}) &= 0, \\ 0(\tau_r^\star - \tau_l^\star) - (v_{\mathbf{n},r}^\star - v_{\mathbf{n},l}^\star) &= 0, \\ -\lambda_r(\tau_r - \tau_r^\star) - (v_{\mathbf{n},r} - v_{\mathbf{n},r}^\star) &= 0. \end{aligned}$$

This implies $v_{\mathbf{n},r}^\star = v_{\mathbf{n},l}^\star$ and we denote $v_{\mathbf{n}}^\star$ the common value of the velocity, *i.e.*, $v_{\mathbf{n}}^\star = v_{\mathbf{n},r}^\star = v_{\mathbf{n},l}^\star$. This shows that (H_1) ensures the continuity of kinematic velocity through the contact waves which is a rather satisfying physical behavior. Finally, the foregoing system boils down to

$$\lambda_l(\tau_l^\star - \tau_l) - (v_{\mathbf{n}}^\star - v_{\mathbf{n},l}) = 0, \tag{4.1.8a}$$

$$\lambda_r(\tau_r^\star - \tau_r) + v_{\mathbf{n}}^\star - v_{\mathbf{n},r} = 0. \tag{4.1.8b}$$

Now, following Chan *et al.* (2021) we complete the Lagrangian Riemann solver characterization introducing the intermediate fluxes for $s = l, r$

$$\overline{\mathbf{G}}_{\mathbf{n},s} = (-\overline{v}_{\mathbf{n},s}, \overline{p}_s, 0, \overline{(pv_{\mathbf{n}})}_s)^t.$$

These intermediate fluxes are nothing but the left and right-sided fluxes defined respectively by (3.3.14a) and (3.3.14b). Therefore $\overline{\mathbf{G}}_{\mathbf{n},l} \equiv \overline{\mathbf{G}}_{\mathbf{n}}^{-}$ and $\overline{\mathbf{G}}_{\mathbf{n},r} \equiv \overline{\mathbf{G}}_{\mathbf{n}}^{+}$ satisfy the system

$$\lambda_l(\mathbf{V}_l^\star - \mathbf{V}_l) + \overline{\mathbf{G}}_{\mathbf{n},l} - \mathbf{G}_{\mathbf{n},l} = \mathbf{0}, \tag{4.1.9a}$$
$$-\lambda_r(\mathbf{V}_r - \mathbf{V}_r^\star) + \mathbf{G}_{\mathbf{n},r} - \overline{\mathbf{G}}_{\mathbf{n},r} = \mathbf{0}, \tag{4.1.9b}$$

where $\mathbf{G}_{\mathbf{n},s} = \mathbf{G}_{\mathbf{n}}(\mathbf{V}_s)$ for $s = l, r$. Combining the first components of (4.1.9a), (4.1.9b) with (4.1.8a), (4.1.8b) leads to $\overline{v}_{\mathbf{n},l} = v_{\mathbf{n}}^\star = \overline{v}_{\mathbf{n},r}$. On the other hand, we make the structural assumption that the intermediate total energy flux writes under the form

$$\overline{(pv_{\mathbf{n}})}_s = \overline{p}_s v_{\mathbf{n}}^\star, \text{ for } s = l, r. \tag{4.1.10}$$

Gathering these results, we arrive at the following expressions of the intermediate states and fluxes

$$\mathbf{V}_s^\star = \begin{pmatrix} \tau_s^\star \\ v_{\mathbf{n}}^\star \\ v_{\mathbf{t},s}^\star \\ e_s^\star \end{pmatrix}, \text{ and } \overline{\mathbf{G}}_{\mathbf{n},s} = \begin{pmatrix} -v_{\mathbf{n}}^\star \\ \overline{p}_s \\ 0 \\ \overline{p}_s v_{\mathbf{n}}^\star \end{pmatrix}, \text{ for } s = l, r.$$

(H_1) hypothesis and the structural assumption (4.1.10) for the total energy flux allow us to reduce the number of scalar unknowns to 9, which must satisfy 8 scalar equations corresponding to the vectorial equations (4.1.9a) and (4.1.9b), that is

$$(\mathcal{S}_l) \begin{cases} \lambda_l(\tau_l^\star - \tau_l) - (v_{\mathbf{n}}^\star - v_{\mathbf{n},l}) = 0, \\ \lambda_l(v_{\mathbf{n}}^\star - v_{\mathbf{n},l}) + \overline{p}_l - p_l = 0, \\ \lambda_l(v_{\mathbf{t},l}^\star - v_{\mathbf{t},l}) = 0, \\ \lambda_l(e_l^\star - e_l) + \overline{p}_l v_{\mathbf{n}}^\star - p_l v_{\mathbf{n},l} = 0, \end{cases} \quad (\mathcal{S}_r) \begin{cases} \lambda_r(\tau_r^\star - \tau_r) + v_{\mathbf{n}}^\star - v_{\mathbf{n},r} = 0, \\ \lambda_r(v_{\mathbf{n}}^\star - v_{\mathbf{n},r}) - (\overline{p}_r - p_r) = 0, \\ \lambda_r(v_{\mathbf{t},r}^\star - v_{\mathbf{t},r}) = 0, \\ \lambda_r(e_r^\star - e_r) - (\overline{p}_r v_{\mathbf{n}}^\star - p_r v_{\mathbf{n},r}) = 0. \end{cases}$$

We observe that the tangential velocity is conserved through the left and right discontinuities, *i.e.*, $v_{\mathbf{t},s}^\star = v_{\mathbf{t},s}$ for $s = l, r$ since $\lambda_s > 0$. It is worth noticing that, the intermediate normal velocity $v_{\mathbf{n}}^\star$ might be viewed as a parameter in terms of which the 8 remaining unknowns might be expressed.

4.1.3.a Positivity preserving and entropy control of the Lagrangian Riemann solver

We briefly describe how to ensure that the foregoing simple Lagrangian Riemann solver ensures not only the positivity of the intermediate specific volume, *i.e.*, $\tau_s^\star \geq 0$, and internal energy, *i.e.*, $\varepsilon_s^\star \geq 0$, but also the intermediate entropy control, *i.e.*, $\eta_s^\star - \eta_s \geq 0$, for $s = l, r$. For a more detailed presentation of this topic the interested reader might refer to Chan *et al.* (2021). First, eliminating the normal velocity increment $v_{\mathbf{n}}^\star - v_{\mathbf{n},s}$ between the mass/volume equation and the momentum equation of $(\mathcal{S}_s)$ yields for $s = l, r$

$$\overline{p}_s - p_s = -\lambda_s^2(\tau_s^\star - \tau_s). \tag{4.1.11}$$

Further, dot-multiplying the momentum equation of $(\mathcal{S}_s)$ by $\frac{1}{2}(v_{\mathbf{n}}^\star + v_{\mathbf{n},s})$, and subtracting it to the total energy equation provides us the internal energy equation for $s = l, r$

$$\varepsilon_s^\star - \varepsilon_s + \frac{p_s + \overline{p}_s}{2}(\tau_s^\star - \tau_s) = 0, \tag{4.1.12}$$

since $\varepsilon_s^\star = e_s^\star - \frac{1}{2}(v_{\mathbf{n},s}^\star)^2 - \frac{1}{2}(v_{\mathbf{t},s}^\star)^2$ for $s = l, r$ and $v_{\mathbf{t},s}^\star = v_{\mathbf{t},s}$. Finally, substituting (4.1.11) into (4.1.12) leads to the expression of the post-discontinuity internal energy for $s = l, r$

$$\varepsilon_s^\star = \varepsilon_s - p_s(\tau_s^\star - \tau_s) + \frac{\lambda_s}{2}(\tau_s^\star - \tau_s)^2. \tag{4.1.13}$$

This equation is fundamental since it allows a straightforward derivation of a positivity condition for those specific internal energies. It also facilitates the study of the entropy production related to the simple approximate Riemann solver. Equation (4.1.13) shows that the specific internal energy $\varepsilon_s^\star$ is a convex quadratic function with respect to $\tau_s^\star - \tau_s$, and thus it is always greater than its minimum value

$$\varepsilon_s^\star \geq \varepsilon_s - \frac{p_s^2}{2\lambda_s^2}.$$

Therefore, the specific internal energy $\varepsilon_s^\star$ is positive provided that the wave speed λ_s satisfies the condition

$$\lambda_s \geq \frac{p_s}{\sqrt{2\varepsilon_s}}.$$

This condition has been already proposed in Vilar *et al.* (2016). Moreover, noticing that for a convex equation of state, *i.e.*, $\tau \mapsto p(\tau, \eta)$ strictly convex, there holds $\frac{a^2}{\tau^2} \geq \frac{p^2}{2\varepsilon}$, see Menikoff & B.J.Plohr (1989), the foregoing positivity condition turns into

$$\lambda_s \geq \frac{a_s}{\tau_s}, \quad \text{for } s = l, r. \tag{4.1.14}$$

The positivity conditions of the specific volumes are readily obtained written under the form

$$\lambda_l \geq -\frac{v_{\mathbf{n}}^\star - v_{\mathbf{n},l}}{\tau_l}, \quad \text{and} \quad \lambda_r \geq \frac{v_{\mathbf{n}}^\star - v_{\mathbf{n},r}}{\tau_r}.$$

It appears that these conditions are parameterized by the normal velocity $v_{\mathbf{n}}^\star$. We are then able to gather the positivity conditions of specific internal energy and specific volume into the global conditions

$$\lambda_l \geq \max\left(\frac{a_l}{\tau_l}, -\frac{v_{\mathbf{n}}^\star - v_{\mathbf{n},l}}{\tau_l}\right), \quad \text{and} \quad \lambda_r \geq \max\left(\frac{a_r}{\tau_r}, \frac{v_{\mathbf{n}}^\star - v_{\mathbf{n},r}}{\tau_r}\right). \tag{4.1.15}$$

Finally, we address the entropy control of the Lagrangian approximate Riemann solver. Decomposing the specific internal energy variation $\varepsilon_s^\star - \varepsilon_s$ across the discontinuities into an isentropic process followed by an isochoric one, we are able to express the entropy production in terms of the specific volume variation $\tau_s^\star - \tau_s$. The study of this entropy production term from Chan *et al.* (2021) shows that specific entropy increases across the discontinuity, *i.e.*, $\eta_s^\star - \eta_s \geq 0$ provided that the wave speed satisfies the condition

$$\lambda_s^2 \geq \frac{a^2(\overline{\tau}_s, \eta_s)}{\overline{\tau}_s^2} \quad \text{for all} \quad \overline{\tau}_s \in (\tau_s, \tau_s^\star), \quad \text{for } s = l, r. \tag{4.1.16}$$

We note that this condition has been also derived in the framework of relaxation scheme utilizing a relatively cumbersome approach, refer to Bouchut (2004).

4.1.3.b Consistency of the Lagrangian Riemann solver with its underlying conservation law

Let us investigate the consistency of the simple Lagrangian Riemann solver with the one-dimensional conservation law (4.1.6), refer to section 3.3.3.b. To this end, we sum relations (4.1.9a) and (4.1.9b) characterizing the intermediate fluxes $\overline{\mathbf{G}}_{\mathbf{n},l}$, $\overline{\mathbf{G}}_{\mathbf{n},r}$ and we get

$$\overline{\mathbf{G}}_{\mathbf{n},r} - \overline{\mathbf{G}}_{\mathbf{n},l} = \lambda_l(\mathbf{V}_l^\star - \mathbf{V}_l) - \lambda_r(\mathbf{V}_r - \mathbf{V}_r^\star) + \mathbf{G}_{\mathbf{n},r} - \mathbf{G}_{\mathbf{n},l}. \tag{4.1.17}$$

On the other hand, utilizing the expression of the components of $\overline{\mathbf{G}}_{\mathbf{n},r}$ and $\overline{\mathbf{G}}_{\mathbf{n},l}$ we arrive at

$$\overline{\mathbf{G}}_{\mathbf{n},r} - \overline{\mathbf{G}}_{\mathbf{n},l} = (\overline{p}_r - \overline{p}_l) \begin{pmatrix} 0 \\ 1 \\ 0 \\ v_{\mathbf{n}}^\star \end{pmatrix}. \tag{4.1.18}$$

Now, we address the consistency of the simple Lagrangian Riemann solver with its underlying conservation law by studying the solutions of $\overline{\mathbf{G}}_{\mathbf{n},r} - \overline{\mathbf{G}}_{\mathbf{n},l} = \mathbf{0}$. To this end, we compute the difference $\overline{p}_r - \overline{p}_l$ summing the second equations of $(\mathcal{S}_l)$ and $(\mathcal{S}_r)$

$$\overline{p}_r - \overline{p}_l = (\lambda_l + \lambda_r) \left\{ v_{\mathbf{n}}^\star - \left[\frac{\lambda_l v_{\mathbf{n},l} + \lambda_r v_{\mathbf{n},r}}{\lambda_l + \lambda_r} - \frac{(p_r - p_l)}{\lambda_r + \lambda_l} \right] \right\}. \tag{4.1.19}$$

This equation incites us to introduce the normal velocity

$$\overline{v}_{\mathbf{n}} = \frac{\lambda_l v_{\mathbf{n},l} + \lambda_r v_{\mathbf{n},r}}{\lambda_l + \lambda_r} - \frac{(p_r - p_l)}{\lambda_r + \lambda_l}, \tag{4.1.20}$$

which is nothing but the normal velocity of the classical Godunov acoustic solver, refer for instance to Toro (1999). Bearing this in mind, we claim that

- If $v_{\mathbf{n}}^\star = \overline{v}_{\mathbf{n}}$, then the simple Lagrangian Riemann solver is consistent with its underlying conservation law. Thus, the simple Lagrangian Riemann solver induces a classical conservative Godunov-type Finite Volume scheme.

- If $v_{\mathbf{n}}^{\star} \neq \overline{v}_{\mathbf{n}}$, then the simple Lagrangian Riemann solver is not consistent with its underlying conservation law and thus does not induce a conservative Godunov-type Finite Volume scheme.

In what follows, we investigate further the second case for which in general $v_{\mathbf{n}}^{\star} \neq \overline{v}_{\mathbf{n}}$ and we shall demonstrate how to retrieve a global conservation property for the Finite Volume scheme by means of the node-based conservation conditions introduced in section 3.4.

4.1.4 The nodal solver

4.1.4.a Expression of the node-based conservation condition

We have seen in section 3.4, that the global conservation of the Finite Volume scheme is ensured provided that the Lagrangian node-based condition (3.4.4) is satisfied. This condition reads

$$\sum_{f \in \mathcal{SF}(p)} l_{pf} \left\{ - \left[\sum_{k=1}^{m} \lambda_k \left(\mathbf{V}_{k+1} - \mathbf{V}_k \right) \right]_{l,r} + \mathbf{G}_{\mathbf{n}_{pf},r} - \mathbf{G}_{\mathbf{n}_{pf},l} \right\} = \mathbf{0}.$$

We are going to develop it in the case of gas dynamics using the properties of the simple Lagrangian Riemann solver constructed previously. First, we express the generic term between curly brackets at the left-hand side of the foregoing conservation condition with an obvious notation adaptation

$$- \left[\sum_{k=1}^{m} \lambda_k \left(\mathbf{V}_{k+1} - \mathbf{V}_k \right) \right]_{l,r} + \mathbf{G}_{\mathbf{n}_{pf},r} - \mathbf{G}_{\mathbf{n}_{pf},l} = \lambda_l (\mathbf{V}_l^{\star} - \mathbf{V}_l) - \lambda_r (\mathbf{V}_r - \mathbf{V}_r^{\star}) + \mathbf{G}_{\mathbf{n},r} - \mathbf{G}_{\mathbf{n},l}, \tag{4.1.21}$$

Here, the left and the right states correspond respectively to the cells c and d since the unit normal $\mathbf{n}_{pcf}$ is pointing from cell c towards cell d. The right-hand side of the foregoing equation coincides precisely with the right-hand side of (4.1.17) which has been derived for studying the consistency of the simple Lagrangian solver with its underlying conservation laws. Thus, by virtue of (4.1.17) and (4.1.18), equation (4.1.21) turns successively into

$$\begin{aligned} - \left[\sum_{k=1}^{m} \lambda_k \left(\mathbf{V}_{k+1} - \mathbf{V}_k \right) \right]_{l,r} + \mathbf{G}_{\mathbf{n}_{pf},r} - \mathbf{G}_{\mathbf{n}_{pf},l} =& \overline{\mathbf{G}}_{\mathbf{n},r} - \overline{\mathbf{G}}_{\mathbf{n},l} \\ =& (\overline{p}_r - \overline{p}_l) \begin{pmatrix} 0 \\ 1 \\ 0 \\ v_{\mathbf{n}}^{\star} \end{pmatrix} = (\overline{p}_{rf} - \overline{p}_{lf}) \begin{pmatrix} 0 \\ 1 \\ 0 \\ v_{\mathbf{n}_{pf}}^{\star} \end{pmatrix}. \end{aligned}$$

We observe that the last line has been written using once more an obvious notation adaptation. Namely, $\overline{p}_{rf}$ and $\overline{p}_{lf}$ denote the left and the right-sided interfacial pressures attached to subface f. Finally, we arrive at the conclusion that our Finite Volume scheme

is conservative provided that the following node-based condition is satisfied

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}(\overline{p}_{rf}-\overline{p}_{lf})\begin{pmatrix}0\\1\\0\\v^{\star}_{\mathbf{n}_{pf}}\end{pmatrix}=\mathbf{0}. \tag{4.1.22}$$

Realizing that the second and third components of the foregoing vector correspond to $\mathbf{n}_{pf}$, then condition (4.1.22) implies

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}(\overline{p}_{rf}-\overline{p}_{lf})\mathbf{n}_{pf}=\mathbf{0}. \tag{4.1.23}$$

We observe that $v^{\star}_{\mathbf{n}_{pf}}$ is still an unknown parameter attached to each subface impinging at node p. Thus, the number of unknown parameters at node p is much greater than the number of equations given by the conservation condition (4.1.22). Therefore, to close this system of equations, we assume that the parameter $v^{\star}_{\mathbf{n}_{pf}}$ is the projection of the unknown nodal vector $\mathbf{v}_p$ onto the unit normal $\mathbf{n}_{pf}$, that is

$$v^{\star}_{\mathbf{n}_{pf}}=\mathbf{v}_p\cdot\mathbf{n}_{pf}. \tag{4.1.24}$$

This fundamental assumption drastically reduces the number of unknowns to the vectorial unknown $\mathbf{v}_p$, which can be interpreted as an approximation of the nodal velocity. With this assumption the conservation condition (4.1.23) is equivalent to the conservation condition (4.1.22). Thanks to (4.1.19) and (4.1.20) we are able to express the interface pressures difference into the node-based conservation which becomes

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}(\lambda_{lf}+\lambda_{rf})(\mathbf{v}_p\cdot\mathbf{n}_{pf}-\overline{v}_{\mathbf{n}_{pf}})\mathbf{n}_{pf}=\mathbf{0},$$

where $\overline{v}_{\mathbf{n},pf}$ is obtained from (4.1.20) with obvious notation adaptation

$$\overline{v}_{\mathbf{n}_{pf}}=\frac{\lambda_{lf}v_{\mathbf{n}_{pf},l}+\lambda_{rf}v_{\mathbf{n}_{pf},r}}{\lambda_{lf}+\lambda_{rf}}-\frac{p_{rf}-p_{lf}}{\lambda_{lf}+\lambda_{rf}}.$$

Finally, the node-based conservation condition (4.1.23) boils down to the system

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}(\lambda_{lf}+\lambda_{rf})(\mathbf{n}_{pf}\otimes\mathbf{n}_{pf})\mathbf{v}_p=\sum_{f\in\mathcal{SF}(p)} l_{pf}(\lambda_{lf}+\lambda_{lf})\overline{v}_{\mathbf{n}_{pcf}}\mathbf{n}_{pf}. \tag{4.1.25}$$

This system always admits a unique solution which provides an approximation of the nodal velocity $\mathbf{v}_p$. It is thus called a nodal solver. We point out that the foregoing system has been already obtained when constructing a cell-centered Finite Volume discretization of multidimensional Lagrangian hydrodynamics, refer to Maire (2009). It has been also retrieved in Shen *et al.* (2014) for designing a Finite Volume scheme for Eulerian gas dynamics. It has thus the same properties, for instance the one-dimensional planar or

radial flows that are aligned with Cartesian or polar grids are preserved.

4.1.4.b Expression of the node-based entropy condition

Gathering the foregoing results, we are now in position to express what has become the nodal-based entropy condition (3.4.6). With the present simple Lagrangian Riemann solver the left-hand side of (3.4.6) writes

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}\left[\sum_{k=1}^{m}\lambda_k\left(\sigma(\mathbf{V}_{k+1})-\sigma(\mathbf{V}_k)\right)\right]_{l,r} = \sum_{f\in\mathcal{SF}(p)} l_{pf}\left[\lambda_l(\eta_l^\star-\eta_l)+\lambda_r(\eta_r^\star-\eta_r)\right]_{l,r}.$$

Here, we have use $\sigma=-\eta$, where η is the physical concave entropy. Consequently the node-based entropy condition for our Finite Volume scheme turns into

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}\left[\lambda_l(\eta_l^\star-\eta_l)+\lambda_r(\eta_r^\star-\eta_r)\right]_{l,r}\geq 0, \tag{4.1.26}$$

which is satisfied provided that the simple Lagrangian Riemann solver is entropy stable, refer to condition (4.1.16). This inequality might be satisfied relatively simply by utilizing a classically tuning of the wave speeds of the Riemann solver similarly to what has been undertaken in the one-dimensional framework, refer to Chan *et al.* (2021).

4.1.5 Corresponding Eulerian simple Riemann solver

Relying on hypothesis (H_1) we can deduce the Eulerian wave speeds Λ_l, Λ_0 and Λ_r from their Lagrangian counterparts setting

$$\begin{aligned}\Lambda_l &= v_{\mathbf{n},l}-\lambda_l\tau_l = v_{\mathbf{n}}^\star-\lambda_l\tau_l^\star,\\ \Lambda_0 &= v_{\mathbf{n}}^\star,\\ \Lambda_r &= v_{\mathbf{n}}^\star+\lambda_r\tau_r^\star = v_{\mathbf{n},r}+\lambda_r\tau_r.\end{aligned}$$

It is clear that the Eulerian wave speeds are ordered, *i.e.* $\Lambda_l\leq\Lambda_0\leq\Lambda_r$, provided that the Lagrangian approximate Riemann solver is positivity preserving, *i.e.*, $\tau_s^\star\geq 0$. This holds true granted that the Lagrangian wave speeds satisfy (4.1.15). Bearing this in mind, we are able to deduce the Eulerian approximate Riemann solver from its Lagrangian counterpart as follows

$$\mathbf{W}_E\left(\mathbf{U}_l,\mathbf{U}_r,\frac{x_{\mathbf{n}}}{t}\right)=\begin{cases}\mathbf{U}_l & \text{if} \quad \frac{x_{\mathbf{n}}}{t}\leq\Lambda_l,\\ \mathbf{U}_l^\star=\mathbf{U}(\mathbf{V}_l^\star) & \text{if} \quad \Lambda_l<\frac{x_{\mathbf{n}}}{t}\leq\Lambda_0,\\ \mathbf{U}_r^\star=\mathbf{U}(\mathbf{V}_r^\star) & \text{if} \quad \Lambda_0<\frac{x_{\mathbf{n}}}{t}\leq\Lambda_r,\\ \mathbf{U}_r & \text{if} \quad \Lambda_r<\frac{x_{\mathbf{n}}}{t}.\end{cases}$$

Here, $\mathbf{V}\mapsto\mathbf{U}(\mathbf{V})$ is the Lagrange-to-Euler mapping introduced in section 3.3 which allows us to define straightforwardly the intermediate states of the simple Eulerian solver

from its Lagrangian counterpart. Consequently, the Eulerian intermediate states read $\mathbf{U}_s^\star = (\rho_s^\star, \rho_s^\star v_{\mathbf{n}}^\star, \rho_s^\star v_{\mathbf{t},s}^\star, \rho_s^\star e_s^\star)^t$ knowing that $\rho_s^\star = (\tau_s^\star)^{-1}$ for $s = l, r$. We observe that the Eulerian approximate Riemann solver is also parameterized by the normal star-velocity. More importantly, by construction, the Eulerian approximate Riemann solver has the same properties as its Lagrangian counterpart, *i.e.*, it preserves the positivity of mass density, specific energy and ensures entropy increase under specific conditions on the Lagrangian wave speeds, refer to (4.1.16).

4.1.6 Summary and algorithm of the Eulerian multidimensional Finite Volume scheme

We recall that the multidimensional Finite Volume scheme writes

$$\mathbf{U}_c^{n+1} - \mathbf{U}_c^n + \frac{\Delta t}{|\omega_c|} \sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{SF}(pc)} l_{pcf} \overline{\mathbf{F}}_{pcf} = \mathbf{0},$$

where $\overline{\mathbf{F}}_{pcf}$ is the left-sided flux with respect to the subface f and the unit outward normal $\mathbf{n}_{pcf}$. The left-sided flux (3.3.7) for a simple Eulerian Riemann solver is obtained by

$$\overline{\mathbf{F}}_{pcf} \equiv \overline{\mathbf{F}}^-_{\mathbf{n}_{pcf}} = \mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_c) - \left[\sum_{k=1}^{m} \Lambda_k^{(-)} (\mathbf{U}_{k+1} - \mathbf{U}_k) \right]_{c,d}. \tag{4.1.27}$$

On the other hand, the right-sided flux writes

$$\overline{\mathbf{F}}^+_{\mathbf{n}_{pcf}} = \mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_d) - \left[\sum_{k=1}^{m} \Lambda_k^{(+)} (\mathbf{U}_{k+1} - \mathbf{U}_k) \right]_{c,d}. \tag{4.1.28}$$

Taking the arithmetic average of the left and the right-sides fluxes allows us to define the averaged flux on the subface f

$$\overline{\mathbf{F}}_{\mathbf{n}_{pcf}} = \frac{1}{2} \left[\mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_c) + \mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_d) \right] - \frac{1}{2} \left[\sum_{k=1}^{m} |\Lambda_k| (\mathbf{U}_{k+1} - \mathbf{U}_k) \right]_{c,d}. \tag{4.1.29}$$

This would be the numerical flux for a classical face-based Finite Volume method.

Now, recalling that the difference between the right and left-sided Eulerian flux coincides with the difference between the right and the left-sided Lagrangian fluxes leads to

$$\begin{aligned} \overline{\mathbf{F}}^+_{\mathbf{n}_{pcf}} - \overline{\mathbf{F}}^-_{\mathbf{n}_{pcf}} =& \overline{\mathbf{G}}^+_{\mathbf{n}_{pcf}} - \overline{\mathbf{G}}^-_{\mathbf{n}_{pcf}}, \quad \text{thanks to (3.3.21)} \\ =& (\overline{p}_{d(c,f)} - \overline{p}_{c(d,f)}) \begin{pmatrix} 0 \\ 1 \\ 0 \\ \mathbf{v}_p \cdot \mathbf{n}_{pcf} \end{pmatrix}. \end{aligned}$$

Finally, combining the foregoing equation with (4.1.29) we arrive at

$$\begin{aligned}\overline{\mathbf{F}}^{-}_{\mathbf{n}_{pcf}} =& \frac{1}{2}\left[\mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_c) + \mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_d)\right] - \frac{1}{2}\left[\sum_{k=1}^{m} |\Lambda_k|(\mathbf{U}_{k+1} - \mathbf{U}_k)\right]_{c,d} \\ & -\frac{1}{2}(\lambda_{c(d,f)} + \lambda_{d(c,f)})\left[\mathbf{v}_p \cdot \mathbf{n}_{pcf} - \overline{v}_{\mathbf{n}_{pcf}}\right]\begin{pmatrix} 0 \\ 1 \\ 0 \\ \mathbf{v}_p \cdot \mathbf{n}_{pcf} \end{pmatrix}.\end{aligned} \tag{4.1.30}$$

This explicit expression of the numerical flux shows that in the case $\mathbf{v}_p \cdot \mathbf{n}_{pcf} = \overline{v}_{\mathbf{n}_{pcf}}$ we retrieve the classical conservative face-based Finite Volume scheme. Moreover, the foregoing expression shall be useful to construct the time implicit discretization.

The algorithm starts from the data of $\mathbf{U}_c^{n=0}$ at initial time index $n = 0$ for all cell index c and from the data for the boundary conditions. We execute the following steps successively until $t_{\text{final}} = t^{n+1}$ is reached:

- **In.** Data $\mathbf{U}_c^n$ for all cell c, fixed mesh ω_c.

  1. **Nodal solver:** Compute nodal velocity v_p for all vertices.
     a. *Evaluate the initial nodal velocity* v_p.
     b. *Evaluate* Λ_l *and* Λ_r with (4.1.15)).
     b. *Evaluate the right-hand side and matrix then solve the system* (4.1.25).
     c. *Iterate in the fixed-point loop to obtain the nodal velocity* v_p. Note that the fixed-point iteration is due to the non-linearity in the nodal solver in order to ensure positivity of the specific volume.
  3. **Riemann problem:** Compute the subface fluxes $\overline{\mathbf{F}}_{pcf}$ for any subface.
     a. *Evaluate the intermediate wave speed* $\Lambda_0 = \mathbf{v}_p \cdot \mathbf{n}$.
     b. *Evaluate* Λ_l *and* Λ_r *with* (4.1.15) *such as* $\Lambda_l = v_{\mathbf{n},l} - \lambda_l\tau_l = v^\star_{\mathbf{n}} - \lambda_l\tau_l^\star$ *and* $\Lambda_r = v_{\mathbf{n},r} + \lambda_r\tau_r = v^\star_{\mathbf{n}} - \lambda_r\tau_r^\star$.
     c. *Evaluate the intermediate states.* using the Lagrangian Riemann solver in section 4.1.3.
     d. *Compute the subface fluxes* $\overline{\mathbf{F}}_{pcf}$ *with* (4.1.30).
  4. **Compute the final face flux**.
  5. **Evaluate** Δt**.**
  6. **Update** $\mathbf{U}_c^{n+1}$ using (3.2.4).

- **Out.** Data $\mathbf{U}_c^{n+1}$ for all cell c.

4.1.7 Extension to second-order of accuracy

The numerical schemes are extended to a nominal second-order of accuracy by means of monotonic piecewise-linear limited polynomial reconstruction and Runge-Kutta time-discretization.

Time discretization We employ the classical second-order explicit Runge-Kutta scheme (RK2) with two stages, see Gottlieb & Shu (1998). Let us rewrite equation (3.2.4) under the following operator form to introduce some notations:

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t\ \mathcal{F}(\mathbf{U}^n, t^n). \tag{4.1.31}$$

The RK2 method then rewrites as

$$\begin{aligned} \delta\mathbf{U}^n &= \Delta t \mathcal{F}(\mathbf{U}^n, t^n), \\ \delta\mathbf{U}^{n+1/2} &= \Delta t \mathcal{F}\left(\mathbf{U}^n + \frac{\delta\mathbf{U}^n}{2}, t^n + \frac{\Delta t}{2}\right), \\ \mathbf{U}^{n+1} &= \mathbf{U}^n + \delta\mathbf{U}^{n+1/2}. \end{aligned} \tag{4.1.32}$$

Space discretization We rely on the classical cell-centered piece-wise linear polynomial reconstruction technique, see Diot (2002) whose coefficients are computed with a least-square procedure and further associated to a limiting function (slope limiter). The limited reconstruction within each cell c is made for each primitive variable (ρ, u, v, P):

$$\widetilde{\mathbf{W}}_c(x) = \mathbf{W}_c + \phi_c^{\mathbf{W}} \nabla\mathbf{W}_c \cdot (\mathbf{x} - \mathbf{x}_c), \tag{4.1.33}$$

where $\mathbf{W}_c$ is the known cell average value, $\nabla\mathbf{W}_c = \left(\frac{\partial\mathbf{W}}{\partial x}, \frac{\partial\mathbf{W}}{\partial y}\right)^t$ is the estimated gradient in cell c, $\phi_c^{\mathbf{W}}$ the limiting function and $\mathbf{x}_c = (x_c, y_c)^t$ the cell centroid. To estimate the components of the gradient, we start by writing the second-order Taylor expansion of $\mathbf{w}$ in cell c :

$$\mathbf{W}(x - x_c, y - y_c) \simeq \mathbf{W}_c + \frac{\partial\mathbf{W}}{\partial x}(x - x_c) + \frac{\partial\mathbf{W}}{\partial y}(y - y_c). \tag{4.1.34}$$

We will choose the partial derivatives in the expansion to minimize the reconstruction error on smooth solutions in a least-square sense over a large enough set of neighbor cells, called the reconstruction stencil. The considered neighbors are the ones sharing at least one vertex with cell c, assume we order them with subscript $k = 1, \ldots, K$. The set of neighbor cells is denoted by $\mathcal{K}(c)$. The resulting least squares problem yields an over-determined system with a rectangular $K \times 2$ matrix $\mathcal{L}$:

$$\underbrace{\begin{bmatrix} \frac{1}{\beta_1}(x_1 - x_c) & \frac{1}{\beta_1}(y_1 - y_c) \\ \vdots & \vdots \\ \frac{1}{\beta_k}(x_k - x_c) & \frac{1}{\beta_k}(y_k - y_c) \\ \vdots & \vdots \\ \frac{1}{\beta_K}(x_K - x_c) & \frac{1}{\beta_K}(y_K - y_c) \end{bmatrix}}_{\mathcal{L}} \nabla\mathbf{W}_c = \underbrace{\begin{bmatrix} \mathbf{W}_1 - \mathbf{W}_c \\ \vdots \\ \mathbf{W}_k - \mathbf{W}_c \\ \vdots \\ \mathbf{W}_N - \mathbf{W}_c \end{bmatrix}}_{\delta\mathbf{W}_c}, \tag{4.1.35}$$

β_k is a weight based on the distance between neighbor cell k and c: $\beta_k = |x_k - x_c|^\theta$, with $\theta = 2$. To solve (4.1.35) we use a QR factorization method to obtain the pseudo-inverse of matrix $\mathcal{L}$, giving the gradient approximation.

Slope limiter The limiting function implemented is the one introduced by Venkatakrishnan (1995) which is a smooth alternative to the Barth & Jespersen (1989) . The goal is to prevent the formation of local extrema. The scheme consists of finding a value $0 \leq \phi_c^{\mathbf{W}} \leq 1$ in each control-volume that limits the gradient of the second-order reconstruction of the solution. The maximum allowable value of $\phi_{c,k}^w$ for each neighbor cells $k \in \mathcal{K}(c)$ and for one component w of $\mathbf{W}$ is expressed as

$$\phi_{c,k}^w = \frac{(\Delta_+^2 + \varepsilon^2) + 2\Delta_-\Delta_+}{\Delta_+^2 + 2\Delta_-^2 + \Delta_-\Delta_+ + \varepsilon^2}, \tag{4.1.36}$$

where $\varepsilon^2 = (K\ell_c)^3$ with the adjustable parameter $K = 2$ and the characteristics length for the control volume $l_c = 2\sqrt{|\omega_c|/\pi}$ and $\Delta_- = w_c - w_k$. For the case $\Delta_- > 0$, $\Delta_+ = w_c^{\max} - w_c$, with $w_c^{\max} = \max_{k\in\mathcal{K}(c)}(w_k, w_c)$, and for the case where $\Delta_- < 0$, $\Delta_+ = w_c^{\min} - w_c$, with $w_c^{\min} = \min_{k\in\mathcal{K}(c)}(w_k, w_c)$. Lastly, the limiter in each control volume yields $\phi_c^w = \min_{k\in\mathcal{K}(c)}(\phi_{c,k}^w)$.
In the following section, the schemes will be named as such:

- `Eul-0` multi-point scheme: 1st order subface-based scheme;
- `Eul-1` multi-point scheme: 2nd order subface-based scheme;
- `Eul-0` two-point scheme: 1st order classical face-based scheme;
- `Eul-1` two-point scheme: 2nd order classical face-based scheme;

4.1.8 Two-dimensional numerical validation

In this section, we run an extensive numerical test case campaign to demonstrate the performance of both the first-order explicit two-point and multi-point schemes. These test cases are ran on various types of grids that can be made of triangular, quadrangular and polygonal cells. The CFL number is set to 0.5 for all cases. For all the test cases the triangular meshes are obtained employing the open source mesh generator Gmsh, refer to Geuzaine & Remacle (2009), the structured quadrangular meshes are directly computed by the Finite Volume code and the polygonal grids result from of an in-house mesher based on Voronoi tessellation Loubère *et al.* (2010).

Modified sod shock tube We start with the modified Sod shock tube as a sanity check. This test case is similar to the one presented in the chapter for the one-dimensional scheme. The simulation is done on a domain of $\Omega = [0,1] \times [0,1]$ on $N = 100 \times 100$ quadrilateral cells and the interface located at $x = 0.3$. The initial set up reads

$$(\rho, u, v, p)_l = (1, 0, 75, 0, 1) \quad , \quad (\rho, u, v, p)_r = (0.125, 0, 0, 0.1) \quad , \quad t_{final} = 0.2.$$

A piston boundary condition is imposed on the left with $u^\star = u_l = 0.75$, $u^\star = 0$ is imposed on the right and outflow conditions on the top and bottom, and we run the simulation with the first-order numerical scheme. This test case once again ensures that the sonic glitch in the rarefaction wave is absent. Figure 4.1 is a comparison of both two-point (red cross) and multi-point schemes (blue circle) with the Roe solver with and without entropy fix (purple diamond and green square respectively). The results show that the sonic glitch is correctly captured by both two-point and multi-point scheme.

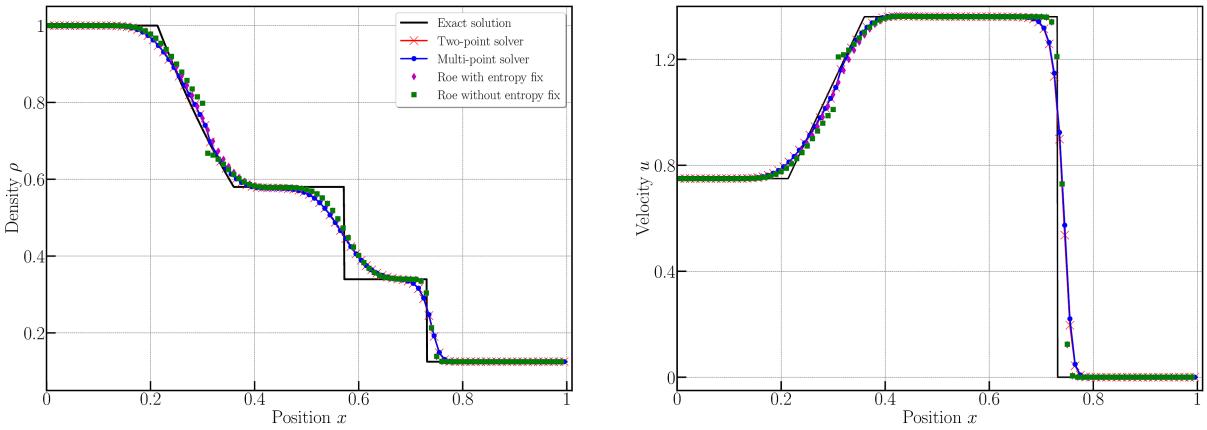


Figure 4.1: Modified Sod shock tube – Comparison of numerical solution of both two-point (red cross) and multi-point schemes (blue circle) on $N = 100 \times 100$ quadrilateral cells with the Roe Riemann solver with (purple diamond) and without entropy fix(gree square).

Radial Sod shock tube We then run a second sanity check on the cylindrical Sod shock tube problem in order to assess the ability of the scheme to capture simple but non-aligned cylindrical waves without deforming them nor producing spurious oscillations. The initial conditions for velocity component are $u, v = 0$ everywhere, while density and pressure are $(\rho_H, p_H) = (1, 1)$ for the central region $r = \sqrt{x^2 + y^2} < 0.5$ and $(\rho_L, p_L) = (0.125, 0.1)$ elsewhere. The computational domain is set to $\Omega = [0; 1] \times [0; 1]$ with symmetry boundary conditions on $x = 0$ and $y = 0$, and outflow boundary conditions otherwise. The final time is set to $t_{\text{final}} = 0.2$ with a $CFL = 0.5$. This flow can be computed also as a one-dimensional Riemann problem in the radial coordinate (see Toro (1999)) and our reference solution is computed using a MUSCL TVD scheme with 25000 quadrilateral cells. We run this test case on both two-point and multi-point solver. Figure 4.2 shows the scatter plot of the numerical density and internal energy of both the solver on 40000 quadrilateral cells. Figures (a) and (b) are comparison of the first-order two-point solver and the first-order multi-point solver with the reference solution and figures (c) and (d) are comparison of the second-order two-point solver and the second-order multi-point solver with the reference solution. The scatter plots demonstrate the efficiency of the scheme to handle cylindrical flow without any loss of symmetry. The same simulation is then done on 41404 triangular cells and the comparison of the numerical density for the different schemes are illustrated in figure 4.3, such that figure (a) is the Eul-0 two-point and multi-point scheme whereas figure (b) is the results of Eul-1

two-point and multi-point scheme. The L^1 and L^2 errors for the density ρ at final time $t_f = 0.2$ are given in table 4.1 for all the computations – First (Eul-0) and second (Eul-1) order for both two-point and multi-point scheme on 40000 quadrilateral and 41404 triangular cells. The errors are of the same order.
We also make use of this test case to compare the efficiency of both two-point and multi-point schemes in first and second order. To do so, the scheme is decomposed to few main blocks, and we analyze the CPU time taken in each step. Table 4.2 is the analysis for the two-point scheme, showing the total CPU time, number of iterations, and the CPU-decomposition for different steps, starting from the initialization, time step computation, second-order reconstruction with its limiter, numerical flux and the march equation. The efficiency of the scheme is then given by calculating the time taken for each iteration and each degree of liberty, such as $\text{eff}[\mu/iter/cell] = CPUtime/(N_{iter} \times N_{dof}) \times 1.0E06$. Similar analysis is done for the multi-point scheme as shown in table 4.3, this time involving an extra block for the computation of the nodal velocity.

2D cylindrical explosion problem - L^1 and L^2 error for ρ					
		Two-point scheme		Multi-point scheme	
		L^1 error	L^2 error	L^1 error	L^2 error
Eul-0	quad	$1.937E-01$	$2.544E-01$	$1.954E-01$	$2.567E-01$
	tri	$2.261E-01$	$2.628E-01$	$2.270E-01$	$2.614E-01$
Eul-1	quad	$1.669E-01$	$1.824E-01$	$1.678E-01$	$1.849E-01$
	tri	$7.673E-03$	$1.366E-02$	$1.504E-02$	$2.297E-02$

Table 4.1: 2D cylindrical explosion problem – L^1 and L^2 norm errors on the density ρ at $t_f = 0.2$ – Eul-0 and Eul-1 two-point scheme on 40000 quadrilateral and 41404 triangular cells and Eul-0 and Eul-1 multi-point scheme on 40000 quadrilateral and 41404 triangular cells.

Two-point scheme										
Order	CPU time [s]	N. Iter	Init.	Time per iteration[s] (CPU decomposition [%])						Efficiency [µs/iter/cell]
				Time step	P1 recon.	Limiter	Residual		u_i^{n+1}	
							States	RS		
1 (Eul-0)	38.7	847	2.32s (6%)	0.397s (1%)	-	-	0.032s (69%)	0.01s (22%)	0.0009s (2%)	1.104
2 (Eul-1)	148.65s	895	1.48s (1%)	0.002s (1%)	0.091s (55%)	0.018s (11%)	0.042s (25%)	0.009s (6%)	0.002s (1%)	4.011

Table 4.2: 2D cylindrical explosion problem – CPU time decomposition for the first order (Eul-0) and second order (Eul-1) two-point scheme with 40000 quadrilateral cells.

Slip line preservation This is a quick test case to validate the preservation of slip line at $y = 0$ for both two-point and multi-point schemes. The simulation is done on a domain of $\Omega = [0, 1] \times [-0.5, 0.5]$ on 100×100 quadrilateral cells and the interface located

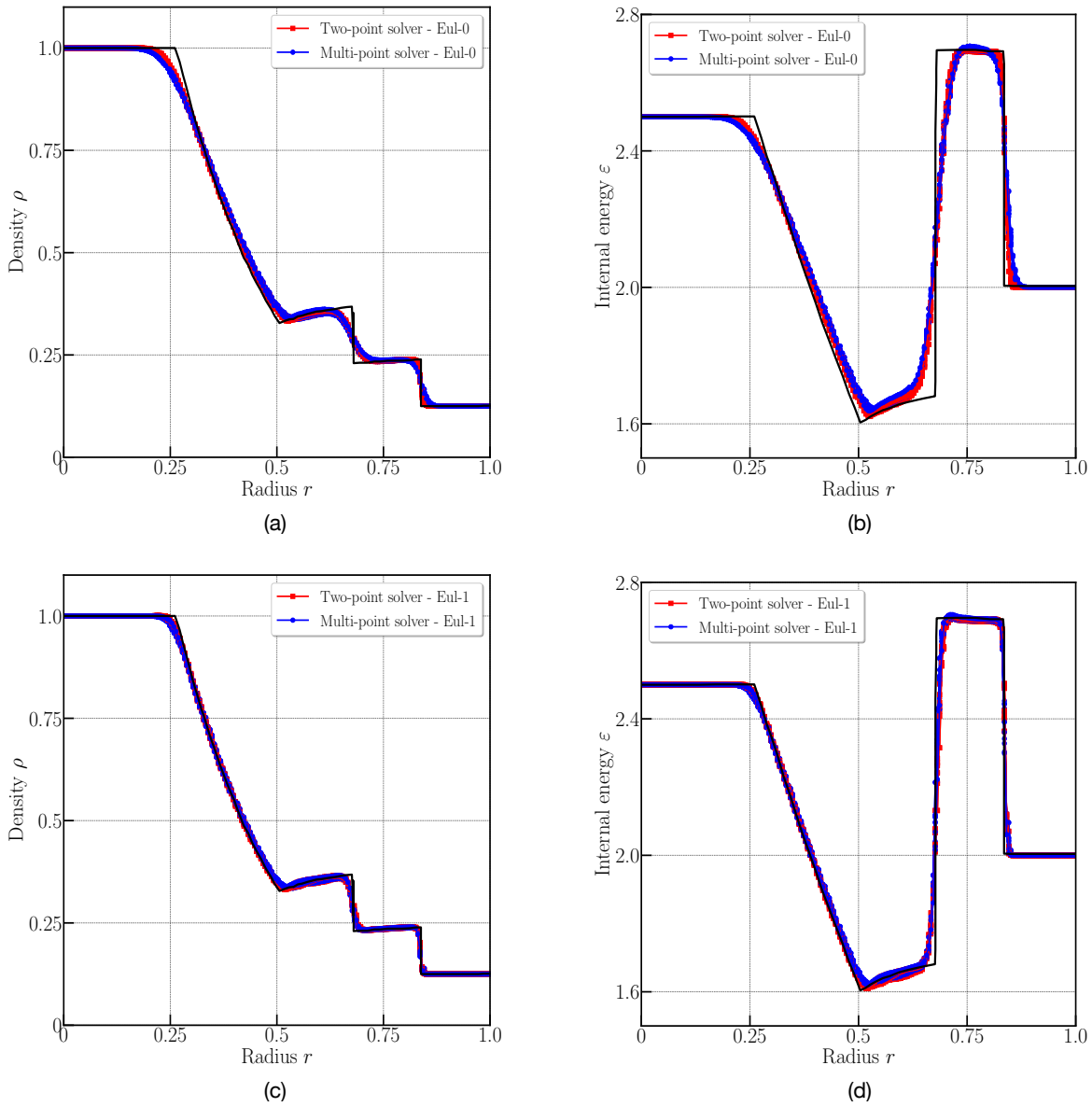


Figure 4.2: 2D cylindrical explosion problem – Comparison of numerical solution on 40000 quadrilateral cells with the reference solution (black line)– (a) Numerical density of Eul-0 Two-point solver (red square) and Eul-0 Multi-point solver (blue circle) – (b) Numerical internal energy of Eul-0 Two-point solver (red square) and Eul-0 Multi-point solver (blue circle) – (c) Numerical density of Eul-1 Two-point solver (red square) and Eul-1 Multi-point solver (blue circle) – (d) Numerical internal-energy of Eul-1 Two-point solver (red square) and Eul-1 Multi-point solver (blue circle).

at $y = 0$. The initial set up reads

$$\begin{aligned} (\rho, u, v, p)_l &= (1, -1, 0, 10) \quad , \quad \text{for } \Omega = [0,1] \times [-0.5, 0] \quad , \\ (\rho, u, v, p)_l &= (10, 1, 0, 10) \quad , \quad \text{for } \Omega = [0,1] \times [0, 0.5] \quad . \end{aligned}$$

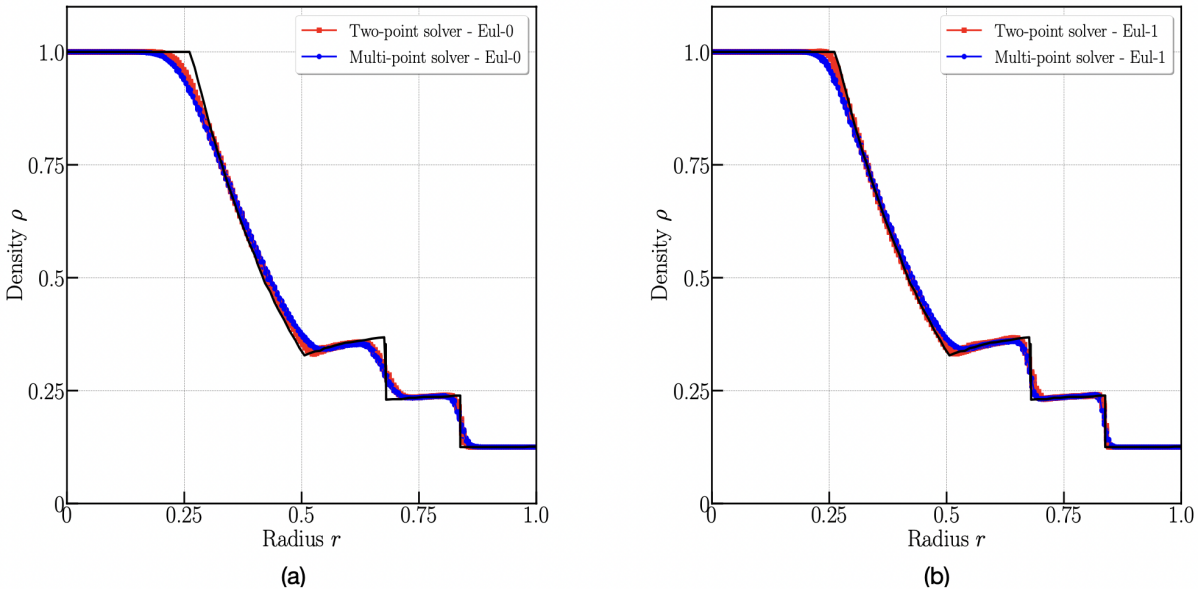


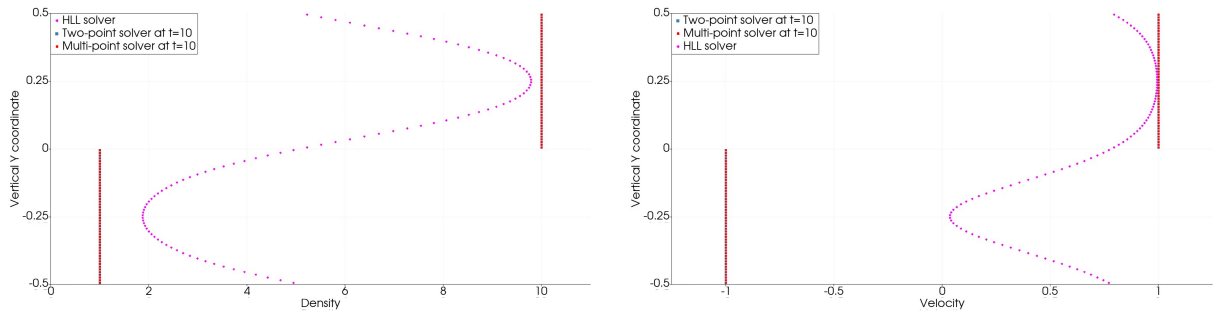
Figure 4.3: 2D cylindrical explosion problem – Comparison of numerical solution on 41404 triangular cells with the reference solution (black line)– (a) Numerical density of Eul-0 Two-point solver (red square) and Eul-0 Multi-point solver (blue circle) – (b) Numerical internal energy of Eul-1 Two-point solver (red square) and Eul-1 Multi-point solver (blue circle).

Multi-point scheme												
Order	CPU time [s]	N. Iter	Time per iteration[s] (CPU decomposition [%])									Efficiency [µs/iter/ cell]
			Time step	Nodal solver			P1 recon.	Limiter	Residual		u_i^{n+1}	
				States	Dukowicz	Nodal velocity			States	RS		
1	176.51s	829	1.765s (1%)	0.066s (26%)	0.023s (11%)	0.066s (31%)	-	-	0.043s (20%)	0.021s (10%)	0.002s (1%)	5.14
2	391.28s	845	3.913s (1%)	0.107s (23%)	0.027s (6%)	0.148s (32%)	0.088s (19%)	0.019s (4%)	0.046s (10%)	0.019s (4%)	0.005s (1%)	11.184

Table 4.3: 2D cylindrical explosion problem –CPU time decomposition for the first order (Eul-0) and second order (Eul-1) multi-point scheme with 40000 quadrilateral cells.

A wall boundary condition is imposed on $y = -0.5$ and $y = 0.5$ and a periodic boundary condition on $x = 0$ and $x = 1$ is imposed. We compare the results of the two-point and multi-point scheme with the numerical solution of the HLL solver since the HLL solver is known to not take into account the contact discontinuities. Figure 4.4(a) shows the density and 4.4(b) the velocity at $y = 0$. The results show that the slip line is correctly preserved for both two-point and multi-point scheme contrary to the one of the HLL solver.

Isentropic vortex in motion The isentropic vortex problem from Shu (1997) tests the accuracy of numerical methods since an exact, smooth and analytic solution exists. The



(a) Numerical density. (b) Numerical velocity.

Figure 4.4: Slip line preservation — Comparison of numerical solution for density and velocity at $y = 0$ -axis on 100×100 quadrilateral cells compared with the exact solution (black line) and the L_2 error for the 1^{st} and 2^{nd} order schemes for both two-point and multi-point scheme.

computational domain is set to $\Omega = [-5, 5] \times [-5, 5]$. The ambient flow is characterized by $\rho_\infty = 1.0$, $u_\infty = 1.0$, $v_\infty = 1.0$, $w_\infty = 1.0$ and $p_\infty = 1.0$, with a normalized ambient temperature $T^*_\infty = 1.0$. At the initial time $t = 0$, onto this ambient flow is superimposed a vortex centered at $(0, 0)$ with the following state: $u = u_\infty + \delta u$, $v = v_\infty + \delta v$, $T^* = T^*_\infty + \delta T^*$, where the increments are given by

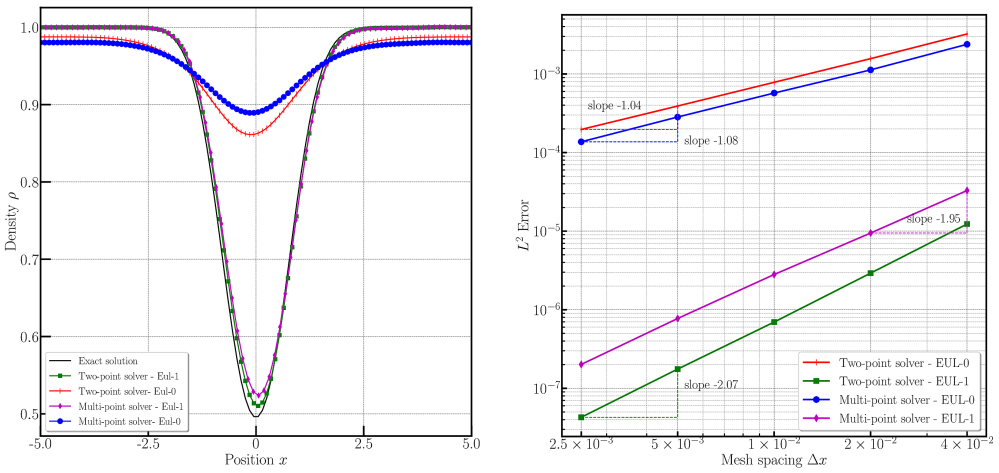
$$\delta u = -y' \frac{\beta}{2\pi} \exp\left(\frac{1-r^2}{2}\right), \quad \delta v = x' \frac{\beta}{2\pi} \exp\left(\frac{1-r^2}{2}\right), \quad \delta T^* = -\frac{(\gamma-1)\beta^2}{8\gamma\pi^2} \exp\left(1-r^2\right),$$

with $r = \sqrt{x^2 + y^2}$. The so-called strength of the vortex is set to $\beta = 5.0$ and the initial density is given by $\rho = \rho_\infty \left(T^*/T^*_\infty\right)^{\frac{1}{\gamma-1}}$. Periodic boundary conditions are prescribed. At final time $t = t_{\text{final}} = 1$ the vortex is back to its original position and the final solution matches the initial one. Because the solution is smooth, it should be simulated with optimal high accuracy, in other words the limiting/stabilization procedure employed in the scheme should not have any effect. We run the isentropic vortex test case with first order (Eul-0) and second order extension (Eul-1) for both two-point scheme and multi-point scheme on quadrilateral cells going from 25×25, 50×50, 100×100, 200×200 and 400×400. The density at $t_{final} = 10$ is given in figure 4.5. The results of the convergence analysis for both two-point scheme and multi-point scheme and are displayed in table 4.4 and shows that the expected rate of convergence is reached. The lower precision of the multi-point scheme observed is due to the numerical dissipation of the numerical scheme.

Odd-even decoupling The odd-even decoupling problem is a difficult simulation for certain numerical schemes. Quirk (1994) proposed this so-called odd-even decoupling test, where one simulates the propagation of a planar shock but on a perturbed Cartesian mesh for which only the center line is slightly shifted. The computational domain is defined by $\Omega = [0, 800] \times [0, 20]$ with a Cartesian uniform structured mesh with $\Delta x = \Delta y = 1$, with $I = 800$ cells in x-direction, $J = 20$ in y-direction. A cell is then labelled with i, j indices as usual for Cartesian mesh. The center-line, at $y = 10$, coincides with the j_0 labeled line

2D Isentropic Vortex in motion - Rate of convergence					
		Two-point scheme		**Multi-point scheme**	
	N	L^2 error	L^2 order	L^2 error	L^2 order
Eul-0	25×25	$3.2265E-03$	–	$2.3836E-03$	–
	50×50	$1.5660E-03$	1.04	$1.1269E-03$	1.08
	100×100	$7.8289E-04$	1.00	$5.5048E-04$	1.03
	200×200	$3.9078E-04$	1.00	$2.7316E-04$	1.01
	400×400	$1.9656E-04$	0.99	$1.3518E-04$	1.01
Eul-1	25×25	$1.2312E-05$	–	$3.2875E-05$	–
	50×50	$2.9202E-06$	2.07	$1.9563E-06$	1.79
	100×100	$6.9651E-07$	2.07	$2.8055E-06$	1.75
	200×200	$1.7560E-07$	1.98	$7.7364E-07$	1.86
	400×400	$4.2756E-08$	2.04	$2.0124E-07$	1.95

Table 4.4: Isentropic vortex in motion — L^2 norm errors on density ρ between the numerical solution and the exact solution of the isentropic vortex in motion problem at $t_{final} = 10$ on quadrilateral cells.



(a) Density cut through x-axis. (b) L_2 error.

Figure 4.5: Isentropic vortex in motion — Numerical solution for density cut through the x-axis on 100×100 quadrilateral cells compared with the exact solution (black line) and the L_2 error for the 1^{st} and 2^{nd} order schemes for both two-point and multi-point scheme.

of mesh with $j_0 = 10$. The perturbation of amplitude 10^{-6} is alternating as follows for line j_0:

$$\begin{aligned} x_{i,j_0} &\longleftarrow x_i, \\ y_{i,j_0} &\longleftarrow y_{j_0} + (-1)^i, 10^{-6}. \end{aligned} \tag{4.1.37}$$

The shock wave is traveling from left to right with a Mach number of $Ma = 6$, the initial domain is filled with a diatomic gas with $(\rho^0, u^0, v^0, p^0, \gamma) = (1, 0, 0, 1, 1.4)$. The left side

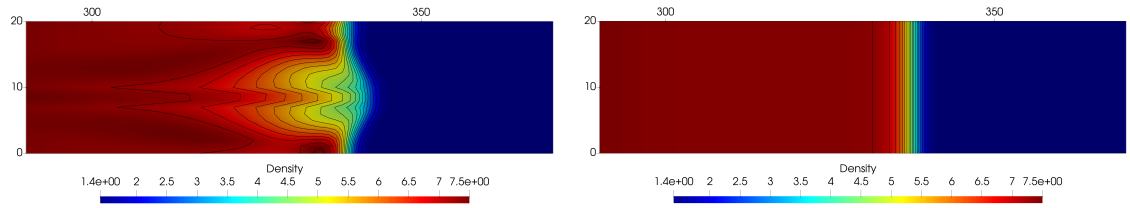
inflow values are determined using the Rankine-Hugoniot relations:

$$\begin{aligned} u_s &= Ma\sqrt{\gamma}, \\ \rho_\infty &= \frac{(\gamma+1)Ma^2}{(\gamma-1)Ma^2+2}, \\ u_\infty &= u_s\frac{2(Ma^2-1)}{(\gamma+1)Ma^2}, \\ p_\infty &= \frac{2\gamma Ma^2-(\gamma-1)}{(\gamma+1)}, \end{aligned}$$

The left Boundary Condition (BC) corresponds to inflow boundary, top and bottom are wall-type BCs while the right one is left open. Obviously the exact solution is a vertical $Ma = 6$ shock wave traveling at speed u_s. The instability reveals itself through symmetry perturbation of the shock wave as it travels along the duct. A sensible measure of this perturbation, see Rodionov (2017), consists in computing the deviation ε_0 of the numerical solution:

$$\varepsilon_0 = \max_{i,j}(|\rho_{i,j} - \overline{\rho}_i|), \qquad \text{with} \qquad \overline{\rho}_i = \frac{1}{J}\sum_{j=1}^{J}\rho_{i,j}. \tag{4.1.38}$$

The density color maps at the stopping time are displayed in figure 4.6 respectively for the two-point (left) and the multi-point (right) schemes. The density contours have been plotted using 20 isolines in the range $[1.4, 7.3]$. The development of the odd-even instability is clear for the two-point scheme whereas it seems to be absent for the multi-point scheme. In figure 4.7 one displays the plot of ε_0 as a function of the distance traveled by the shock wave, $X_s(t) = u_s t$, for the two-point scheme (red curve) and the multi-point one (blue curve). The presence of the instability can be quantified with the plot of ε_0. We observe a fast increase up to $\varepsilon_0 > 1 = \Delta x$ for the two-point scheme, while the multi-point scheme value remains consistently at the order $10^{-4} \ll \Delta x$.



(a) Two-point scheme. (b) Multi-point scheme

Figure 4.6: Odd-even decoupling — Density contour of odd-even decoupling test case.

A more demanding version of the odd-even decoupling problem has been proposed by Rodionov (2017) by changing the seed of the instability. Here, we simulate the propagation of the same shock wave. However a small perturbation of the form

$$\hat{x}_{i_0,j} = x_{i_0,j} + 10^{-4}(2\zeta_j - 1)$$

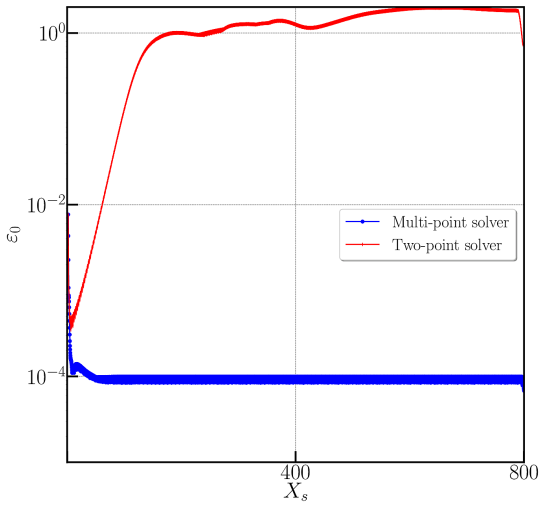
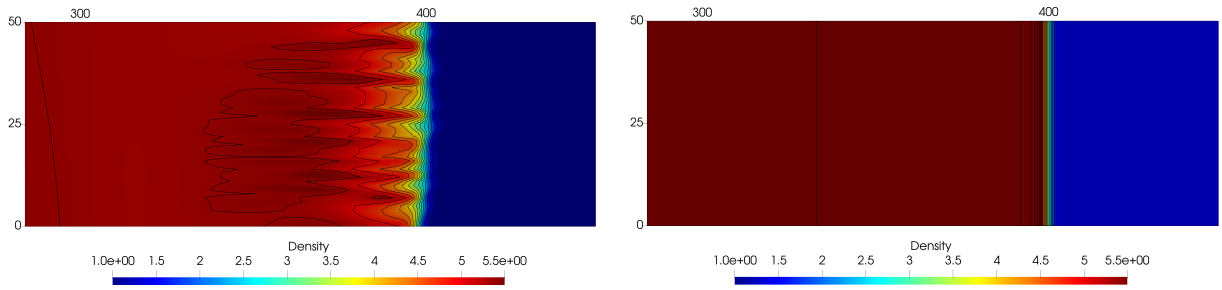


Figure 4.7: Odd-even decoupling — Comparison of solution deviation.

is introduced in the transverse grid line $i_0 = 10$, where $\zeta_j \in [0, 1]$ are random numbers. The y size of the domain is increased so that $J = 50$ and one maintains $\Delta x = \Delta y = 1$. The upper and lower BCs are periodic ones. The results are displayed as previously for the odd-even decoupling case. The instability is much more pronounced with higher frequency for the two-point scheme. Contrarily the multi-point scheme does not present such an amplification of the initial perturbations, see figure 4.8. This is also observed on the ε_0 curves, refer to figure 4.9.



(a) Two-point scheme. (b) Multi-point scheme.

Figure 4.8: Modified odd-even decoupling — Density contour of modified odd-even decoupling test case.

Hypersonic flow over half cylinder The hypersonic flow over a half cylinder test case is a well-documented test case to challenge numerical methods. In particular for this flow some schemes may develop the infamous carbuncle phenomena when classical face-based Finite Volume upwind schemes are employed. Instead of having a smooth bow shock profile upstream of the half cylinder, the carbuncle failing features a pair of oblique shock ahead of the stagnation region, compromising the overall flow predictions around the cylinder, refer for instance to Peery & Imlay (1988); Gaitonde (1992); Pandolfi

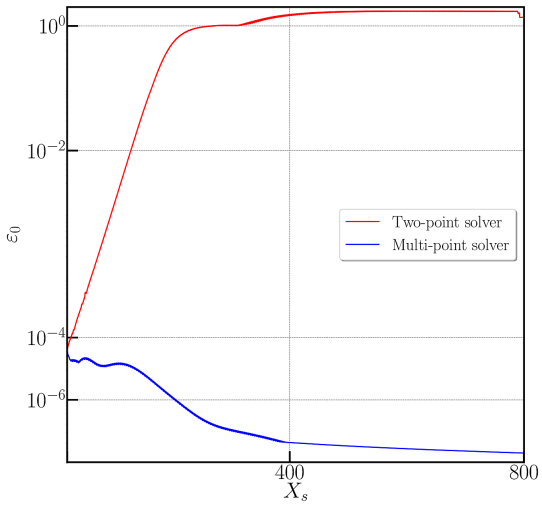
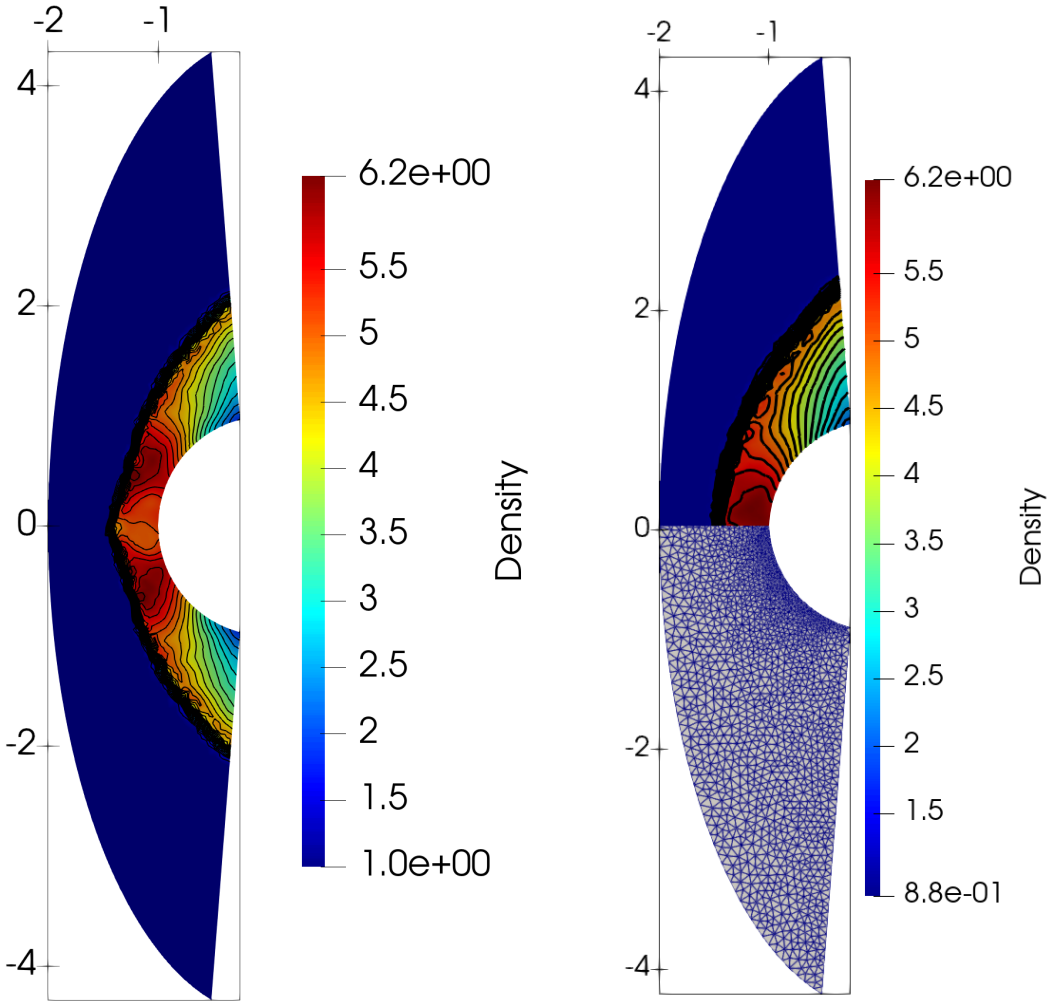


Figure 4.9: Modified odd-even decoupling — Comparison of the solution deviations ε_0.

& D'Ambrosio (2001) just to cite few references. Here, following Rodionov (2017), we simulate an inviscid flow at Mach $Ma = 20$ around a half cylinder blunt body subject to a incoming hypersonic flow characterized by $(\rho_0, u_0, v_0, p_0, \gamma) = (1, Ma\sqrt{\gamma}, 0, 1, 1.4)$. The steady-state resulting flow is simulated by means of an explicit time marching procedure, that is, the simulation ends when the residual is 6 orders of magnitude smaller than its initial value. The computational domain covers a large enough domain which contains half of a cylinder centered at the origin with a radius $R = 1$, and a left incoming hypersonic flow. At the cylinder surface a wall-type boundary condition is considered, while bottom/upper boundary conditions are free outflow and inflow condition at the left boundary. Three types grid are tested to assess the robustness of our unstructured Finite Volume multi-point scheme: a triangular grid is composed of 5671 unstructured triangles, a quadrangular one made of 5000 structured quadrangles, and, a polygonal grid with 5632 unstructured polygons.
Firstly, this test case is ran with the two-point scheme over the triangular grid, and, as can be observed in figure 4.10(a) the carbuncle instability clearly develops. On the contrary, such instability does not develop with the multi-point scheme over the triangular, quadrangular and polygonal grids, refer respectively to figures 4.10(b), 4.11(a) and 4.11(b). A quantitative comparison of these numerical results can be achieved by plotting the pressure coefficient, $C_p = \frac{p-p_0}{\frac{1}{2}\rho_0 u_0^2}$, at the wall with respect to the angular position. The numerical C_p is then compared to the approximate analytical value coming from the Newtonian theory, see Anderson (2006). The pressure coefficient, C_p, computed by the multi-point scheme matches the Newtonian theory for angles below 40°, while the two-point scheme results are polluted by the carbuncle effect, refer to figure 4.12(a). A rather good convergence is observed for the multi-point scheme in figure 4.12(b) for successively refined structured grids made of 25×50, 50×100 and 100×200 quadrangular cells.

We pursue our numerical study comparing both two-point and multi-point schemes over the same polygonal grid for the half cylinder test case. It is worth mentioning that the polygonal grid results from a Voronoi tessellation. For this particular grid, a

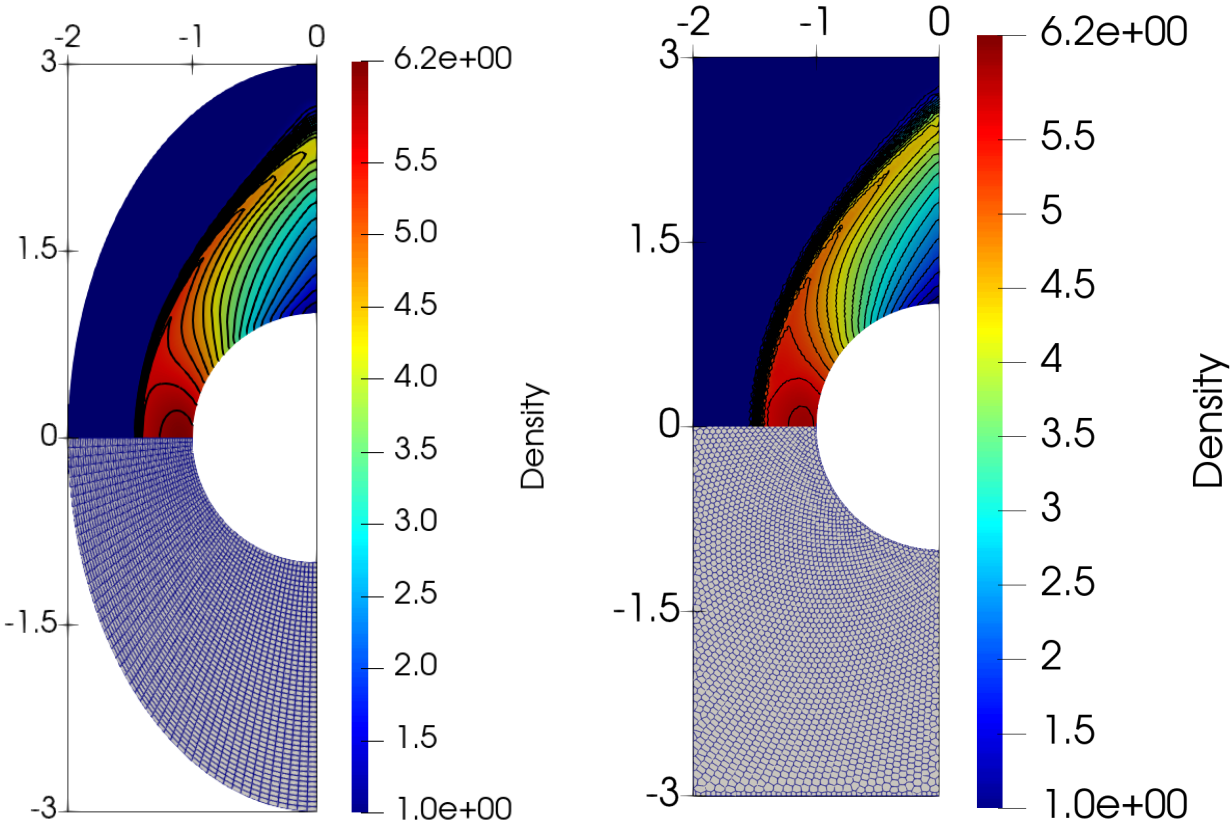


(a)Triangular grid, two-point scheme. (b) Triangular grid, multi-point scheme.

Figure 4.10: Hypersonic flow over half cylinder — Density maps over triangular grids for both two-point and multi-point schemes with 20 isolines over the interval $[1, 6.2]$.

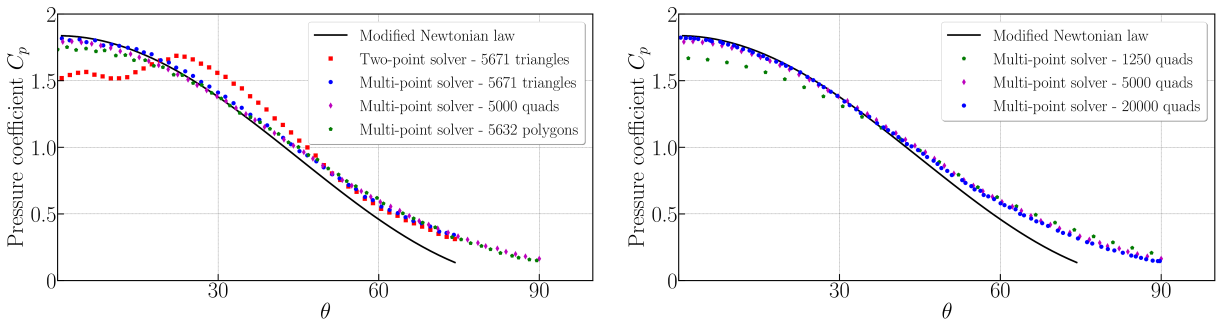
generic polygonal cell has exactly the same number of face-based or vertex-based neighbors. This implies, that the two-point Finite Volume scheme and the multi-point Finite Volume scheme share the very same stencil. The density map obtained with both schemes (top/bottom panels: two/multi-point schemes) over the polygonal grid under consideration is displayed in figure 4.13. 20 isolines ranging from $\rho = 1$ to $\rho = 6.2$ are plotted. Although the two-point scheme does not exhibit large spurious spike as the one observed in figure 4.14(a) , the isolines exhibit emerging instabilities behind the shock, see also the 3D view in figure 4.14. This test case demonstrates that the multi-point solver does not generate any carbuncle-like instability on any type of grid. As such one has to be careful when praising the low-dissipation property of a solver such as the two-point one.

Sedov blast wave problem The Sedov test case is a point explosion problem for which an exact solution can be derived, refer to Sedov (1959). In the case of two-dimensional pla-



(a) Quadrangular grid, multi-point scheme. (b) Polygonal grid, multi-point scheme.

Figure 4.11: Hypersonic flow over half cylinder — Density maps over quadrangular and polygonal grids for both multi-point scheme with 20 isolines over the interval $[1, 6.2]$.



(a) C_p comparisons. (b) Grid convergence.

Figure 4.12: Hypersonic flow over half cylinder — Various plots of the pressure coefficient C_p at the wall with respect to the angular position employing the foregoing grids.

nar geometry, the problem consists of a cylindrical explosion generating a diverging shock wave. The computational domain $\Omega = [-1.2, 1.2] \times [-1.2, 1.2]$ is initially filled with a perfect gas at rest characterized by the initial conditions $(\rho^0, u^0, v^0, p^0, \gamma) = (1, 0, 0, 10^{-6}, \frac{7}{5})$. The point explosion is initiated by an energy deposition in the vicinity of the origin, *i.e.*, the pressure in the cells in contact with the origin is set to the value 0.397056. Reflective

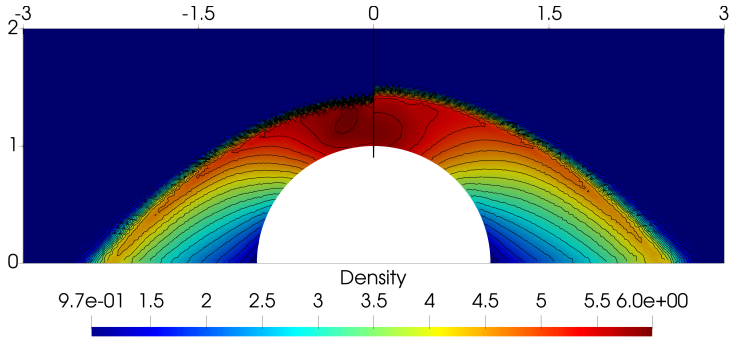
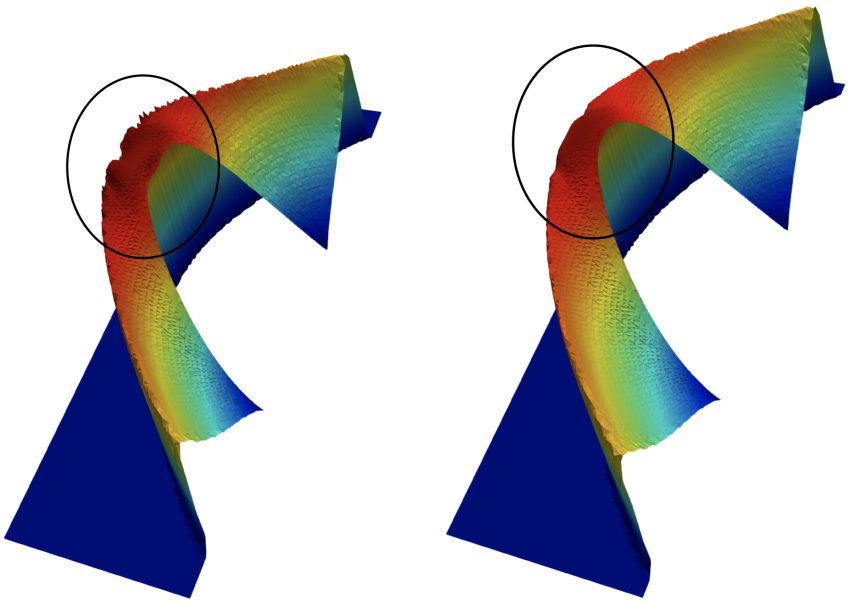


Figure 4.13: Hypersonic flow over half cylinder — Comparison between density map on polygonal grid: two-point scheme (left) and multi-point scheme (right).

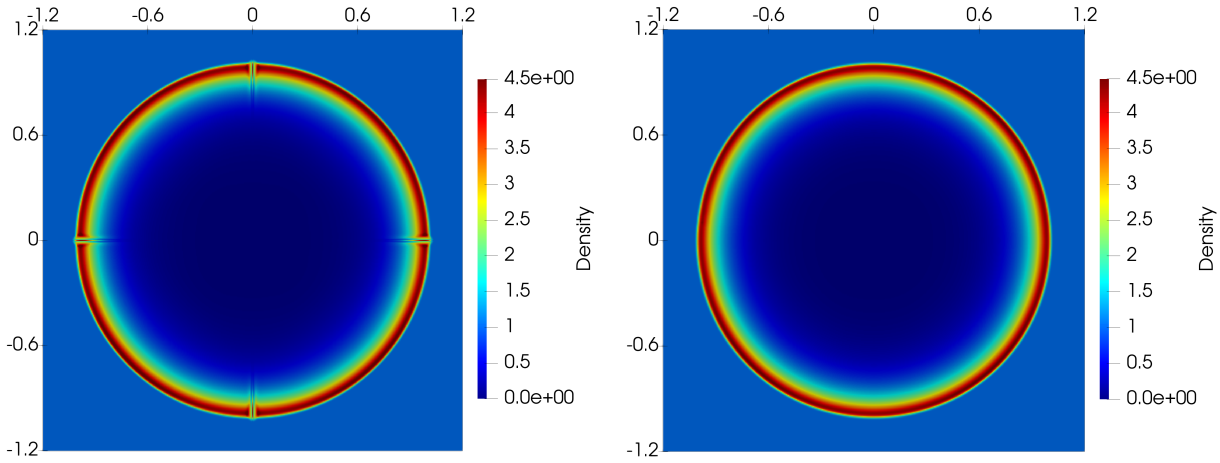


(a) Two-point scheme (b) Multi-point scheme.

Figure 4.14: Hypersonic flow over half cylinder — 3D extruded views for the density field obtained with both two-point and multi-point schemes over the same polygonal grid.

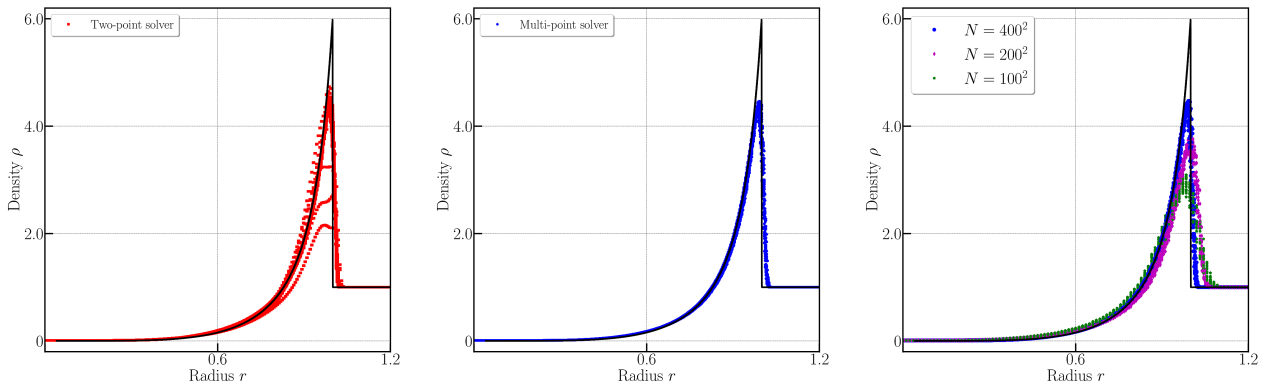
boundary conditions are applied on all boundaries. With this particular setup taken from Kamm & Timmes (2007), the cylindrical shock wave radius is $r_{\text{shock}} = \sqrt{x^2 + y^2} = 1$ at the final time $t_{\text{final}} = 1$ with a peak density $\rho^{\max} = \frac{\gamma+1}{\gamma-1} = 6$. The exact density as a function of radius is plotted using a continuous black line in figure 4.16.
Similarly to what has been done for the previous test cases, we compared the numerical solutions obtained by the two-point and multi-point schemes on 400×400 uniform quadrangles in figure 4.15(a) and in figure 4.15(b) respectively. Then, the scattered plot of density with respect to the radius of each cell center is displayed in figures 4.16(a) and 4.16(b) for both schemes versus the analytical solution. These results exhibit the occurrence of a spurious effect along the x and y axes for the two-point scheme. Notice that the full 2π Sedov problem is actually run so that the boundary conditions cannot

be blamed. The scattered plots also confirm the loss of cylindrical symmetry for the two-point scheme. Contrarily, the multi-point scheme does not present such parasitical effect, and, as such preserves the cylindrical symmetry, while being slightly more dissipative. These observations have already been done in Rodionov (2017) for instance. In order to confirm such a good behavior, the multi-point scheme is used on the following successively refined quadrangular grids: 100×100, 200×200 and 400×400. The resulting scattered plots are displayed in figure 4.16(c), where we observe a convergence of the solution without spurious effects.



(a) Two-point scheme (b) Multi-point scheme.

Figure 4.15: Sedov blast wave problem — Density contour at $t_{\text{final}} = 1$ on a 400×400 grid.



(a) Two-point scheme (b) Multi-point scheme (c) Grid convergence.

Figure 4.16: Sedov blast wave problem — Scattered plots for density.

Noh implosion problem The problem designed by Noh (1987) is a well known test case used to validate numerical schemes in the regime of infinite strength shock wave. The initial computational domain is defined by a quarter of a circle of radius of $r = 1$. A perfect gas with $\gamma = 5/3$ is initially assigned with a uniform density $\rho^0 = 1$ and a

unit radial inward pointing velocity, hence the velocity components are initialized with $\mathbf{v}^0(x,y) = (-x/r, -y/r)$ where $r = \sqrt{x^2+y^2}$ denotes the radius and $|\mathbf{v}^0| = 1$. The initial pressure is $p^0 = 10^{-6}$ everywhere. A diverging cylindrical shock wave is generated at the origin. The state behind the shock wave is constant, whereas the state in front is not anymore due to the cylindrical converging flow. Thus, the exact solution at time $t > 0$ writes

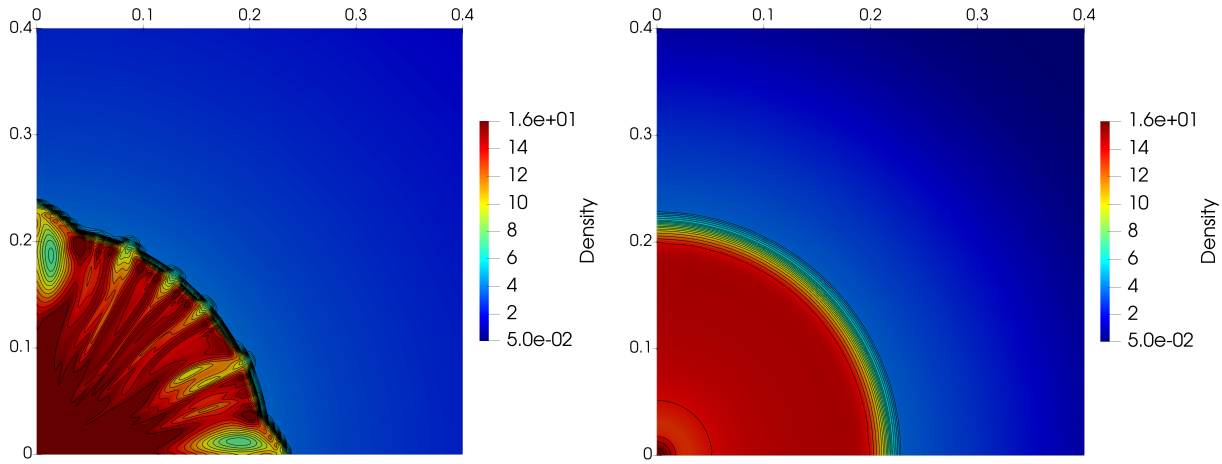
$$(\rho, \mathbf{v}, p)^{\text{ex}}(t) = \begin{cases} \left(\rho_0 \left(\dfrac{\gamma+1}{\gamma-1}\right)^2,\ \mathbf{0},\ \dfrac{1}{2}\rho_0 \dfrac{(\gamma+1)^2}{\gamma-1}\right), & \text{if} \quad r < r_s, \\ \left(\rho_0 \left(1 - \left(\dfrac{t}{r}\right)\right)^2, \mathbf{v}_0, 10^{-6}\right), & \text{if} \quad r \geq r_s. \end{cases} \tag{4.1.39}$$

Here, the radius of the shock wave is $r_s(t) = v_s t$ with $v_s = \dfrac{1}{2}(\gamma-1)|\mathbf{v}^0|$. With this setup and the final time $t_{\text{final}} = 0.6$, we arrive at $r_s = 0.2$ and a post-shock state characterized by the values $\rho^{\text{ex}} = 16$, $\mathbf{v}^{\text{ex}} = \mathbf{0}$ and $p^{\text{ex}} = \dfrac{16}{3}$. Symmetry BCs are prescribed on the axis $x = 0$ and $y = 0$, whereas at the outer radius, the exact space/time dependent velocity is imposed employing the analytical solution (4.1.39). A radial/polar grid made of $N \times N$ quadrangles is constructed such that $\Delta r = 1/N$ and $\Delta\theta = \pi/(4N)$ are constants. Let us notice that the cells in contact with the origin are triangles which are considered as degenerated quadrangles. We present the numerical density (colors and isolines) for a $N = 200 \times 200$ polar grid at the final time $t_{\text{final}} = 0.6$ for the two-point and the multi-point schemes in figures 4.17 and 4.17(b) respectively. The two-point scheme produces strong post-shock instabilities. We have verified that those instabilities are not generated by inappropriate boundary conditions. Contrarily the multi-point scheme presents a smooth symmetrical solution. Consistently with the results of Sedov problem, we present the scattered plot of the density with respect to the radius of the cell center for both schemes in figure 4.18(a). Again we observe that the multi-point scheme handles such cylindrical flow without any loss of symmetry. Finally, a grid convergence analysis for the multi-point scheme is performed using the sequence of grids characterized by 50^2, 100^2 and 200^2 polar cells. The results for the density are plotted in figure 4.18(b). We point out that the multi-point scheme indeed converges towards the exact solution. Note that the undershoot close to the origin corresponds to the so called wall-heating effect, refer to Menikoff (1994); Rider (2000).

Forward-facing step Next we run the forward facing step problem proposed by Woodward & Colella (1984). This test case is a Mach 3 wind tunnel with a step. The computational domain is given by $\Omega = [0,3] \times [0,1]$ $[0.6,3] \times [0,0.2]$ with the following initial conditions

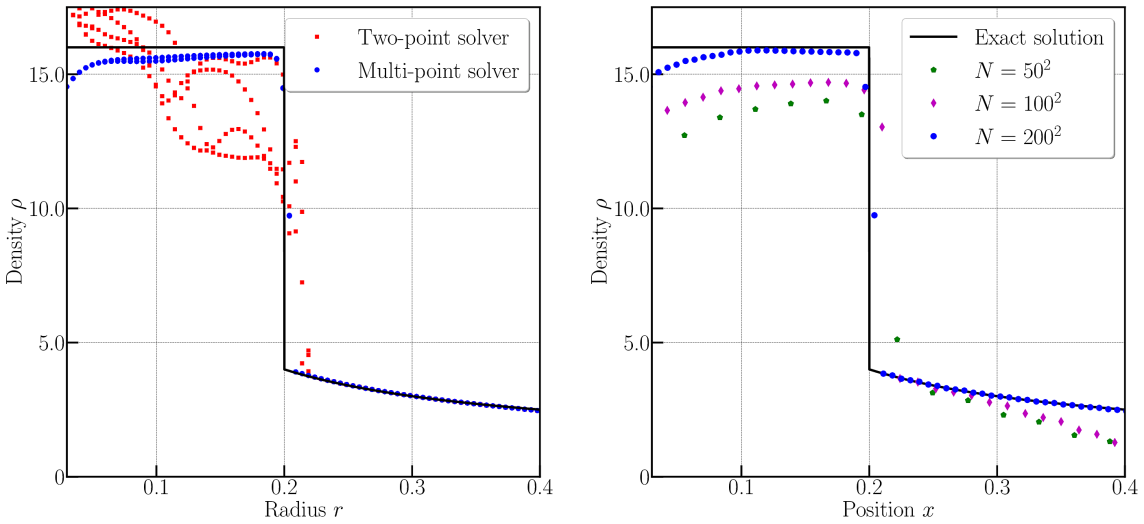
$$\rho = \gamma, \quad \mathbf{v} = (3,0)^t, \quad p = 1, \quad \gamma = 7/5.$$

The final time is $t_{\text{final}} = 4$ and reflective boundary conditions are applied on the upper and lower boundaries of the domain, whereas inflow and outflow boundary conditions are



(a) Two-point scheme (b) Multi-point scheme.

Figure 4.17: Noh implosion problem on polar grids — Density maps for a 200×200 quadrilateral polar grid.



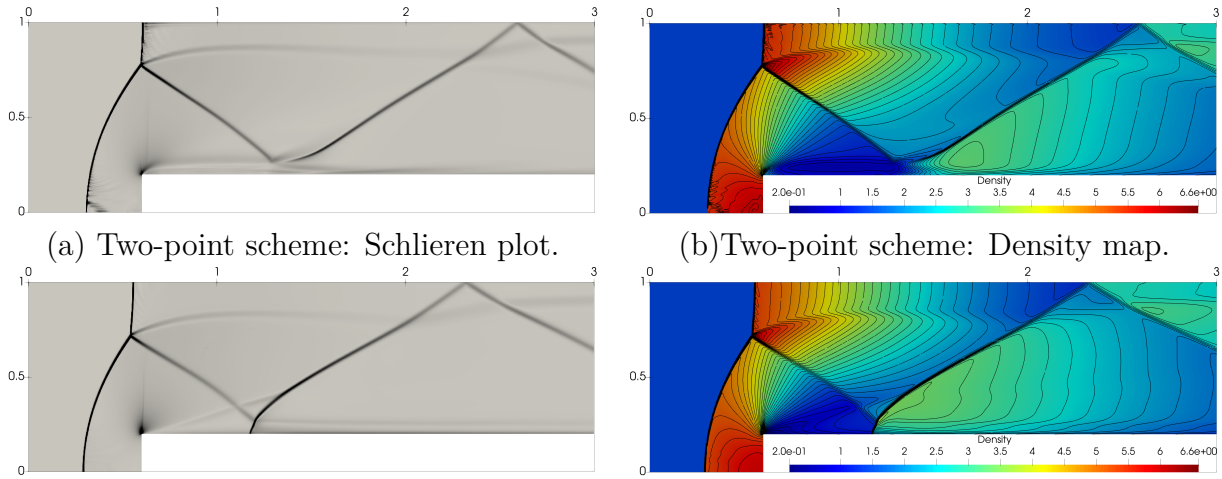
(a) Numerical density. (b) Grid convergence.

Figure 4.18: Noh implosion problem — Scattered plots of the density.

applied at the left entry and the right exit respectively. The solution presents several shock waves further interacting with the wall boundaries. Once again we use this test case to compare both the two-point and multi-point schemes on meshes constituted of quadrangles.

In figures 4.19(a) and 4.19(b), (resp. figures 4.19(a) and 4.19(b)) we present the results for the two-point schemes (resp. multi-point schemes) employing a structured grid made of $N = 334043$ quadrilateral cells. The density gradient (numerical Schlieren) and density contours (colors and iso-contours) are both illustrated. The general shape and position of the multiple shocks appear correctly captured by the two schemes. However, we also observe in details that the two-point scheme develops some spurious phenomena along

the flat step up to the reflection Y-shaped region, and, along the primary bow shock, especially close to the $y = 0$ axe. The numerical Schlieren plot enhances these behaviors. On the contrary the multi-point scheme in figures 4.19(c) and 4.19(d) does not present this pathological behavior.



(a) Two-point scheme: Schlieren plot.

(b)Two-point scheme: Density map.

(c) Multi-point scheme: Schlieren plot.

(d) Multi-point scheme: Density map.

Figure 4.19: Forward-facing step — Numerical results of both two-point (top) and multi=-point schemes (bottom) on structured grid composed of $N = 334043$ quadrilateral cells.

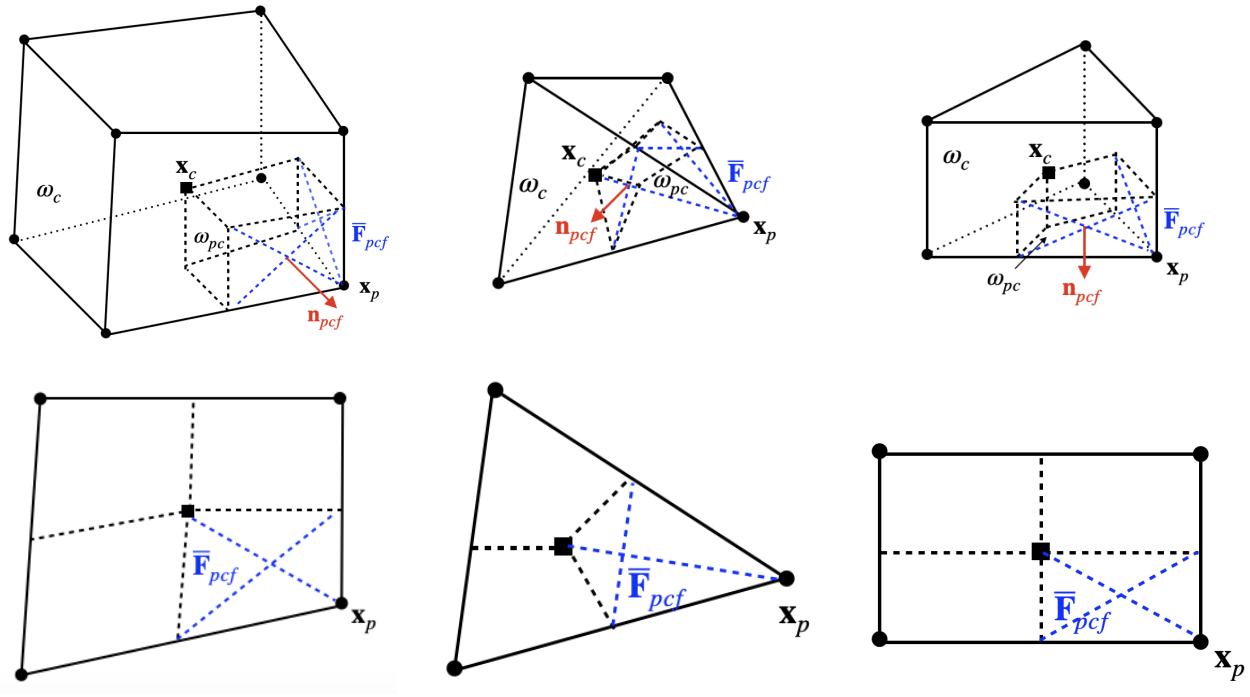
4.2 Application to three-dimensional gas dynamics equations

As a sequel of the study for the multidimensional Godunov-type scheme, we apply the scheme on the three-dimensional gas dynamics equations. This is a straightforward extension of the two-dimensional one by defining $d = 3$ and instead of computing along the length of an edge, the discretization is now based on a surface of a face.

4.2.1 Governing equations and notations

The spatial domain consists of non-overlapping unstructured cells and ω_c denotes a generic polyhedral cell. These cells can be hexahedrals, tetrahedrals, prisms or pyramids, such as illustrated in figure 4.20. We introduce the set $\mathcal{F}(c)$ as the list of faces of cell c and the set $\mathcal{SF}(pc)$ the list of faces of cell c impinging at point p. The former set is linked to the latter by $\mathcal{F}(c) = \cup_{p\in\mathcal{P}(c)}\mathcal{SF}(pc)$. Here, we consider a mesh composed of polyhedral cells and the term polyhedral cells signify a volume enclosed by an arbitrary number of faces, each determined by an arbitrary number of vertices. For instance, a face with four or more vertices can be non-coplanar, thus the face is not a plane and the unit outward normal is difficult to be defined. The polyhedral cell is therefore decomposed into elementary subcells. To do so, for a given polyhedral cell c, we consider the vertex

$p \in \mathcal{P}(c)$ that belongs to the face $f \in \mathcal{F}(c)$ and the edge e, and a subcell is constructed by connecting p, the centroid of cell c, midpoints of each surface related to point p and the midpoint of edge e. The set of subfaces attached to the corner pc are the set of faces of subcell ω_{pc} impinging at point p and the cell volume is defined by $|\omega_c| = \sum_{p\in\mathcal{P}(c)} |\omega_{pc}|$. The surface area and the unit outward normal of the subface f are denoted as A_{pcf} and $\mathbf{n}_{pcf}$ respectively. Here, we assume that the faces are coplanar.



(a) Hexahedron. (b) Tetrahedron. (c) Prism.

Figure 4.20: Geometrical entities attached to a cell ω_c (top) and its subface fluxes partition of a face f of cell ω_c (bottom).

The three-dimensional gas dynamics equation under the Eulerian representation reads

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbb{F}(\mathbf{U}) = \mathbf{0}. \tag{4.2.1}$$

This system now has three components in velocity and the vector of conservative variables and flux projected on the normal direction $\mathbf{n}$ write

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho v_{\mathbf{n}} \\ \rho \mathbf{v_t} \\ \rho e \end{pmatrix}, \quad \mathbf{F_n} = \mathbb{F}\mathbf{n} = \begin{pmatrix} \rho v_{\mathbf{n}} \\ \rho v_{\mathbf{n}}^2 + p \\ \rho v_{\mathbf{n}} \mathbf{v_t} \\ \rho v_{\mathbf{n}} e + p v_{\mathbf{n}} \end{pmatrix}, \tag{4.2.2}$$

with the velocity expressed as $\mathbf{v} = v_{\mathbf{n}}\mathbf{n} + \mathbf{v_t}$, where $v_{\mathbf{n}} = \mathbf{v} \cdot \mathbf{n}$. Therefore, the one-dimensional system associated to this three-dimensional gas dynamics system (4.2.1) in

the normal direction where $x_{\mathbf{n}} = \mathbf{x} \cdot \mathbf{n}$ reads

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_{\mathbf{n}}}{\partial x_{\mathbf{n}}} = \mathbf{0}. \tag{4.2.3}$$

With these notations, we build a three-dimensional subface-based Finite Volume scheme. A numerical simulation code associated to this scheme is then developed. Due to the huge amount of data, an efficient parallel numerical method is required. In the following section, we present the summary of the three-dimensional Godunov-type scheme before introducing a few important libraries from the open-source PETSc data management tool to build the three-dimensional code.

4.2.2 Summary of the three-dimensional Eulerian Godunov-type scheme

Following 4.1.6, we recall that the three-dimensional extension of the previous scheme with obvious notation adaptation writes

$$\mathbf{U}_c^{n+1} - \mathbf{U}_c^n + \frac{\Delta t}{|\omega_c|} \sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{SF}(pc)} A_{pcf} \overline{\mathbf{F}}_{pcf} = \mathbf{0},$$

with $\overline{\mathbf{F}}_{pcf}$ the left flux with respect to the subface f and the unit outward normal $\mathbf{n}_{pcf}$. Note that the left and the right states correspond respectively to the cells c and d since the unit normal $\mathbf{n}_{pcf}$ is pointing from cell c towards cell d. The left flux for a simple Eulerian Riemann solver yields

$$\begin{aligned} \overline{\mathbf{F}}^{-}_{\mathbf{n}_{pcf}} =& \frac{1}{2}\left[\mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_c) + \mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_d)\right] - \frac{1}{2}\left[\sum_{k=1}^{m} |\Lambda_k| (\mathbf{U}_{k+1} - \mathbf{U}_k)\right]_{c,d} \\ & - \frac{1}{2}(\lambda_{c(d,f)} + \lambda_{d(c,f)})\left[\mathbf{v}_p \cdot \mathbf{n}_{pcf} - \overline{v}_{\mathbf{n}_{pcf}}\right] \begin{pmatrix} 0 \\ 1 \\ \mathbf{0} \\ \mathbf{v}_p \cdot \mathbf{n}_{pcf} \end{pmatrix}. \end{aligned} \tag{4.2.4}$$

This explicit expression of the numerical flux shows that in the case $\mathbf{v}_p \cdot \mathbf{n}_{pcf} = \overline{v}_{\mathbf{n}_{pcf}}$ we retrieve the classical conservative face-based Finite Volume scheme.

The approximation of the vectorial nodal velocity parameter $\mathbf{v}_p$ is retrieved by the node-based conservation condition of the numerical scheme as mentioned in section 3.2.6. The condition reads

$$\sum_{f \in \mathcal{SF}(p)} A_{pf} \left\{ - \left[\sum_{k=1}^{m} \lambda_k \left(\mathbf{V}_{k+1} - \mathbf{V}_k\right)\right]_{l,r} + \mathbf{G}_{\mathbf{n}_{pf},r} - \mathbf{G}_{\mathbf{n}_{pf},l} \right\} = \mathbf{0}, \tag{4.2.5}$$

and in the case of Lagrangian gas dynamics, the term in the braces yields

$$-\left[\sum_{k=1}^{m}\lambda_k\left(\mathbf{V}_{k+1}-\mathbf{V}_k\right)\right]_{l,r}+\mathbf{G}_{\mathbf{n}_{pf},r}-\mathbf{G}_{\mathbf{n}_{pf},l}=\lambda_l(\mathbf{V}_l^\star-\mathbf{V}_l)-\lambda_r(\mathbf{V}_r-\mathbf{V}_r^\star)+\mathbf{G}_{\mathbf{n},r}-\mathbf{G}_{\mathbf{n},l}, \tag{4.2.6}$$

Replacing the components of the Lagrangian variables in the above equation and algebraic manipulations, we arrive at the conclusion that the Finite Volume scheme is conservative provided that the following node-based condition is satisfied

$$\sum_{f\in\mathcal{SF}(p)}A_{pf}(\overline{p}_{rf}-\overline{p}_{lf})\begin{pmatrix}0\\1\\\mathbf{0}\\v^\star_{\mathbf{n}_{pf}}\end{pmatrix}=\mathbf{0}, \tag{4.2.7}$$

with $\overline{p}_{rf},\overline{p}_{lf}$ the right and left intermediate pressure respectively. We assume that the parameter $v^\star_{\mathbf{n}_{pf}}$ is the projection of the unknown nodal vector $\mathbf{v}_p$ onto the unit normal $\mathbf{n}_{pf}$, that is

$$v^\star_{\mathbf{n}_{pf}}=\mathbf{v}_p\cdot\mathbf{n}_{pf}. \tag{4.2.8}$$

With this fundamental assumption, the vectorial $\mathbf{v}_p$, is now an unknown that can be interpreted as an approximation of the nodal velocity. The node-based conservation can be express in terms of the expression of the interface pressure and yields

$$\sum_{f\in\mathcal{SF}(p)}A_{pf}(\lambda_{lf}+\lambda_{rf})(\mathbf{v}_p\cdot\mathbf{n}_{pf}-\overline{v}_{\mathbf{n}_{pf}})\mathbf{n}_{pf}=\mathbf{0},$$

with

$$\overline{v}_{\mathbf{n}_{pf}}=\frac{\lambda_{lf}v_{\mathbf{n}_{pf},l}+\lambda_{rf}v_{\mathbf{n}_{pf},r}}{\lambda_{lf}+\lambda_{rf}}-\frac{p_{rf}-p_{lf}}{\lambda_{lf}+\lambda_{rf}}.$$

Finally, the node-based conservation condition (4.1.23) boils down to the Lagrangian nodal solver which is a linear system

$$\sum_{f\in\mathcal{SF}(p)}A_{pf}(\lambda_{lf}+\lambda_{rf})(\mathbf{n}_{pf}\otimes\mathbf{n}_{pf})\mathbf{v}_p=\sum_{f\in\mathcal{SF}(p)}A_{pf}(\lambda_{lf}+\lambda_{lf})\overline{v}_{\mathbf{n}_{pcf}}\mathbf{n}_{pf}. \tag{4.2.9}$$

This system always admits a unique solution which provides an approximation of the nodal velocity $\mathbf{v}_p$.

4.2.3 Parallel implementation: PETSc

For the application of the 3D numerical schemes, the computational power needed to solve any problem is considerably larger than two-dimensional cases. This is due to the high memory consumption and long computational time. These problems can be somewhat overcome with the parallelization of the numerical scheme by splitting the global problem into smaller local ones that will run on different processors. The implementation of the

parallelization includes handling the distribution of the memory and the communication among processors to solve the global problem. In this present work, we make use of the Portable, Extensible Toolkit for Scientific Computation (PETSc) library (refer to Balay *et al.* (2021*b*,*a*, 1997)). This section focuses on the interaction between our application and the supporting libraries from PETSc to provide the infrastructure for an efficient data management.

4.2.3.a PETSc library

PETSc is a suite of data structures and routines that provide frames to develop application codes on both serial and parallel computations. It can be used in application codes written in many languages, and in our case, it is written in Fortran. PETSc library is organized hierarchically, allowing users to employ the appropriate level of abstraction for a particular problem. Figure 4.21(a) shows the available numerical libraries in PETSc taken from Balay *et al.* (2021*a*) and Figure 4.21(b) shows the libraries used in our application code.

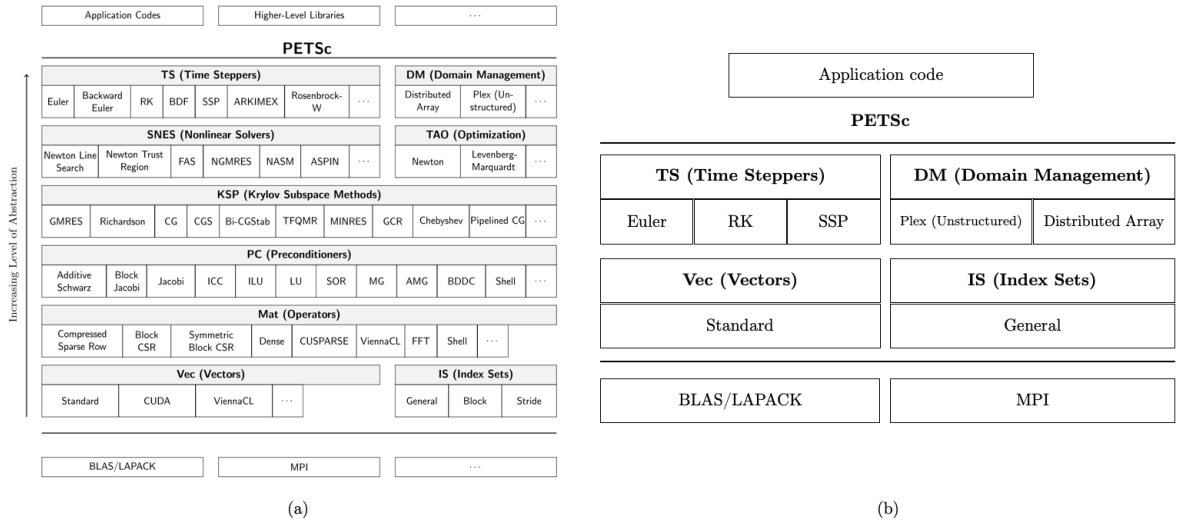


Figure 4.21: (a) Organization of numerical libraries available in PETSc Balay *et al.* (2021*a*), (b) PETSc numerical libraries used in our application code.

The object-oriented style of PETSc is managed through handles to opaque data structures which are created, accessed and destroyed by calling appropriate library routines. The variety of components allows to manipulate a particular family of objects and provide functionality required for many parallel solutions. For instance, the `Vec` component provides the vector operations required for setting up and solving large-scale linear and nonlinear problems, including easy-to-use parallel scatter and gather operations, as well as special purpose code for handling ghost points for regular data structures; the `TS` component consists of code for the time evolution of solutions of PDEs and also provides pseudo-transient continuation techniques for computing steady-state solutions.

In what follows, we present the different PETSc objects and libraries implemented

to handle unstructured mesh, time integration, Finite Volume discretization and parallel computing.

Unstructured mesh with PETSc DMPlex Data management, DM in short is the abstract PETSc object that manages an abstract grid object and its interactions with algebraic solvers such as vectors, matrices and solvers. "Plex" is a subclass of DM in PETSc that encapsulates an unstructured mesh, see Lange *et al.* (2015). The DMPlex object allows to:

- handle unstructured grids using the generic DM interface;
- handle interface with the PETSc FV trial discretization objects;
- provide parallel redistribution for load balancing.

The main advantage of DMPlex in representing topology is that all the different entities and connectivity of a mesh, including cells, faces, edges and vertices, are handled exactly the same way. All pieces of the mesh are treated as points. A mesh is built by relating points to other points. For example, an edge is defined by being covered by two vertices, and a triangle face can be defined by being covered by three edges, or even by three vertices. In order to make use of this object, the mesh pieces have to be numbered consecutively. The PETSc convention in 3D is to number first the cells, second the faces, third the edges and lastly the vertices.

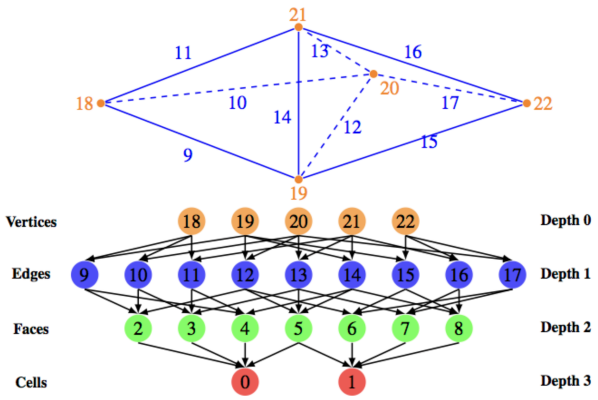


Figure 4.22: The Hasse diagram for two tetrahedra in a 3D mesh Knepley (1997).

DMPlex's internal representation of mesh topology provides an abstraction layer that decouples the mesh from the underlying file format, allowing support for multiple mesh file formats to be added generically. In this present work, the input meshes read by DMPlex are in the Gmsh format. As for the output, DMPlex provides routines that generate solution output in HDF5 and vtk formats compatible with Paraview.

PETSc time integrator The Time Integrator object in PETSc, TS in short, manage all time-steppers for solving time-dependent PDEs by providing Ordinary Differential

Equations (ODE) integrators. The `TSSetType` library allows to set up the actual time-stepping integrator required. The time-stepping methods that we are interested are :

1. `TSEULER` for the Euler forward method ;
2. `TSSSPRKS2` for the second-order Runge-Kutta method.

The `TSSetDM` then allows to connect the time and space discretization.

Finite Volume with PETScFV The `PETScFV` is a PETSc object that manages the Finite Volume discretization. We start by creating an empty PETScFV object, followed by setting the type of object using `PetscFVSetType`. The type that we used is obviously the `fvm` type for the Finite Volume method.

Parallel simulation The parallel message-passing communication in PETSc is done through the standard Message-passing Interface (MPI), providing low-level routines to exchange data primitives between processors. The main advantages are that, we can call MPI directly if needed and the same code can be used for both sequential and parallel runs. The DMPlex library provides parallel data gathering through automated parallel distribution of the DMPlex and the pre-allocation of parallel matrix and vector data structured. Mesh partitioning is provided via internal interfaces to several partitioner libraries, *i.e* Metis and data migration is based on PETSc's internal Star Forest communication abstraction (`PetscSF`). PETSc built-in communicators, namely `MPI_COMM` are objects allowing to define process group and synchronization channel.

4.2.4 Three-dimensional numerical validation

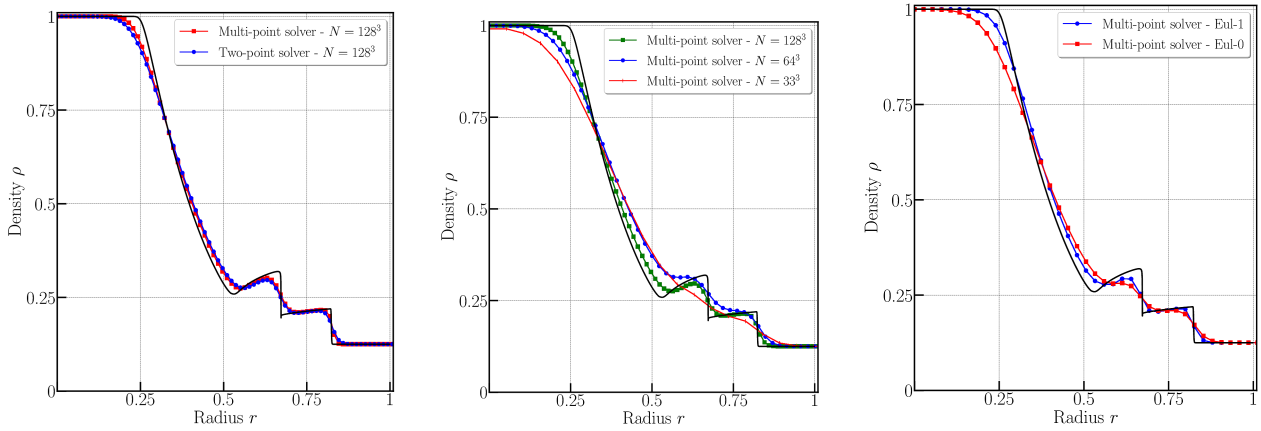
In this section, we run a set of representative test cases to demonstrate the performance of the three-dimensional extension. These test cases are ran on various types of grids that can be made of tetrahedral, hexahedral and prism cells. The CFL number is set to 0.5 for all cases. Similarly to the two-dimensional cases, we make use of Gmsh to generate three-dimensional meshes.

3D spherical Sod shock tube Here, we consider a spherical explosion problem in 3D. This test case allows to assess the ability of the schemes to capture simple but non-aligned spherical waves without producing spurious oscillations. The computational setup represents a spherical extension of the classical Sod problem with a domain $(x, y, z) \in [0 : 1.2]^3$ with a discontinuity at $r = \sqrt{x^2 + y^2 + z^2} = 0.5$, and separating the initial states:

$$(\rho, \mathbf{v}, p)_l = (1, 0, 1) \quad , \quad (\rho, \mathbf{v}, p)_r = (0.125, 0, 0.1) \quad , \quad t_{final} = 0.2,$$

with symmetry boundary conditions on $x = 0$ and $y = 0$, and outflow boundary conditions otherwise. The numerical densities are compared with a reference solution represented with the black line in figure 4.23. Figure (a) shows the comparison between the two-point and multi-point solvers on $N = 128^3$ hexahedral cells where the numerical density

of two-point solver is represented with red square and multi-point one with blue circles. Figure (b) shows the grid convergence for the multi-point solver with 32^3 cells represented with green diamonds, 64^3 cells with red squares and 128^3 cells with blue circle. Lastly, in figure (c) we compare the numerical density obtained from both solvers at second-order and once again the numerical density of two-point solver are represented with red squares and multi-point are with blue circles

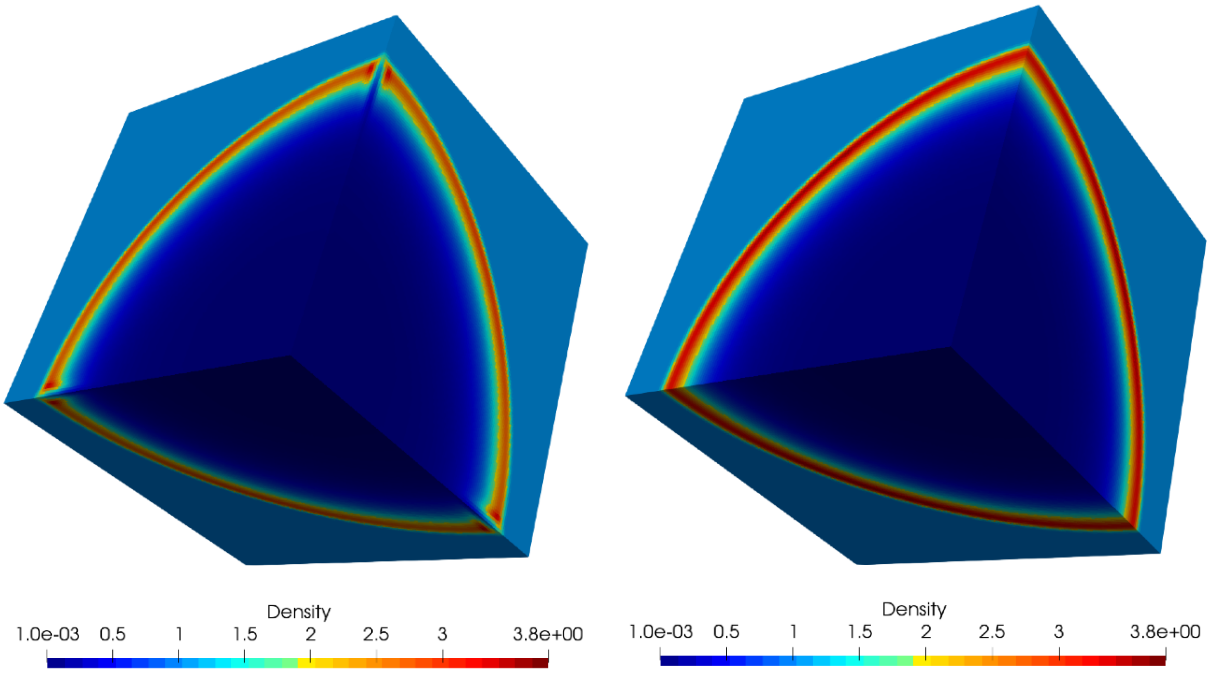


(a) Scheme comparison. (b) Grid convergence. (c) Second-order scheme.

Figure 4.23: Spherical Sod shock tube —Comparison of numerical density with reference solution (black line) at $t_{final} = 0.2$

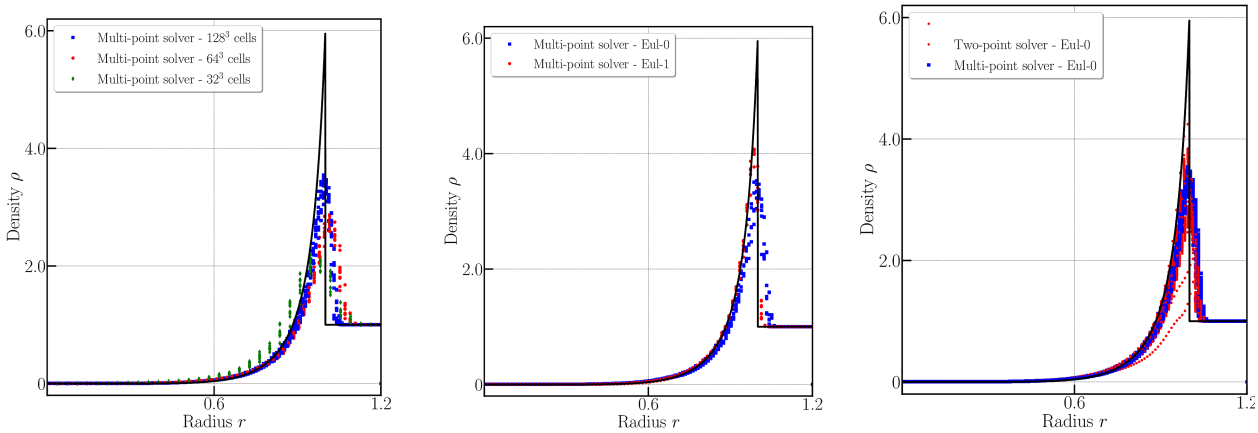
Sedov blast wave problem The Sedov problem Sedov (1959) is a spherical symmetric explosion. The domain is given by $(x, y, z) \in [0 : 1.2]^3$ initially filled with perfect gas at rest such as $(\rho^0, u^0, v^0, p^0, \gamma) = (1, 0, 0, 1.10^{-6}, 1.4)$. A total energy of $E_{total} = 0.851072$ is concentrated at the cells touching the origin. Symmetry boundary conditions are applied on all boundaries. This configuration corresponds to spherical symmetric explosion at which the shock front reaches $r = \sqrt{x^2 + y^2 + z^2} = 1$ at $t_{final} = 1$ with a density peak of $\rho = 6$. Figure 4.24 illustrates the numerical density for the Sedov blast wave problem with 128^3 hexahedral cells with (a) is the result of the two-point scheme and (b) the results of the multi-point scheme. In figure 4.25(a) is a scatter plot of both the numerical schemes where we once again observe a nonsymmetrical solution from the two-point solver. Figure 4.25(b) illustrates the mesh convergence of the multi-point scheme on $32^3, 64^3$ and 128^3 cells. Finally, figure 4.25(c) compares the solution of the first and second order multi-point scheme where we observe an improvement in capturing the numerical density with the second order scheme.

Hypersonic flow over a sphere The computational domain for a hypersonic flow over a sphere consists of an inner sphere of radius $R = 0.5$ at which no-slip wall boundary conditions are applied and, an outer sphere of radius $R = 1$, at the left of which a uniform flow of $Ma = 20$ is imposed. The right side of the domain is a outflow boundary condition. We run the test case on 410000 hexahedral elements and refined at the wall



(a) Two-point scheme. (b) Multi-point scheme.

Figure 4.24: Sedov blast wave problem — Numerical density at $t_{final} = 1.0$.



(a) Scheme comparison. (b) Grid convergence. (c) Second-order scheme.

Figure 4.25: Sedov blast wave problem — Comparison of numerical density with the exact solution (black line) at $t_{final} = 1.0$.

to better capture the solution, as illustrated in 4.26(a). The comparison of the numerical density for both the two-point and multi-point solver is illustrated in figure 4.26(b) where the top half (resp. bottom half) represents the solution of the two-point solver (resp. multi-point solver). A cut through the numerical solution in the place $z = 0$ is drawn figure 4.27(a), and we compare the numerical density between both solvers. The two-point scheme shows a jump in density though the shock whereas the multi-point scheme

shows a symmetrical revolution of density. Figure 4.27(b) then presents a comparison of the pressure coefficient along the wall of the sphere computed with both schemes with the modified Newtonian Law represented with the black line. Other than a slight oscillation of pressure coefficient around the stagnation point for the two-point, the solutions are coherent with the modified Newtonian law for angles below 60°.

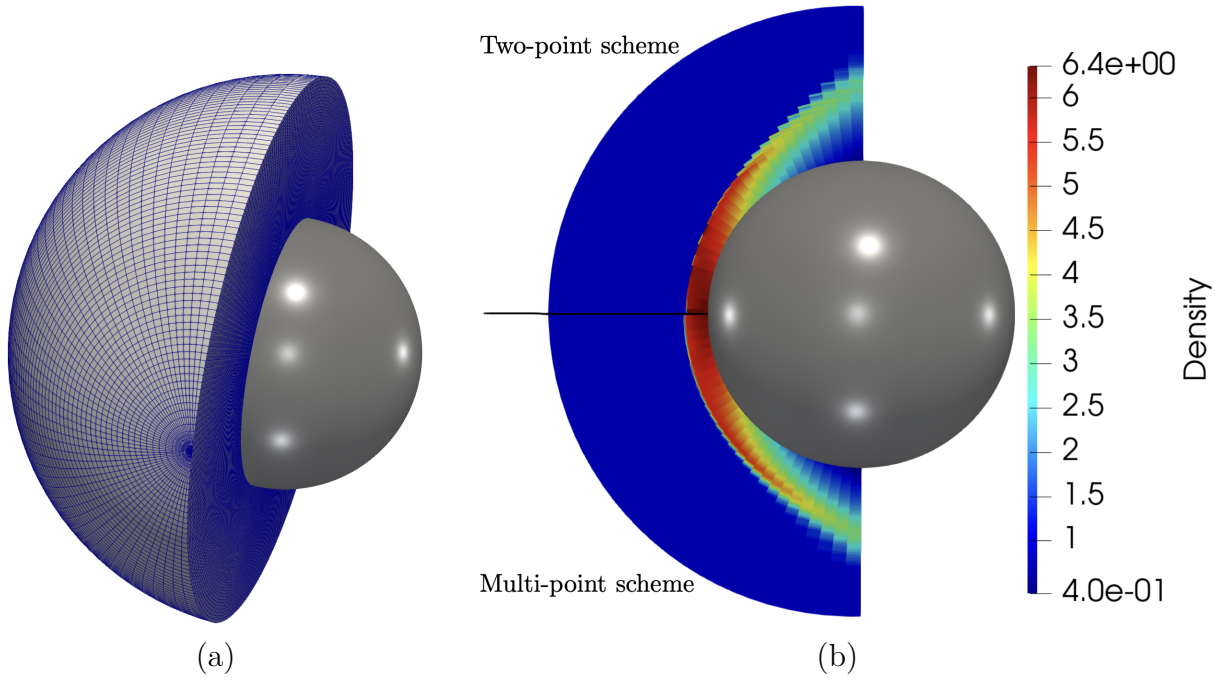


Figure 4.26: Hypersonic flow over a sphere —(a) Mesh of computational domain — (b) Comparison of the numerical density for the two-point (top half) and multi-point scheme (bottom half).

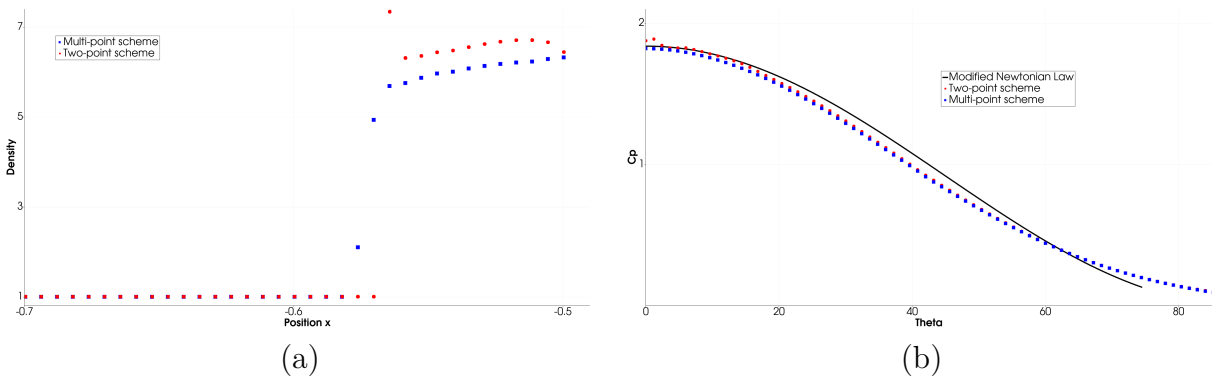
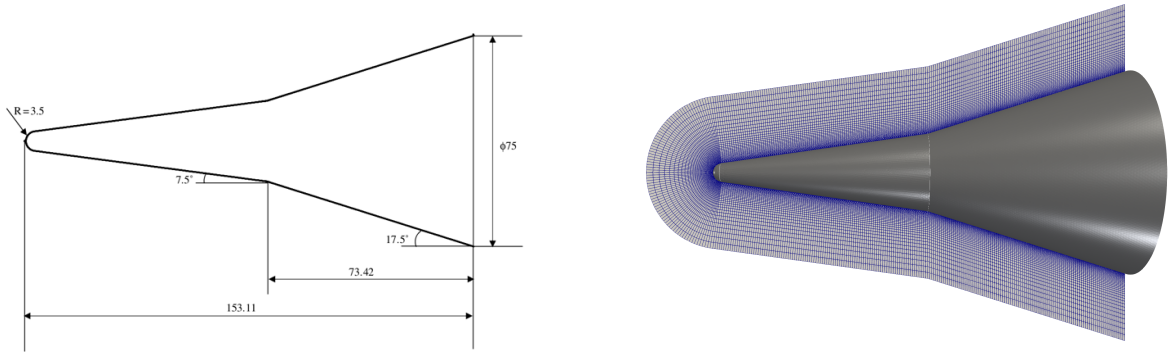


Figure 4.27: Hypersonic flow over a sphere — Comparison plot of numerical solution between two-point and multi-point schemes — (a) Density plot along x-axis — (b) Comparison of pressure coefficient along the wall of the sphere compared with the modified Newtonian Law (black line).

Blunt cone-flare We run the blunt cone-flare test case from Savino & Paterna (2005). The free stream conditions correspond to the exit nozzle conditions of the $H3$ wind

tunnel at Von Karman Institute (VKI) Dieudonne *et al.* (1999), such as $Ma = 6, T_\infty = 67.07K, p_\infty = 673.67Pa$. Figure 4.28(a) illustrates the geometry extracted from Savino & Paterna (2005) and figure 4.28(b) shows the computational mesh made of hexahedra of the domain. In order to validate the 3D scheme in the first place, we compare the numerical solution with a computed 2D axisymmetric solution. The 2D axisymmetric solution is obtained from a quick additional extension implemented in the simulation code with a classical 2D axisymmetric scheme, refer to the appendix 1.2. Figure 4.29(a) compares the density profile over the blunted cone flare of the 3D multi-point scheme on the top half and the 2D axisymmetric solution on the bottom half. In figure 4.29(b), the pressure on the surface of the cone flare object is extracted and we compare once again the numerical results using the 3D multi-point scheme with 2D axisymmetric solution represented with the red circles at $t_{final} = 3$. The 3D scheme shows good coherence with the axisymmetric solution. Lastly, the pressure (log scale) and Mach number profile computed using the 3D multi-point scheme is illustrated in figure 4.30. We observe good coherence for both the pressure and Mach profile compared to the results from Savino & Paterna (2005).

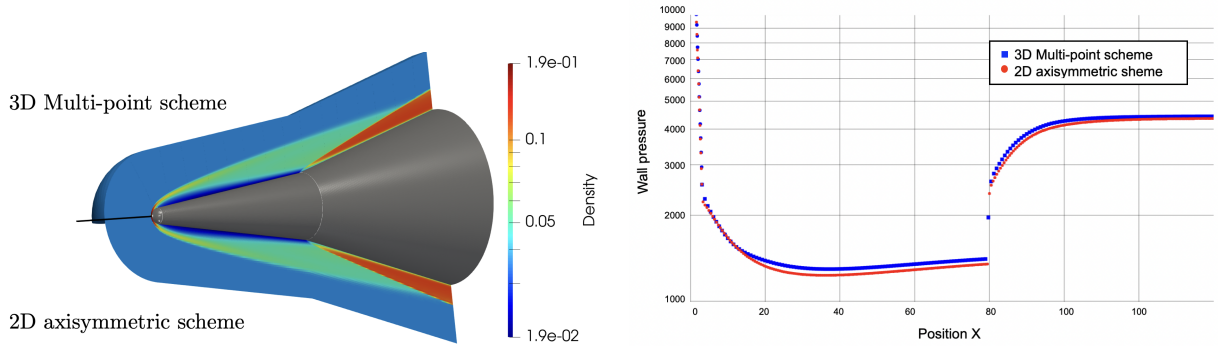


(a) Geometry. (b) Domain mesh.

Figure 4.28: Blunted cone flare — Computational domain.

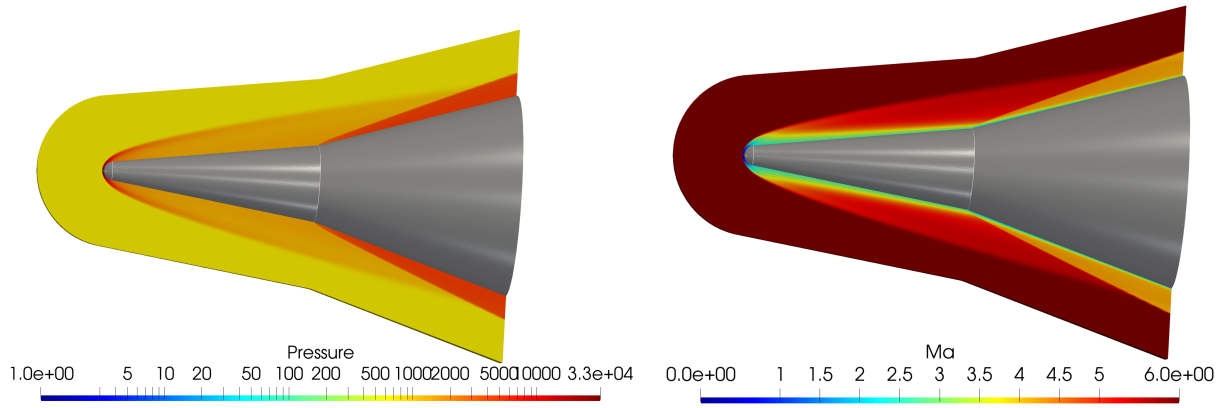
Manoeuvrable Re-entry Vehicle (MaRV) The Manoeuvrable Re-entry Vehicle (MaRV) is a type of ballistic missile capable of altering its trajectory aerodynamically after boost, typically after re-entry. Moreover, it may be capable of significant glide phases and maneuvers, usually at hypersonic speeds. Here, we run this test case using the geometries for the MaRV from Petsopoulos & Regan (1996). The free stream conditions of this computation fixed at $Ma = 5, T_\infty = 223.15K$ at an altitude of $10km$ where $P_\infty = 26500Pa, \rho_\infty = 0.4$ and $a_\infty = 300m/s$. Figure 4.31(a) shows the geometry of the MaRV object and figure 4.31(b) the mesh of the computational domain consists of 24 million tetrahedral cells. With this configuration, we ran the simulation with 3 different angles of incidence of $0°, 5°$ and $10°$ respectively and the numerical density profile along the object (yz-plane) and behind the object (xy-plane) are illustrated in figure 4.32.

We then run a simulation with the $N = 31$ million cells where the mesh are being refined at the back of MaRV such as illustrated in figure 4.33(a). We run the computation



(a) (b)

Figure 4.29: Blunted cone flare — (a) Comparison of density profile between the 3D multi-point scheme (top half) and 2D axisymmetric scheme (bottom half. — (b) Comparison of the pressure along the wall of the blunted cone-flare computed with the 3D multi-point scheme and 2D axisymmetric solution (red circle).



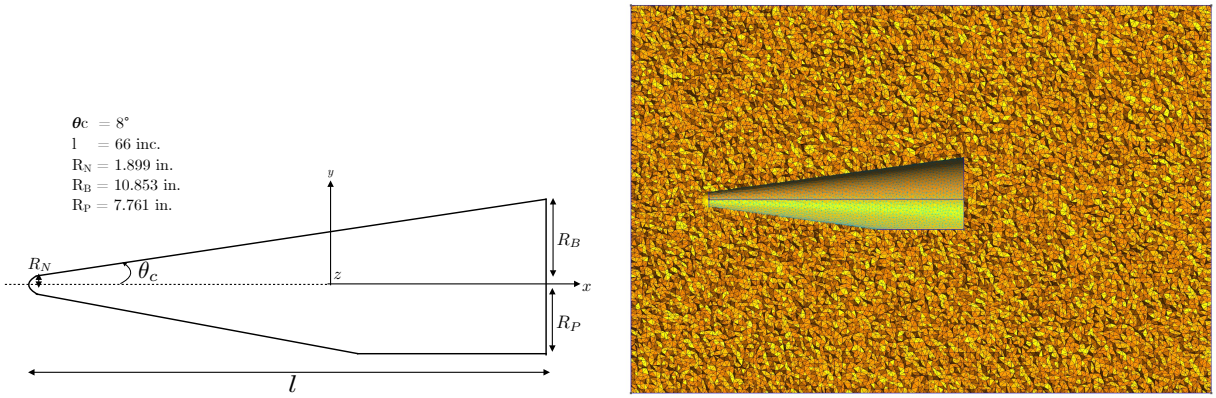
(a) Numerical pressure. (b) Mach number.

Figure 4.30: Blunted cone flare — Profiles of numerical pressure and the Mach number computed with the 3D multi-point scheme.

with an angle of incidence of 10° to observe the vorticity generated by the forebody of the object, see figure 4.33(b).

Atmospheric Reentry Demonstrator (ARD) The Atmospheric reentry demonstrator (ARD) is a test vehicle towards developing space transportation vehicles that can return to Earth to carry payloads or even human. Validation of theoretical aerodynamics prediction is one of its main objectives. This demonstrator is a 3-axis stabilized automatic capsule and has an axisymmetric shape composed of a spherical capsule followed by a tore and a conical surface. Here, we obtained the geometry from Annaloro *et al.* (2017). The free stream conditions used for the computations are of $Ma = 24, T_\infty = 224.5K$ at an altitude of $65.83km$ where $P_\infty = 10.23Pa, \rho_\infty = 1.586E10^{-4}kg/m^3$ and $u_\infty = 7212.43m/s$.

We run a computation with the Eul-1 multi-point scheme with an angle of incidence of 20° and no side slip angle on $N = 6.8$ millions cells. Figure 4.34(b) illustrates the



(a) Geometry. (b) Domain mesh.

Figure 4.31: MaRV — Computational domain and geometry.

numerical pressure profile of the object. In figure 4.35(b), we present the pressure coefficient, C_p distribution on the ARD. The result shows good coherence with the solution presented by Annaloro *et al.* (2017). We then compare the pressure coefficient with the flight data from Walpot (2002) along the symmetry plane ($y = 0m$) of the ARD and the results are presented in figure 4.35(b).

Pre-X The Pre-X hypersonic vehicle was developed in order to test thermal protection systems (TPS) for reusable launch vehicles, and, at the same time gather aerothermodynamic data in the hypersonic regime to improve process of in flight measuring. Here, we once again obtained the geometry from Annaloro *et al.* (2017). The free stream conditions used for the computations are of $Ma = 25, T_\infty = 207K$ at an altitude of $73.6km$ where $P_\infty = 3.11Pa, \rho_\infty = 5.5546.10^{-5}kg/m^3$ and $u_\infty = 7205m/s$.

We run a computation with the Eul-1 multi-point scheme with the perfect gas EOS and an angle of incidence of 40° and no side slip angle on $N = 6.4$ millions cells, the mesh of the domain is illustrated in figure 4.36. Figure 4.37(a) illustrates the numerical pressure profile on the flow around the object. In figure 4.37(b), the pressure coefficient C_p distribution on the Pre-X of illustrated. We observe similar C_p distribution as the one in Figure 15 by Annaloro *et al.* (2017). We then compare the pressure coefficient with the CFD computation from MISTRAL, a high-fidelity code (see Annaloro *et al.* (2017)). The comparison is done along two different plans $y = 0m$ and $y = 0.3m$. The results show good coherence as presented in figure 4.38 in general. The slight difference between the numerical solutions is due to the fact that the MISTRAL code is ran on the Navier-Stokes equations.

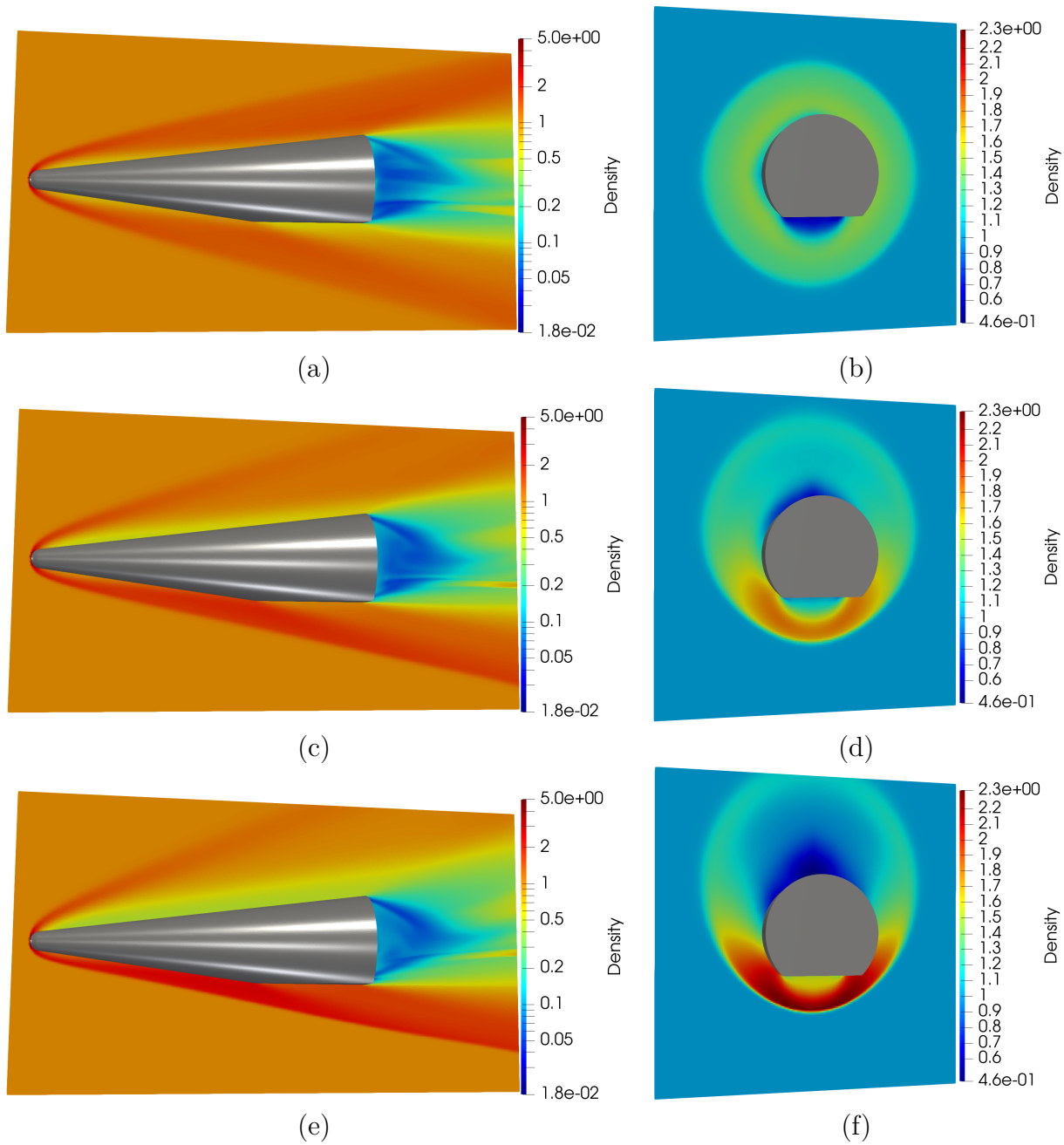
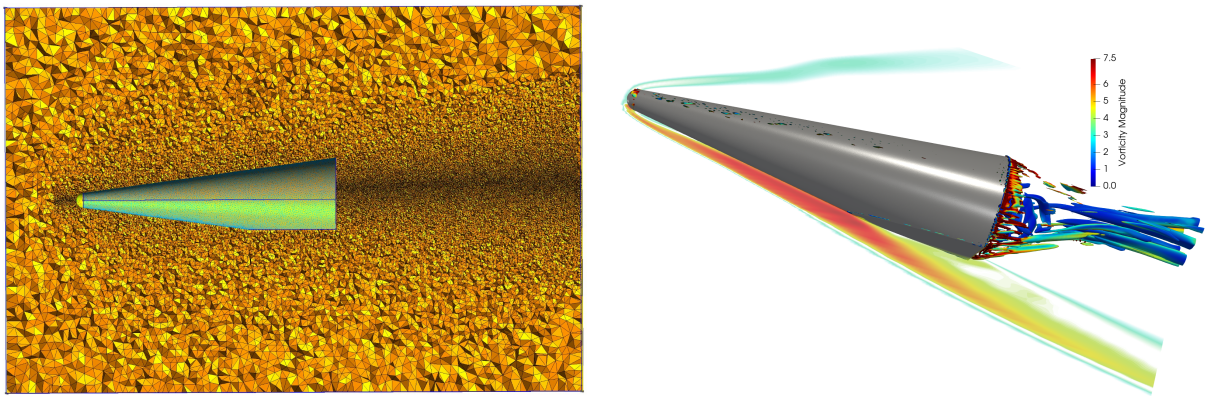


Figure 4.32: MaRV — Numerical density profile at the yz-plane (left) and xy-plane (right) — (a-b) 0° angle of incidence, (c-d) 5° angle of incidence and (e-f) 10° angle of incidence.

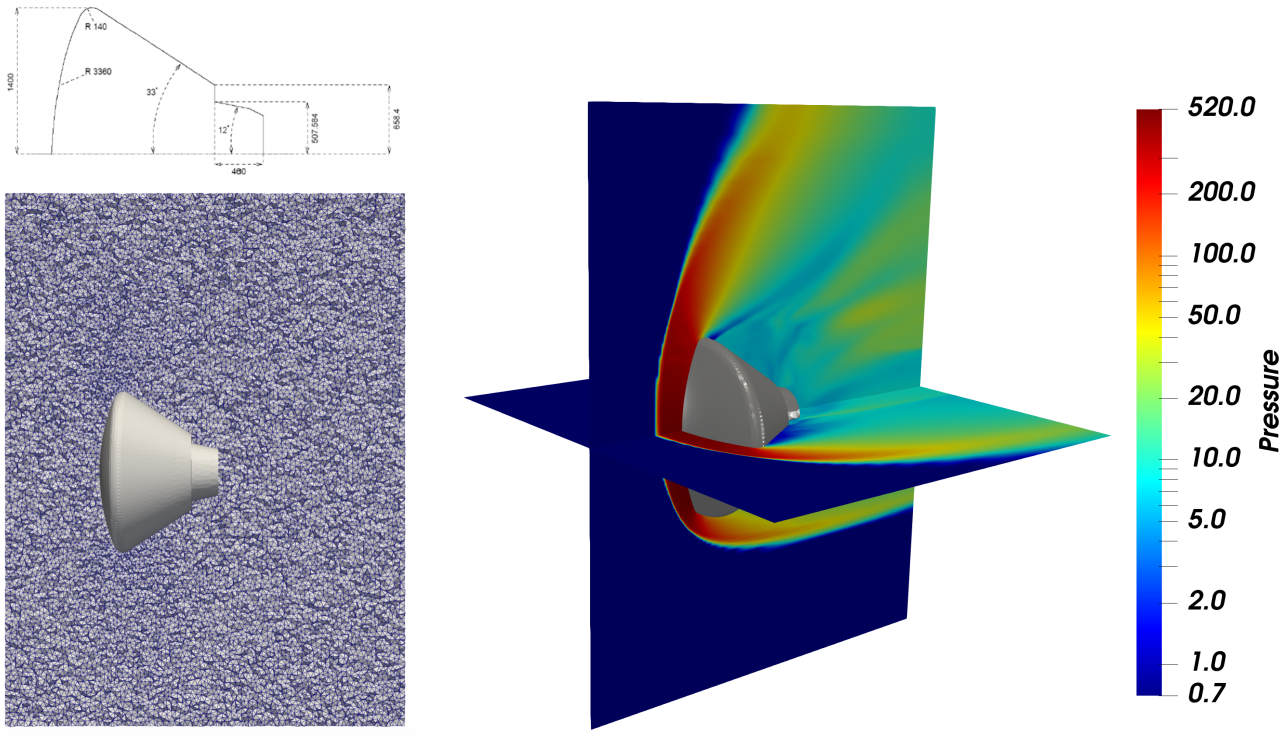
4.3 Implicit time-stepping scheme

In this section, the implicit method, also known as the backward Euler method is proposed with the aim of studying steady inviscid flows employing a time-marching procedure, see



(a) Refined mesh. (b) Vorticity.

Figure 4.33: MaRV — Refined mesh and the streamline around MaRV.

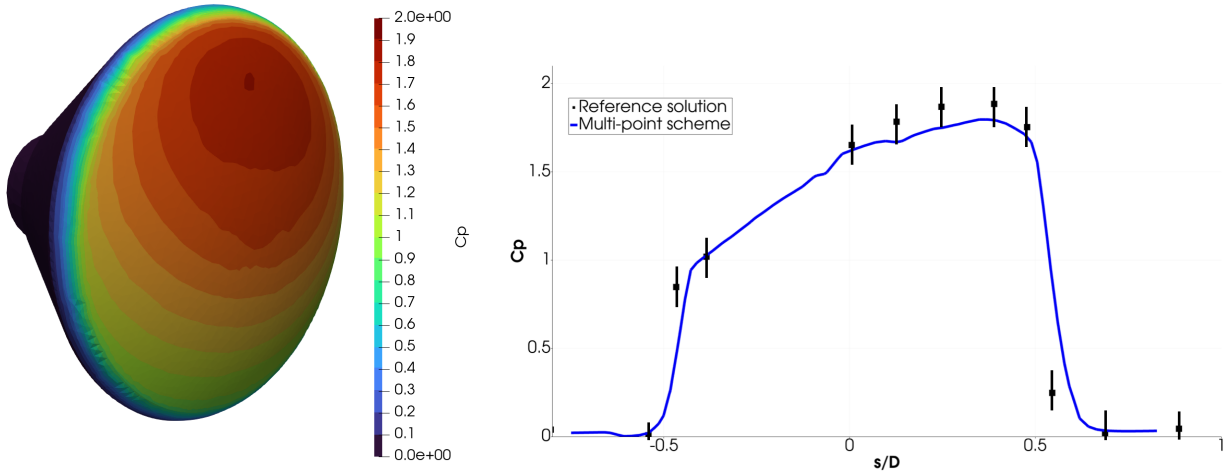


(a) Geometry and domain mesh. (b) Numerical pressure.

Figure 4.34: ARD — Geometry of the object from Annaloro *et al.* (2017) and the numerical pressure profile with the Eul-1 multi-point scheme.

Yee *et al.* (1982). Recalling the equation of the subface-based discretization

$$\mathbf{U}_c^{n+1} - \mathbf{U}_c^n + \frac{\Delta t}{|\omega_c|} \sum_{p \in \mathcal{P}(c)} \sum_{\mathcal{SF}(pc)} l_{pcf} \overline{\mathbf{F}}_{pcf} = \mathbf{0},$$



(a) C_p distribution. (b) C_p comparison.

Figure 4.35: ARD — Pressure coefficient distribution and the comparison with flight data (black squares) Annaloro *et al.* (2017).

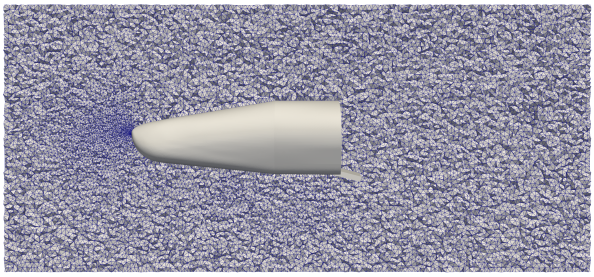


Figure 4.36: Pre-X — Domain mesh.

we start by defining the non-linear function of the residual of cell c such as

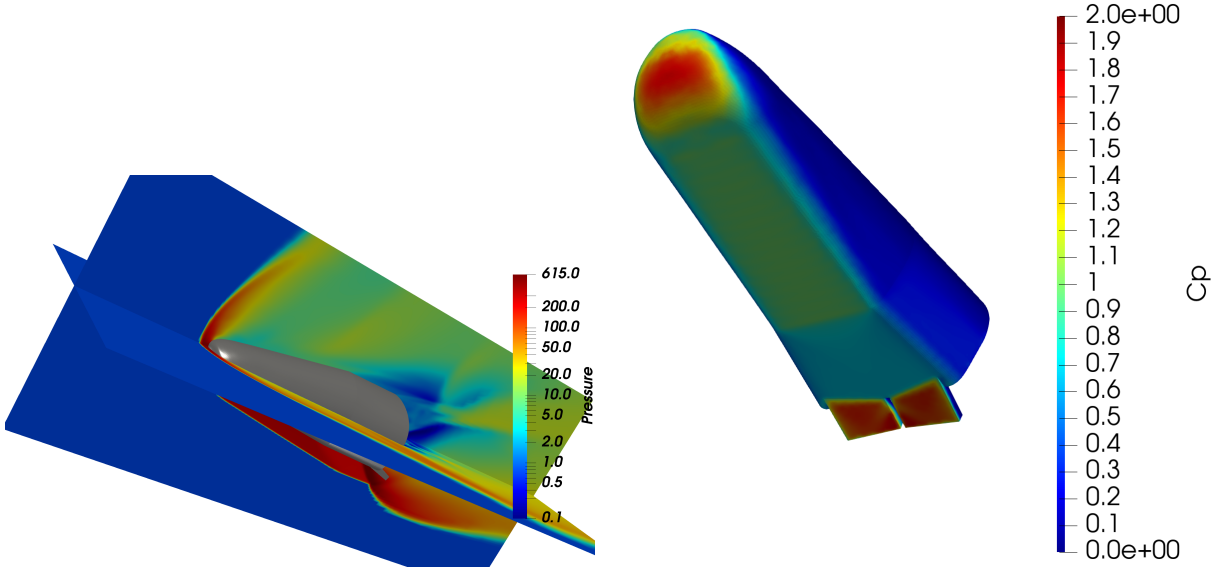
$$\mathbf{R}_c(\mathbf{U}) = \sum_{p\in\mathcal{P}(c)} \sum_{f\in\mathcal{SF}(pc)} l_{pcf}\overline{\mathbf{F}}_{pcf}. \tag{4.3.1}$$

with $\mathbf{R}_c = \mathbf{R}_c(\mathbf{U}_c, \mathbf{U}_{d(c,f)}, \mathbf{v}_p) \in \mathbb{R}^{d+2}$. We also introduce

$$\delta\mathbf{U}_c = \mathbf{U}_c^{n+1} - \mathbf{U}_c^n, \tag{4.3.2}$$

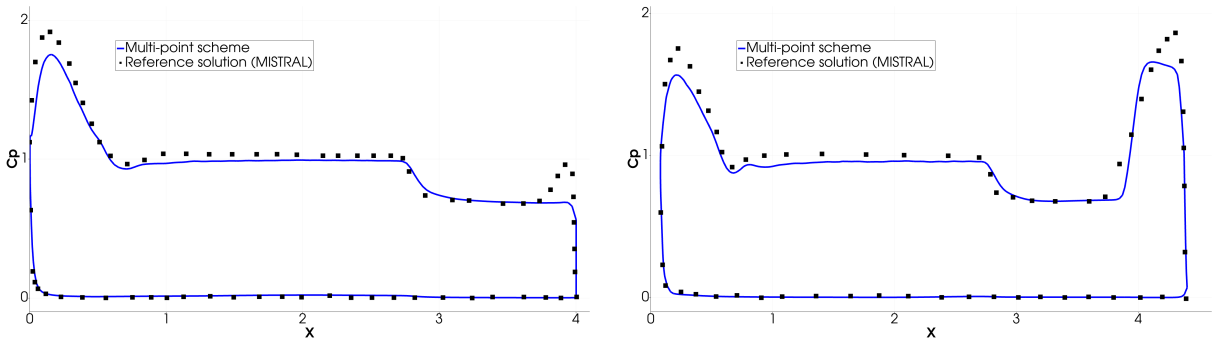
where $\delta\mathbf{U}_c \in \mathbb{R}^{d+2}$ is the unknown and with this, we have $\mathbf{U}_c^{n+1} = \mathbf{U}_c^n + \delta\mathbf{U}_c$. The simple backward Euler time implicit scheme applied to equation (3.2.4) and using this notation yields

$$\delta\mathbf{U}_c - \frac{\Delta t}{|\omega_c|}\mathbf{R}_c(\mathbf{U}^{n+1}) = \mathbf{0}. \tag{4.3.3}$$



(a) Pressure profile. (b) Cp.

Figure 4.37: Pre-X — Numerical pressure profile and the pressure coefficient distribution on Pre-X.



(a) C_p at $y = 0$. (b) C_p at $y = 0.3m$.

Figure 4.38: Pre-X — Comparison of pressure coefficient on Pre-X between the multi-point scheme and MISTRAL code.

This is an implicit non-linear system at time t^{n+1}. For clarity in the following section, we re-write the subface-based discretization equation in the following form

$$\frac{|\omega_c|}{\Delta t}\delta\mathbf{U}_c = -\mathbf{R}_c(\mathbf{U}^{n+1}). \tag{4.3.4}$$

We have to find an expression of the residual on the right-hand side, $\mathbf{R}_c$ at time t^{n+1}. A solution of this time-implicit problem is to formulate a Newton method for solving the non-linear problem from t^n to t^{n+1}. Let us define q as the Newton or nonlinear iteration counter index, such that

$$\mathbf{U}_c^{q+1} = \mathbf{U}_c^q + \delta\mathbf{U}_c^q, \tag{4.3.5}$$

and, for $q = 0$, we set $\mathbf{U}_c^0 = \mathbf{U}_c^n$. The traditional Newton's method can be written as:

$$\frac{|\omega_c|}{\Delta t}(\mathbf{U}_c^q + \delta\mathbf{U}_c^q - \mathbf{U}_c^n) = -\mathbf{R}_c(\mathbf{U}^{q+1}). \tag{4.3.6}$$

The solution of (4.3.6) requires the evaluation of the residual at the new iteration, that is, $\mathbf{R}_c(\mathbf{U}^{q+1})$. Since $\mathbf{U}_c^{q+1}$ is not known, the solution cannot be found directly. However, the residual $\mathbf{R}_c(\mathbf{U}^{q+1})$ can be linearized at the current iteration with a truncated Taylor expansion, giving

$$\mathbf{R}_c(\mathbf{U}^{q+1}) \approx \mathbf{R}_c(\mathbf{U}^q) + \left(\frac{\partial \mathbf{R}_c}{\partial \mathbf{U}_c}\bigg|_{\mathbf{U}^q}\right)\delta\mathbf{U}_c^q, \tag{4.3.7}$$

with $\dfrac{\partial \mathbf{R}_c}{\partial \mathbf{U}} \in \mathbb{R}^{d+2} \times \mathbb{R}^{d+2}$ the Jacobian of the numerical flux. Substituting the linearization (4.3.7) into (4.3.6), we obtain the following implicit scheme:

$$\left[\frac{|\omega_c|}{\Delta t} + \frac{\partial \mathbf{R}_c(\mathbf{U}^q)}{\partial \mathbf{U}_c}\right]\delta\mathbf{U}_c^q = -\frac{|\omega_c|}{\Delta t}(\mathbf{U}_c^q - \mathbf{U}_c^n) + \mathbf{R}_c(\mathbf{U}^q). \tag{4.3.8}$$

In practice, the Newton method is known to converge quadratically and generally requires a small number of nonlinear iterations. However, each nonlinear iteration requires to solve a large matrix system which is a process that can be prohibitively expensive- *n.b.* the matrix is a sparse matrix with real coefficients. In order to solve (4.3.8) for steady state, we want to find $\delta\mathbf{U} \to 0$. This can be achieved by considering the first iteration of the Newton algorithm which leads to the classical linearized implicit scheme:

$$\left[\frac{|\omega_c|}{\Delta t} + \frac{\partial \mathbf{R}_c(\mathbf{U}^n)}{\partial \mathbf{U}_c}\right]\delta\mathbf{U}_c = -\mathbf{R}_c(\mathbf{U}^n). \tag{4.3.9}$$

The term in square brackets on the left-hand side of equation (4.3.8) is referred to as the implicit operator. The implicit operator constitutes a large, sparse and non-symmetric block matrix with dimensions equal to the total number of cells. The matrix consists of the flux Jacobian $\dfrac{\partial \mathbf{R}_c}{\partial \mathbf{U}_c}$ evaluated at t^n, augmented by a diagonal term $\dfrac{\omega_c}{\Delta t}\mathbb{I}_d$. The solution of this system (4.3.9) requires the inversion of the implicit operator. This can be done using iterative methods that will be presented in section 4.3.2. In the following, we discuss the matrix form of the implicit operator.

4.3.1 Form of the matrix operator

Referring to (4.3.7), the linearization of the residual $\mathbf{R}^{n+1}$ can be written in the form

$$\mathbf{R}_c^{n+1} \approx \mathbf{R}_c^n + \sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)}\frac{\partial}{\partial \mathbf{U}}\left(l_{pcf}\overline{\mathbf{F}}_{pcf}\right)\delta\mathbf{U}_c + \mathcal{O}(\Delta t^2), \tag{4.3.10}$$

with $f \in \mathcal{SF}(pc)$ the subface f attached to the corner pc of the control volume ω_c. The flux Jacobian has to be conceived as an operator which acts on the update $\delta\mathbf{U}$. With the development from equation (4.3.10), equation (4.3.9) now can be written as:

$$\left(\frac{\mathbb{M}}{\Delta t} + \mathbb{E}\right)\delta\mathbf{U} = -\mathbf{R}_c(\mathbf{U}^n), \tag{4.3.11}$$

where $\mathbb{M}$ is the diagonal block-matrix containing the volumes of the cells defined by $\mathbb{M}_c = |\omega_c|\mathbb{I}$, $\mathbb{E}$ is the block-matrix containing the Jacobians of the numerical flux computed at t^n and $\mathbf{R}^n$ is the global block-vector containing the residuals.

In practice, we look for an approximate version of the Jacobian matrix to solve (4.3.11) to steady state. A simple way to approximate the Jacobian of the residuals is to use the simple formulation of the flux proposed by Rusanov (see *e.g* Toro (1999)) independently of the actual Riemann solver used in the numerical scheme. Recalling the definition of the Rusanov flux:

$$\overline{\mathbf{F}}_{Rusanov} = \frac{1}{2}(\mathbf{F}_l + \mathbf{F}_r) + \frac{|\Lambda_k|}{2}(\mathbf{U}_l - \mathbf{U}_r), \tag{4.3.12}$$

where $|\Lambda_k|$ is the absolute value of the largest eigenvalue of the linearised problem at the interface between the left and right states. The same strategy is implemented for the multi-point scheme. We then use a truncated Taylor expansion to write

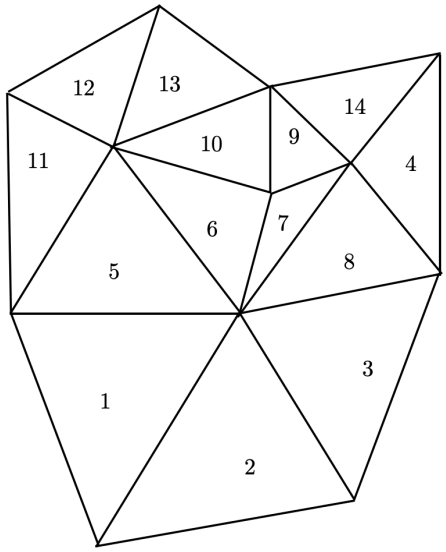
$$\mathbf{F}_c^{n+1} = \mathbf{F}_c^n + \left.\frac{\partial \mathbf{F}^n}{\partial \mathbf{U}}\right|_c \delta\mathbf{U}_c + \mathcal{O}(\Delta t^2). \tag{4.3.13}$$

where $\mathbf{F}_c^{n+1}$ represents the physical flux in cell c at time t^{n+1}, $\mathbf{F}_c^n$ represents the physical flux at time t^n and $\left.\dfrac{\partial \mathbf{F}^e}{\partial \mathbf{U}}\right|_c$ is the exact Jacobian of the Euler equations estimated at $\mathbf{U}_c$ such that for a perfect gas EOS:

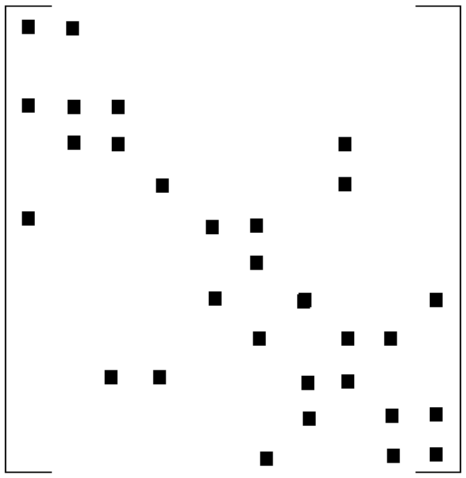
$$\left.\frac{\partial \mathbf{F}^e}{\partial \mathbf{U}}\right|_c = \begin{bmatrix} 0 & \mathbf{n}^t & 0 \\ (\gamma-1)\dfrac{q^2}{2}\mathbf{n} - q_n\mathbf{n} & \mathbf{v}\otimes\mathbf{n} + (1-\gamma)\mathbf{n}\otimes\mathbf{v} + q_n\mathbb{I} & (\gamma-1)\mathbf{n} \\ \left(\dfrac{(\gamma-1)}{2}q^2 - H\right)q_n & H\mathbf{n}^t - (1-\gamma)\mathbf{v}^t q_n & \gamma q_n \end{bmatrix} \tag{4.3.14}$$

where $q^2 = u^2 + v^2$, $\mathbf{v} = [u, v]^t$, H is the enthalpy, $\mathbb{I}$ the 2×2 identity matrix and here $\mathbf{n} = \mathbf{n}_{pcf}$. The implicit flux writes

$$\mathbf{F}^{n+1}(\mathbf{U}_c, \mathbf{U}_d) = \mathbf{F}^n(\mathbf{U}_c, \mathbf{U}_d) + \frac{1}{2}\left(\left.\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right|_c \delta\mathbf{U}_c + \left.\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right|_d \delta\mathbf{U}_d\right) + \frac{|\Lambda_m^{c,d}|}{2}(\delta\mathbf{U}_c - \delta\mathbf{U}_d). \tag{4.3.15}$$



(a) 2D unstructured grid.



(b) Example of a system matrix for an unstructured mesh.

Figure 4.39: Illustration of the implicit operator on unstructured mesh.

The expression of the diagonal block at cell c, $\mathbb{E}^n_{cc}$, writes

$$\mathbb{E}^n_{cc} = \frac{1}{2} \sum_{d \in \mathcal{C}_f(c)} l_{cd} \left(\left. \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \right|_c + |\Lambda^{cd}_m| \mathbb{I} \right). \tag{4.3.16}$$

and the extra diagonal blocks for the line c for all $d \in \mathcal{C}_f(c)$ writes

$$\mathbb{E}^n_{cd} = \frac{1}{2} l_{cd} \left(\left. \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \right|_d - |\Lambda^{cd}_m| \mathbb{I} \right). \tag{4.3.17}$$

An illustration of an unstructured grid is sketched in figure 4.39(a). For instance, the stencil for cell number 6 includes the cells $5, 7$, and 10. An example of the block system matrix for this unstructured grid is displayed on figure 4.39(b) where nonzero block matrices are displayed as filled rectangle The appearance of the system matrix for unstructured grids has no regular pattern since the grid cells are in general numbered in an arbitrary order and only the main diagonal is always present.

4.3.2 Iterative method and preconditioning

There are many methods in linear algebra literature for solving non-symmetric sparse systems. Let us re-write the implicit scheme (4.3.8) as

$$\mathbb{J} \delta \mathbf{U} = -\mathbf{R}, \tag{4.3.18}$$

where $\mathbb{J}$ represents the non-symmetric block structured system matrix. A particularly suitable class of iterative techniques for the solution of large linear equations systems of this form are called the Krylov subspace methods. In this present work, we implement the generalized minimal residual (GMRES) Krylov subspace technique to solve the sparse non-symmetric linear system. The GMRES method minimizes the norm of the global residual, that is

$$\|\mathbb{J}\Delta\mathbf{U} + \mathbf{R}\| \tag{4.3.19}$$

over a set of m orthonormal vectors which span the Krylov subspace $\mathcal{K}_m$. Here, we make use of the KSP object from PETSc library. The KSP object provides efficient access to all of PETSc's linear system solvers, including the GMRES solver.

The efficiency of the iterative method depends strongly on a good preconditioner, allowing to accelerate the iterative process. Here, we implement the incomplete lower upper factorization (ILU) pre-conditioner with a convergence tolerance of 10^{-6}. We once again make use of the KSP object from PETSc to compute the pre-conditioner.

4.3.3 CFL ramping

A CFL ramping technique is used to help the scheme to converge to the steady state. The CFL ramping formula is given as

$$CFL = \min\left(1, \frac{k}{k_t}\right) CFL_t + \left(1 - \min\left(1, \frac{k}{k_t}\right)\right) CFL_s, \tag{4.3.20}$$

where k is the iteration number, k_t the target iteration number, CFL_s the starting CFL which is usually taken around 0.5 and CFL_t the target CFL number given by the user. Therefore, the scheme starts with a small CFL to let the iteration process mimic an explicit integration.

Remark 4.3.1. *The implicit time-marching scheme described here is implemented for both two-point and multi-point schemes. Since the approximation of the Jacobian for the residuals is linearised using the simple Rusanov flux, the system matrix of the multi-point scheme is therefore considered to be 'face-based' and only the explicit term on the right-hand side of the multi-point scheme takes into account the nodal velocity vectorial parameter. Future work for the implicit multi-point scheme will involve a more consistent linearisation of the Jacobian using the vector of nodal velocity.*

4.3.4 Numerical validation

Hypersonic flow over a half cylinder test case We simulate an inviscid flow at Mach 20 around a half cylinder blunt body subject to an incoming hypersonic flow characterized by $(\rho_0, u_0, v_0, p_0, \gamma) = (1, Ma\sqrt{\gamma}, 0, 1, 1.4)$. The steady state resulting flow is simulated with both the two-point and multi-point implicit time integrating procedure. The simulation ends when the residual (4.3.19) is smaller than 10^{-12}. The computational

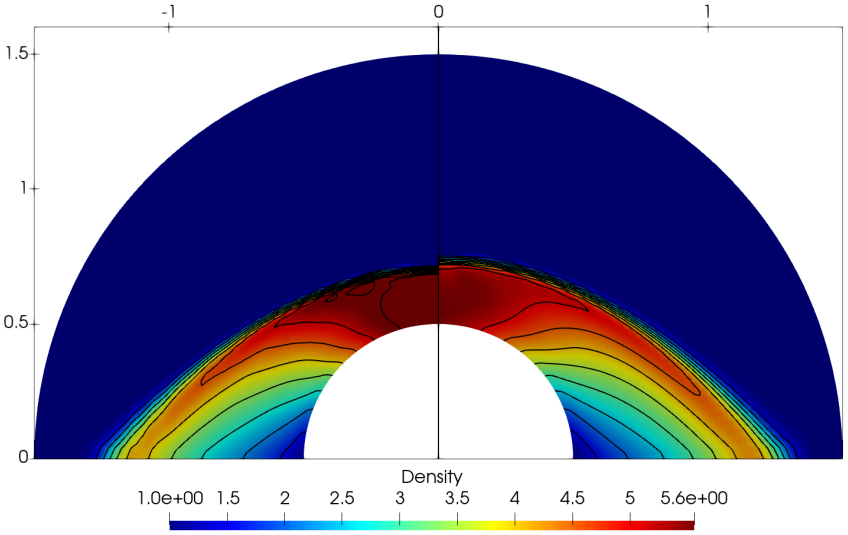
domain covers a large enough domain which contains half of a cylinder centered at the origin with a radius $r = 0.5$, and a left incoming hypersonic flow. The cylinder surface is a wall boundary condition, the bottom/upper boundary conditions are free outflow and the left boundary is an inflow condition. We run this test case on 5000 structured quadrangles and 5671 unstructured triangles.

Figure 4.40(a) illustrates the numerical density of both two-point (left) and multi-point (right) implicit schemes with $\text{CFL}_t = 10$ and $k_t = 10$ using the CFL ramping method on 5000 quadrangles. We observe post-shock oscillation on the left side, suggesting that the carbuncle instability is developesd with the two-point scheme. This can be further proven by comparing the density computed at the stagnation point. The exact density at the stagnation point can be calculated analytically with the Rankine-Hugoniot relation for supersonic steady flow over a blunt body and gives $\rho^{ex} = 6.4378$ (refer to appendix 1.3). The numerical density at the stagnation point computed with the two-point scheme is of $\rho^{two-point} = 6.67$, which exceeds the analytical solution due to the carbuncle instability, and the numerical density at stagnation point of the multi-point scheme gives $\rho^{multi-point} = 5.56$. The convergence of the $L2$ residual is displayed in figure 4.40(b). The two-point scheme does not converge due to the shock instability no matter the CFL used. As for the multi-point scheme, we observe a faster convergence for $\text{CFL} = 10$.

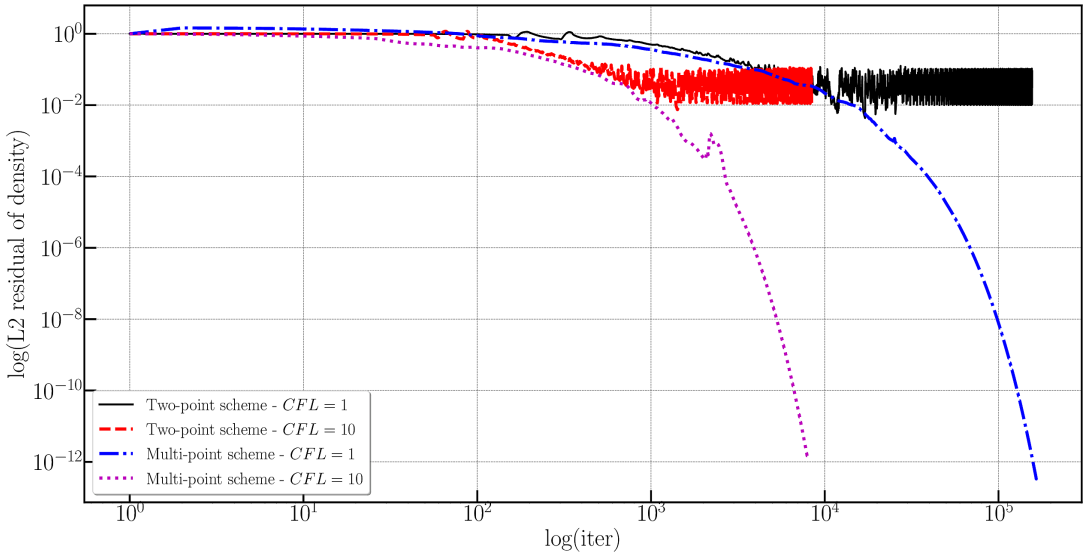
Similarly to the previous study, figure 4.41(a) illustrates the numerical density of both two-point (left) and multi-point (right) implicit schemes with $CFL = 10$ this time on 5671 triangles. We observe once again the carbuncle instability with the two-point scheme which is also proved in figure 4.41(b) where the $L2$ convergence of the of two-point scheme does not converge due to this instability. The residual converges appropriately for the multi-point scheme, and the residual converges at a faster rate with $CFL = 10$ as expected. The numerical density at the stagnation point computed with the two-point scheme on triangular mesh is of $\rho^{two-point} = 7.31$, which once again exceeds the analytical density, and the numerical density at stagnation point of the multi-point scheme gives $\rho^{multi-point} = 5.91$.

Supersonic jet test case This test case is a one-dimensional Riemann problem in a two-dimensional domain. The computational domain is $\Omega \in [0, 1] \times [0, 1]$ with 200×200 Cartesian mesh. The left boundary is an inflow condition split into two parts: $(\rho, u, v, p, Ma, \gamma)_{top} = (0.25, 4, 0, 0.25, 3.38, 1.4)$, and $(\rho, u, v, p, Ma, \gamma)_{bottom} = (1, 2.4, 0, 0.5, 2.87, 1.4)$. The remaining boundaries are treated as supersonic outlets. Figure 4.42 is the numerical density of the two-point (a) and multi-point (b) scheme with $CFL = 10$. We then compare the numerical solutions at the outlet $x = 1$ to the analytical solution obtained with and exact Riemann solver and a reference solution from Nishikawa & Kitamura (2008) in 4.43(a). Lastly, figure 4.43(b) illustrates the $L2$ residual convergence in terms of the number of iteration taken in the log scale for $CFL = 1, 10, 100$ & 1000 with the multi-point scheme.

Oblique shock test case This test case presents an oblique shock along a wedge where the analytical solution can be calculated. The computational domain is $\Omega \in [0, 0.21] \times [0, 0.1]$ and the wedge has an angle of $\theta = 10°$ starting from $x = 0.05$. and a Mach

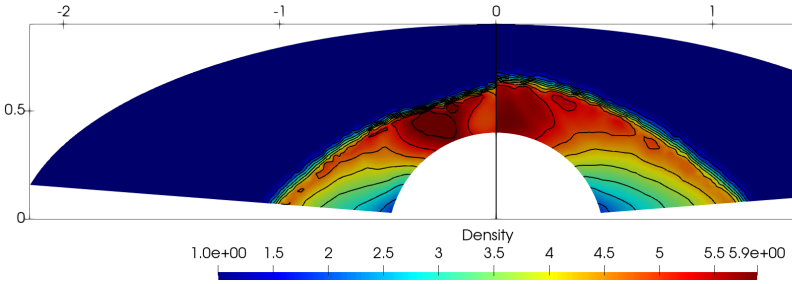


(a) Numerical density.

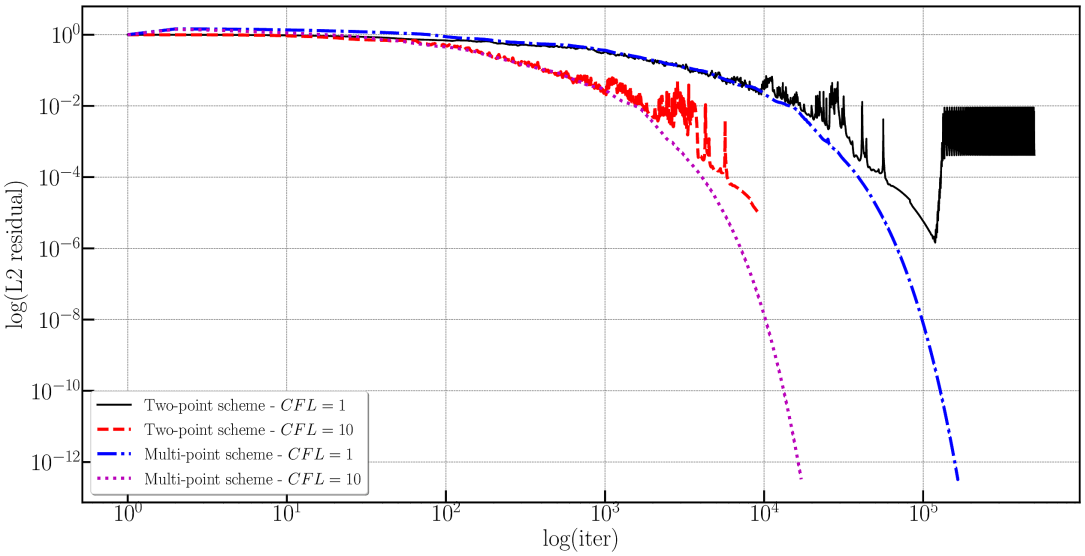


(b) L2 residual convergence.

Figure 4.40: Mach 20 flow over half cylinder — Density iso-line of both two-point and multi-point implicit schemes on 5000 quadrangle cells at $CFL = 10$ and residual convergence.

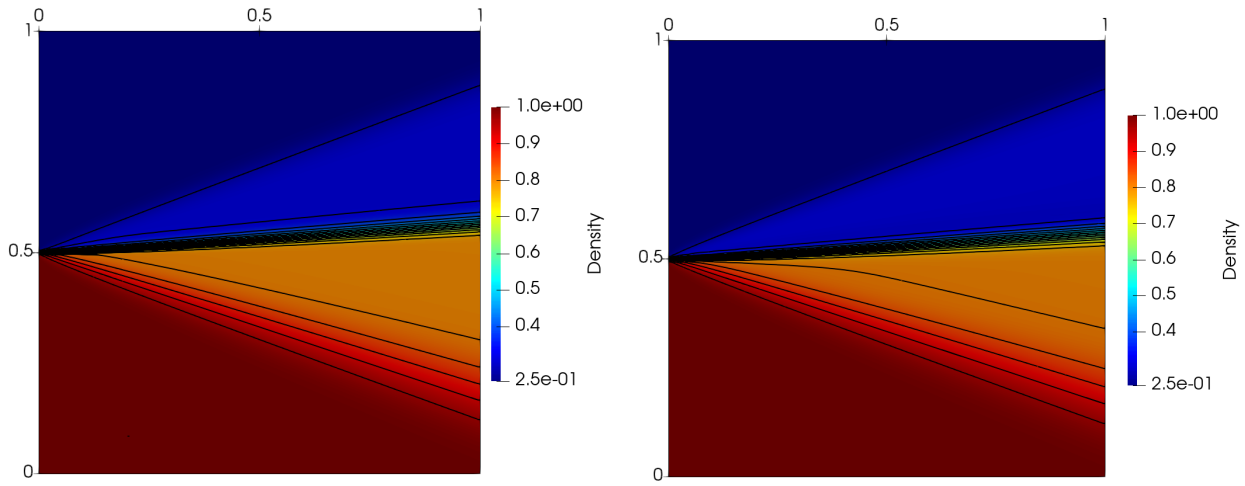


(a) Numerical density.



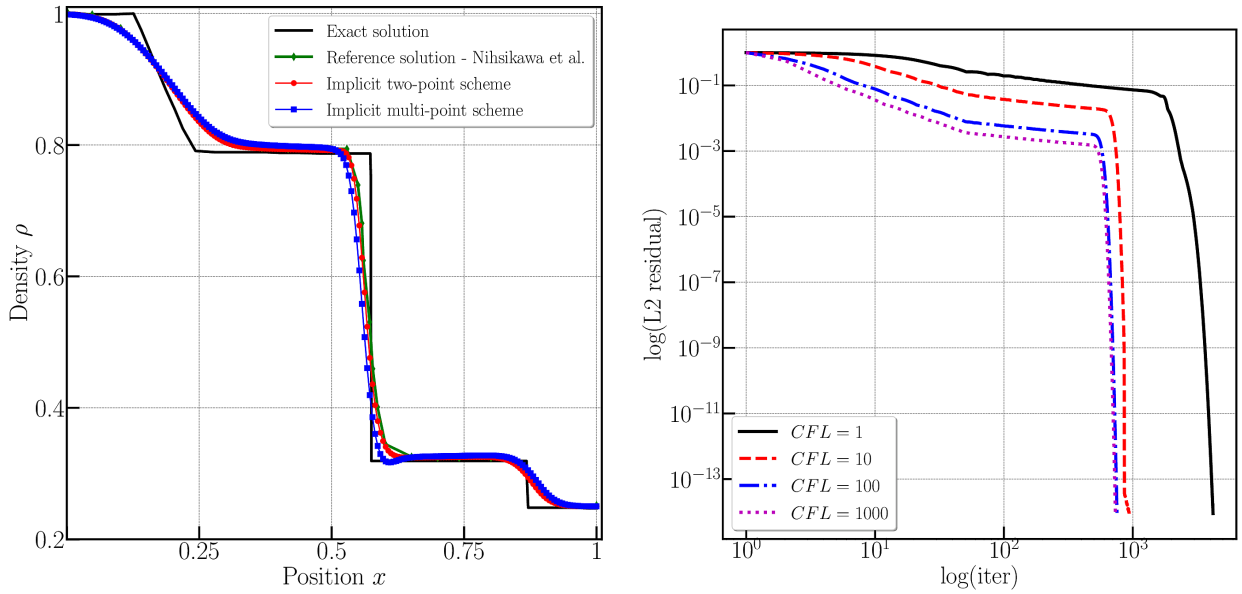
(b) L2 residual convergence.

Figure 4.41: Mach 20 flow over half cylinder — Density iso-line of both two-point and multi-point implicit schemes on 5671 triangle cells at $CFL = 10$ and residual convergence.



(a) Two-point scheme. (b) Multi-point scheme.

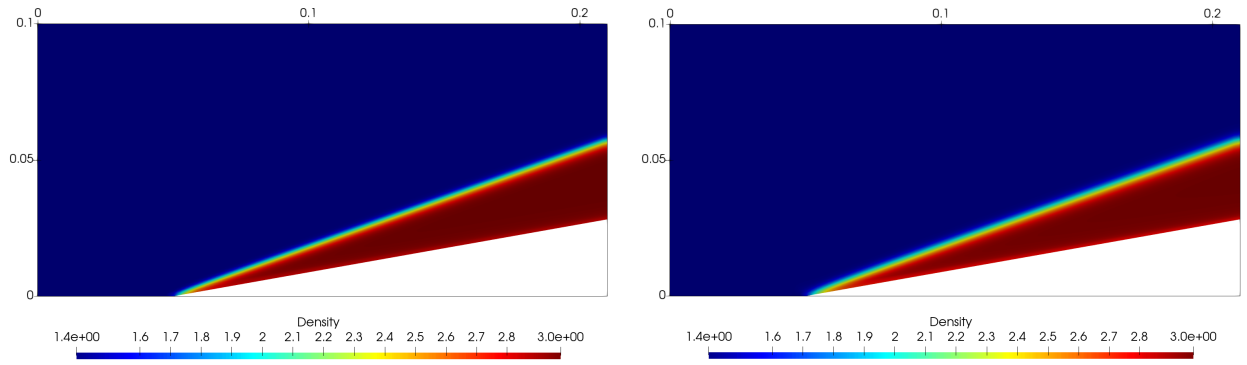
Figure 4.42: Supersonic jet test case — Comparison of numerical density of both implicit two-point and multi-point scheme at $CFL = 10$.



(a) Numerical density. (b) Residual convergence.

Figure 4.43: Supersonic jet test case — Comparison of numerical density with analytical and reference solution and residual convergence with different CFL.

5 inflow. The initial condition of this test case is $(\rho_0, u_0, v_0, p_0, \gamma) = (1.4, Ma, 0, 1, 1.4)$ and we run the test case on 50×50 Cartesian mesh. The steady state resulting flow is simulated with both the two-point and multi-point implicit time marching procedure and the simulation ends when the residual is smaller than 10^{-12}. Figure 4.44 illustrates the numerical density of the implicit two-point (a) and multi-point (b) schemes respectively. The numerical density and pressure is then compared with theoretical values as shown in figure 4.45 where profile of the solution at the outlet $x = 0.2$ is plotted. We also compare



(a) Two-point scheme.

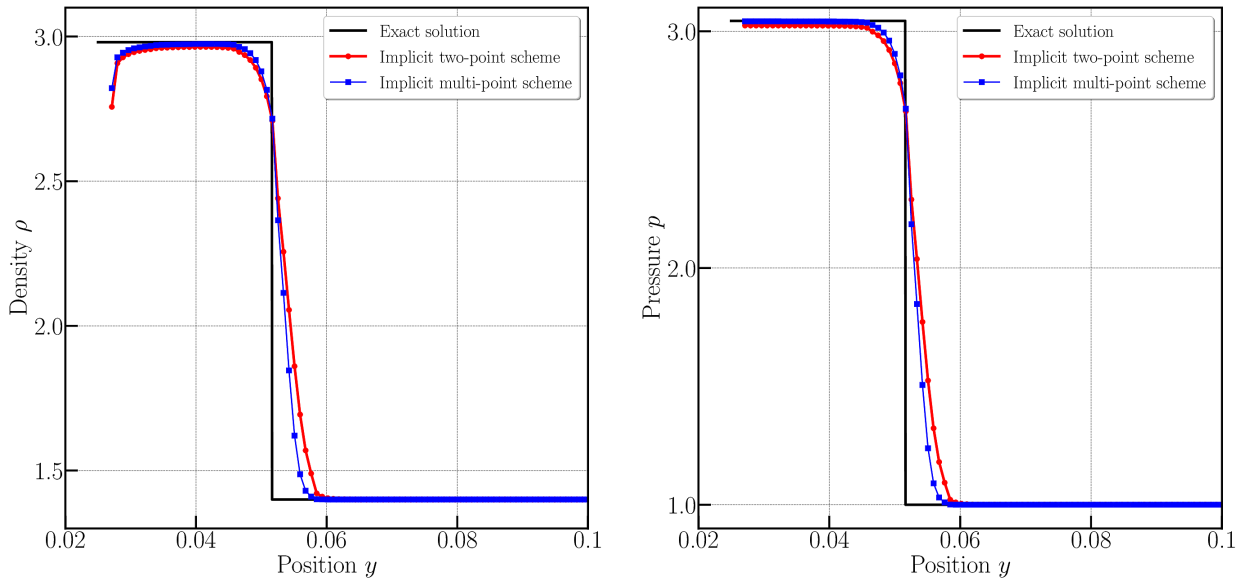
(b) Multi-point scheme.

Figure 4.44: Oblique shock test case — Comparison numerical density of two-point and multi-point implicit scheme at $CFL = 10$.

the convergence of the $L2$ residual in terms of the number of iterations for both schemes at $CFL = 1$ and $CFL = 10$.

Blunt cone-flare test case The blunt cone-flare test case from Savino & Paterna (2005) has the dimension and computational mesh as illustrated in figure 4.28 with figure 4.28(a) presenting the geometry of the object and figure 4.28(b) the $N = 61494$ triangular mesh on the computational domain. The free stream condition is a $Ma = 6$ inviscid flow and the initial condition of the test case is $(\rho_0, u_0, v_0, p_0, \gamma) = (1, Ma\sqrt{\gamma}, 0, 1, 1.4)$. We will be using a symmetry boundary condition on the bottom of the axis. We run this test case using the multi-point implicit scheme using $CFL = 10, 100$ and 1000. The numerical density and pressure are illustrated in figure 4.47 whereas the comparison of the residual convergence is shown in figure 4.48.

ARD and Pre-X Last but not least, we run the implicit time discretization scheme on both the ARD and Pre-X test case that were previously described in 4.2.4. Here, the simulation is ran on a two-dimensional slice of the object with the objective of assessing the capacity of the two-dimensional time implicit scheme. Therefore, at this stage, we will not be analyzing the flow, nor the pressure coefficient of the object. The free stream conditions remain unchanged for both test cases. The computation is ran on $N = 320000$ cells for ARD and $N = 320000$ cells for Pre-X, both using the multi-point scheme with $CFL = 10000$. The numerical pressure profiles are illustrated in figures 4.49(a) and 4.50(a) respectively. Figures 4.49(b) and 4.50(b) show the convergence of the $L2$ residual at $CFL = 10000$.



(a) Comparison of density. (b) Comparison of pressure.

Figure 4.45: Oblique shock test case — Comparison of two-point and multi-point scheme implicit scheme at $CFL = 10$.

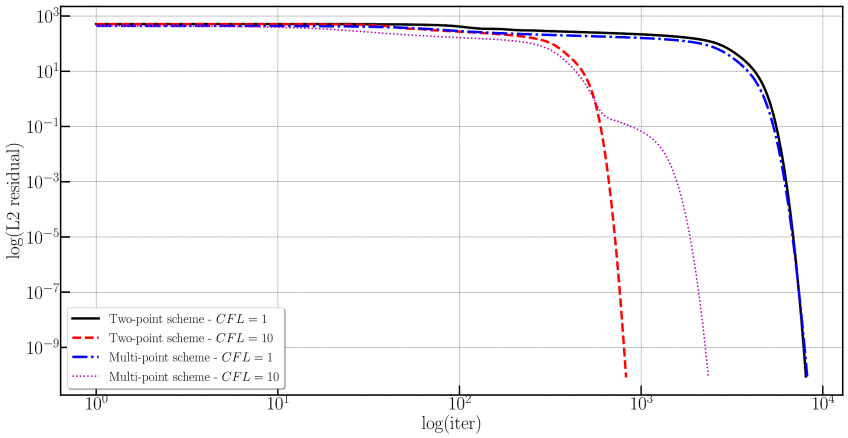
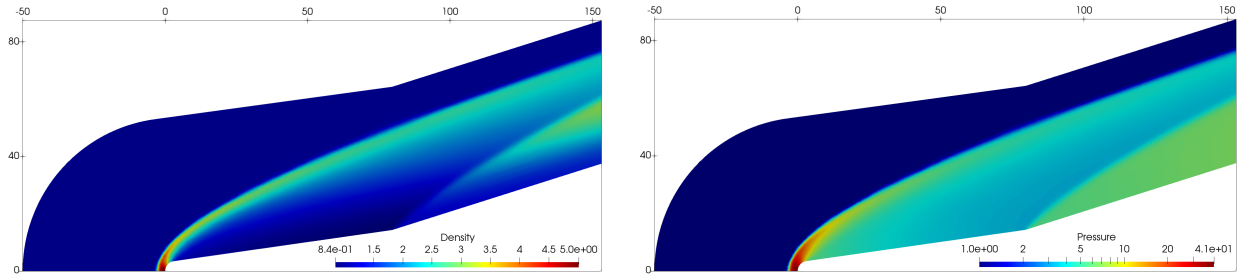


Figure 4.46: Oblique shock test case — Comparison of two-point and multi-point scheme implicit scheme at $CFL = 10$.

4.4 Chapter summary

In the previous chapter, the theoretical framework of a generic multidimensional Godunov-type Finite Volume scheme for systems of conservation laws on unstructured grids is presented. This novel scheme is subface-based, requiring subface fluxes by calling upon simple approximate Riemann solvers and a node-based sufficient condition to ensure conservativity of the scheme.

To present a thorough demonstration of this numerical scheme, an application to the system of gas dynamics has been thoroughly depicted in this chapter, starting off with the



(a) Numerical density. (b) Numerical pressure.

Figure 4.47: Blunted cone-flare test case — Numerical results of multi-point scheme at $CFL = 1000$

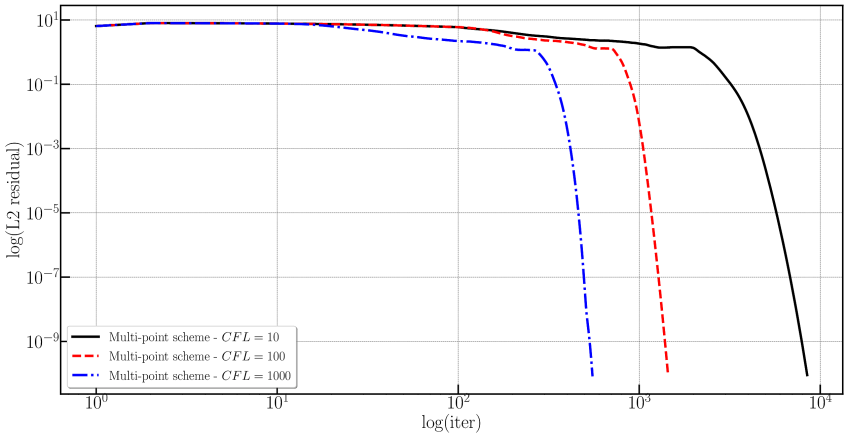
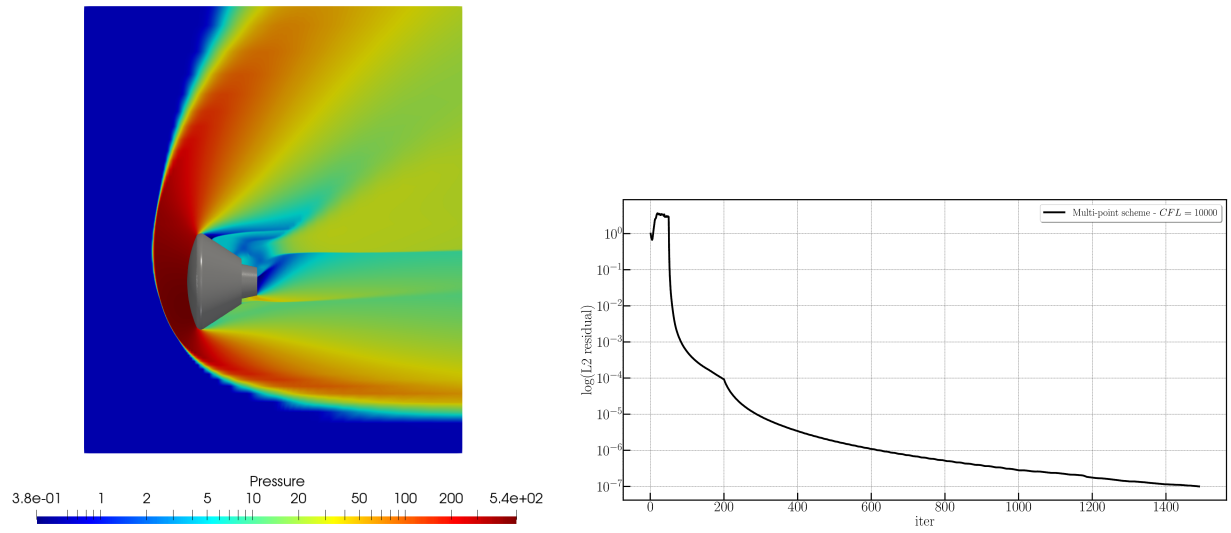
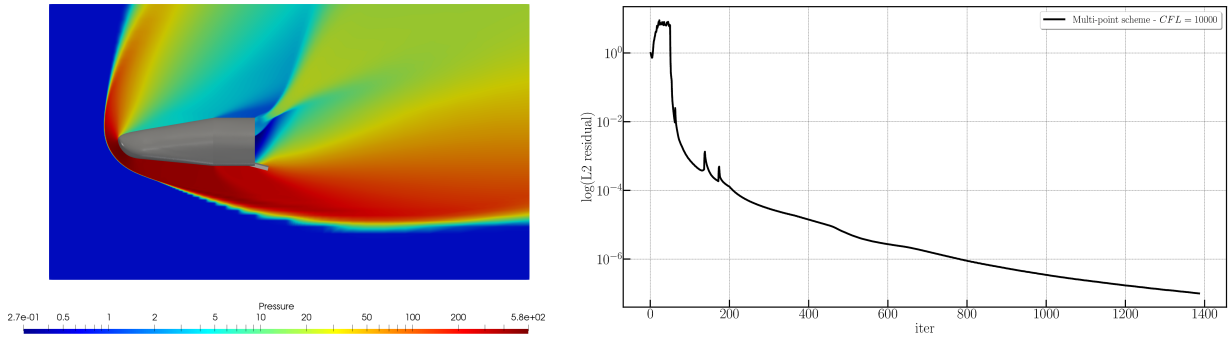


Figure 4.48: Blunted cone-flare test case — Comparison of residual convergence of the multi-point scheme implicit scheme with $CFL = 10, 100$ and 1000.



(a) Numerical pressure. (b) Residual convergence.

Figure 4.49: ARD — Numerical pressure profile and the residual convergence of ARD computed with the implicit multi-point scheme at $CFL = 10000$.



(a) Numerical pressure. (b) Residual convergence.

Figure 4.50: Pre-X — Numerical pressure profile and the residual convergence of Pre-X computed with the implicit multi-point scheme at $CFL = 10000$.

two-dimensional case. A Lagrangian approximate Riemann solver is first established for the Lagrangian gas dynamics equations projected onto the normal direction $\mathbf{n}$. Explicit conditions on the discontinuity velocities are derived to ensure positivity of specific volume and internal energy and entropy control. The conservation condition applied onto the two-dimensional gas dynamics equations then boils down to the Lagrangian nodal solver, allowing us to approximate the vectorial parameter for the nodal velocity $\mathbf{v}_p$.

We then use these elements as the building block to design the corresponding Eulerian simple Riemann solver. Once again, with the Lagrange-to-Euler mapping, hypothesis (H_1) and (H_2) and the fundamental relation, the Eulerian solver is obtained, and by construction the associated Eulerian scheme is multi-point, conservative, entropic, and positive under a well-defined CFL condition. Moreover, the wave-speeds of the Riemann solver are appropriately ordered by construction. A summary on the scheme with its associated subface-based flux is presented. An extension to the second-order of accuracy is also achieved using classical piecewise linear reconstruction associated with a slope limiter for the space discretization and the Runge-Kutta method for the time discretization.

This novel subface-based scheme is named the multi-point scheme and the classical one as two-point scheme. A set of test cases have been simulated with the aim of assessing its robustness by comparing the results of both two-point and multi-point schemes. A few sanity checks were done such as the modified Sod shock tube which was compared with the Roe solver with and without entropy fix, and the cylindrical Sod shock tube with a comparison of CPU time for both two-point and multi-point schemes on first and second order. Then, the classical isentropic vortex is done in order to run a convergence analysis for the first and second order accuracy. A series of test cases mainly from Quirk (1994)'s catalog was also implemented. The aim here is to exhibit the characteristic of the multi-point scheme where it is insensitive to numerical instabilities while comparing to the two-point scheme. These test cases include the odd-even decoupling and its modified version, hypersonic flow over a half cylinder, the Sedov blast wave problem, the Noh implosion and the forward-facing step test case.

A straightforward extension to the three-dimensional gas dynamics equations is then accomplished. Before presenting the numerical results, we introduced the three-

dimensional geometrical entities used, a summary of the three-dimensional subface-based scheme and also a few important PETSc libraries that were used for the 3D code data management, especially for the MPI implementation. A second-order extension is also accomplished to improve accuracy. The numerical validation for the 3D gas dynamics include a sanity check with the 3D spherical Sod shock tube, the Sedov blast wave and the hypersonic flow over a sphere where once again we compare with the two-point scheme, the blunted cone-flare where we compare the 3D solution with the 2D axisymmetric one, and lastly 3D re-entry vehicles such as the MaRV, ARD and Pre-X. These test cases were compared with reference solutions from experimental results or code results and show good coherence.

A final extension involving the implicit time-stepping discretization was included in this study. The objective of the implicit scheme is to study steady inviscid flows. The theoretical framework for this scheme is first presented, followed by the presentation of the iterative method and preconditioning used to solve the non-symmetric sparse system. A two-dimensional sequential simulation code for the implicit time discretization is developed and series of classical test cases were implemented to compare the residual convergence with different CFL for both two-point and multi-point schemes.

5 A Godunov-type Finite Volume scheme for the shallow water equations

Contents

In this chapter, we will be implementing the Godunov-type Finite Volume scheme presented in chapter 2 and chapter 3 on a system of conservation laws with source term, namely the shallow water equations. Shallow water flows are typical for flows where the vertical dimension is much smaller than the typical horizontal one, such in ocean, rivers, lakes, close to coasts for example. The challenge here is to take into account the source term in the equation, at the same time preserving the well-balanced nature of the system.

Similarly to the work accomplished for the gas dynamics model in the previous chapters, this chapter is once again inspired by the work of Gallice (2002*b*) where we first introduce the simple Riemann solver in the one-dimensional Lagrangian system of coordinates, this time taking into account the source term in the definition of the jump terms. Conservation laws with source terms often have steady states in which the flux gradients are nonzero but exactly balanced by the source terms. This Lagrangian Riemann solver is then used as a building block to construct the Eulerian simple Riemann solver using the Lagrange-to-Euler relation. This relation between Lagrangian and Eulerian system of coordinates allows for a clear definition of ordered Eulerian wave speeds, positivity preservation under a well-defined CFL like condition and an embedded well-balanced property treatment. Moreover, special care has to be taken to ensure that steady state solutions are maintained.

Similarly to the one-dimensional case, we then incorporate the subface-based numerical scheme (refer Chapter 3) to the two-dimensional shallow water equations. The multi-point scheme involves a nodal solver that allows to take into account all cell states surrounding a given node. Therefore, the aim of this section is to:

- Recall the shallow water equations in both Lagrangian and Eulerian representation;

- Constructing a well-balanced Godunov-type Finite Volume scheme that is positivity-preserving;
- Incorporating the Finite Volume scheme on the one-dimensional and two-dimensional shallow water equations;
- Validate the numerical scheme on demanding test cases.

5.1 Governing equations and notations

One considers the free surface flow of water in a channel by assuming that the water is incompressible, non-viscous, non-heat conducting, without bottom friction and subject to gravitational forces. The horizontal plan is given by coordinate x and y. The body force vector is denoted by $\mathbf{g} = (0, -g)$ where g is the gravitational acceleration that will be assumed constant in this work. The two-dimensional computational domain is denoted by $\mathcal{D}$. The bathymetry of the channel $B(x, y)$, is assumed fixed in time but may be irregular, and the water depth is denoted by $h(x, y, t)$, see figure 5.1. Lastly, the velocity vector of the fluid is referred to as $\mathbf{u} = (u, v)$.

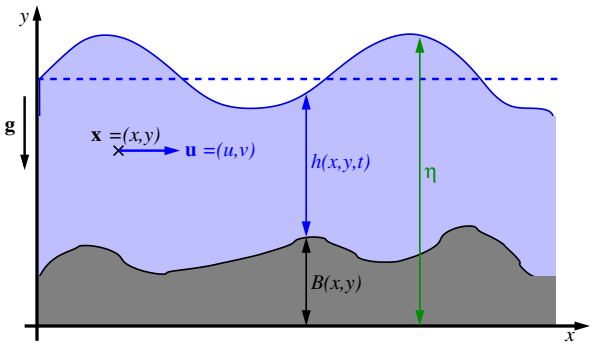


Figure 5.1: Illustration of the shallow water context.

The shallow water equations (see Leveque (1998)), are balanced laws and are represented by the following non-linear system of Partial Differential Equations (PDEs):

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbb{F}(\mathbf{U}) = \mathbf{S}(\mathbf{U}), \tag{5.1.1}$$

where, $\mathbf{U} = \mathbf{U}(\mathbf{x}, t)$, for $\mathbf{x} = (x, y) \in \mathbb{R}^2$ and $t \geq 0$, is the vector of conservative variables which takes values in $\mathbb{R}^3$ and $\mathbb{F} = \mathbb{F}(\mathbf{U})$ is the flux tensor in $\mathbb{R}^3 \times \mathbb{R}^2$, while $\mathbf{S}(\mathbf{U})$ in $\mathbb{R}^3$ is the source term involving the bathymetry. Let $\mathbf{e}_x$, $\mathbf{e}_y$ be the vectors of the Cartesian basis of $\mathbb{R}^2$, then $\mathbf{F} = \mathbb{F}\mathbf{e}_x \in \mathbb{R}^3$ and $\mathbf{G} = \mathbb{F}\mathbf{e}_y \in \mathbb{R}^3$ the components of the tensor flux. Its divergence then writes

$$\nabla \cdot \mathbb{F}(\mathbf{U}) = \frac{\partial}{\partial x}\mathbf{F}(\mathbf{U}) + \frac{\partial}{\partial y}\mathbf{G}(\mathbf{U}), \tag{5.1.2}$$

so that system (5.1.1) rewrites in the more classical form

$$\frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x}\mathbf{F}(\mathbf{U}) + \frac{\partial}{\partial y}\mathbf{G} = \mathbf{S}(\mathbf{U}), \tag{5.1.3}$$

where the vector or conservative variables $\mathbf{U}$, the flux vectors $\mathbf{F}(\mathbf{U})$ and $\mathbf{G}(\mathbf{U})$, and the source term vector $\mathbf{S}(\mathbf{U})$ are explicitly given by

$$\mathbf{U} = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} hu \\ hu^2 + p \\ huv \end{pmatrix},$$

$$\mathbf{G}(\mathbf{U}) = \begin{pmatrix} hv \\ huv \\ hv^2 + p \end{pmatrix}, \quad \mathbf{S}(\mathbf{U}) = \begin{pmatrix} 0 \\ -gh\partial_x B \\ -gh\partial_y B \end{pmatrix}. \tag{5.1.4}$$

The variable p can be seen as the pressure and in the case of shallow water equations it takes the simple form: $p \equiv p(h) = \frac{1}{2}gh^2$. If the source term is identically null, $\mathbf{S}(\mathbf{U}) = \mathbf{0}$, then system (5.1.3) becomes a hyperbolic system of conservation laws of which the eigenvalues are: $\lambda^- = u - a$ and $\lambda^+ = u + a$, where $a = \sqrt{gh}$ plays the role of the sound speed. Moreover, we can also interpret the homogeneous version of system (5.1.3) as the isentropic Euler system if $\rho \equiv h$ and a polytropic pressure law $p(\rho) = K\rho^\gamma$ is considered, with $K = 1$ and $\gamma = 2$. With such analogy we define the internal energy $\varepsilon = \varepsilon(h)$ with $\frac{d\varepsilon}{dh} = \frac{p}{h^2}$ and the energy as $E = \varepsilon + \frac{1}{2}|\mathbf{u}|^2$ such that the entropy inequality is given by

$$\frac{\partial}{\partial t}(hE) + \frac{\partial}{\partial x}(uhE + pu) + \frac{\partial}{\partial y}(vhE + pv) \leq -gh(u\partial_x B + v\partial_y B). \tag{5.1.5}$$

We remark that in the case of a regular bathymetry, $\partial_x B = 0$ and $\partial_y B = 0$, the foregoing equation is equivalent to

$$\frac{\partial}{\partial t}(hE_B) + \frac{\partial}{\partial x}(uhE_B + pu) + \frac{\partial}{\partial y}(vhE_B + pv) \leq 0, \tag{5.1.6}$$

where $E_B = E + B$. The exact solutions of system (5.1.3) are usually not known, generally tremendously complicated in particular due to the presence of irregularities such as shock waves, rarefaction corners or contact discontinuities. The set of admissible solutions of system (5.1.3) is characterized by the positivity of the water depth h:

$$\mathcal{A} = \{\mathbf{U},\ \text{s. t. } h > 0\}. \tag{5.1.7}$$

Numerical methods that are able to maintain numerical solutions within $\mathcal{A}$ are referred

to as being *positivity preserving*. Moreover, there exist large sets of stationary solutions for the shallow water equations when the bottom bathymetry is non zero. Of particular importance, are the so-called *Lake at Rest* solutions defined by:

$$\forall (x,y) \in \mathcal{D},\ \forall t > 0, \quad u(x,y,t) = 0, \quad h(x,y,t) + B(x,y) = \text{constant}. \tag{5.1.8}$$

Lake at rest solutions are important solutions that should be captured and maintained exactly when resorting to numerical methods. In such case, the numerical method is said to be *well-balanced* (WB) (refer to Gallice (2002*b*); Bouchut (2004); Gallardo *et al.* (2007)). In this work, we only consider wet situations where $h > 0$, and, we postpone the issue of dealing with wet/dry transition area for future developments.

5.2 One-dimensional well-balanced Godunov-type scheme

In this section, we will be tackling the one-dimensional shallow water equations by building a simple Lagrangian Riemann solver in the first place. After ensuring the properties of the Lagrangian solver, we then derive the Eulerian counterpart. The associated first order Finite Volume numerical scheme ensures well-balance and positivity preservation properties. A second-order accurate extension is then proposed based on Runge-Kutta time discretization and piece wise-linear limited reconstructions.

5.2.1 Governing equations

The one-dimensional shallow water equations in the Eulerian description are given by the following system:

$$\frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x}\mathbf{F}(\mathbf{U}) = \mathbf{S}(\mathbf{U}). \tag{5.2.1}$$

Here, $\mathbf{U} = \mathbf{U}(x,t)$, $\mathbf{F}(\mathbf{U}) = \mathbf{F}(\mathbf{U}(x,t))$ and $\mathbf{S}(\mathbf{U}) = \mathbf{S}(\mathbf{U}(x,t))$ are respectively the vector of conservative variables, flux vector and source term, more precisely,

$$\mathbf{U} = \begin{pmatrix} h \\ hu \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} hu \\ hu^2 + p \end{pmatrix}, \quad \mathbf{S}(\mathbf{U}) = \begin{pmatrix} 0 \\ -gh\partial_x B \end{pmatrix}.$$

The entropy inequality is given by

$$\frac{\partial}{\partial t}(hE) + \frac{\partial}{\partial dx}(uhE + pu) \leq -gh(u\partial_x B). \tag{5.2.2}$$

On the other hand, the one-dimensional shallow water equations in the Lagrangian coordinates can be written in a simpler form

$$\frac{\partial}{\partial t}\mathbf{V} + \frac{\partial}{\partial m}\mathbf{G}(\mathbf{V}) = \mathbf{P}(\mathbf{V}), \tag{5.2.3}$$

where m is the mass variable defined as $\mathrm{d}m = h\,\mathrm{d}x$, and

$$\mathbf{V} = \begin{pmatrix} 1/h \\ u \end{pmatrix}, \quad \mathbf{G}(\mathbf{V}) = \begin{pmatrix} -u \\ p \end{pmatrix}, \quad \mathbf{P}(\mathbf{V}) = \begin{pmatrix} 0 \\ -gh\partial_m B \end{pmatrix},$$

and for which the entropy inequality becomes

$$\frac{\partial}{\partial t}E + \frac{\partial}{\partial m}(pu) \leq -ghu\,\partial_m B. \tag{5.2.4}$$

Now that we introduced the one-dimensional shallow water equations in Eulerian and Lagrangian representation, we proceed to establishing a simple approximate Lagrangian Riemann solver.

5.2.2 Simple approximate Riemann solver for Lagrangian shallow water

Once again inspired by the seminal works of Gallice (2002*a*), we start by designing an approximate Riemann solver for shallow water equations written under the Lagrangian representation. Considering a Godunov-type scheme for the Lagrangian system (5.2.3), the associated simple approximate Riemann solver with wave speeds $-\lambda_l$, 0 and $\lambda_r > 0$ is given by

$$\mathbf{W}^{Lagr}\left(\mathbf{V}_l, \mathbf{V}_r, \frac{m}{t}\right) = \begin{cases} \mathbf{V}_l & \text{if} \quad \dfrac{m}{t} \leq -\lambda_l, \\ \mathbf{V}_l^\star & \text{if} \quad -\lambda_l \leq \dfrac{m}{t} < 0, \\ \mathbf{V}_r^\star & \text{if} \quad 0 < \dfrac{m}{t} \leq \lambda_r, \\ \mathbf{V}_r & \text{if} \quad \dfrac{m}{t} \geq \lambda_r. \end{cases} \tag{5.2.5}$$

Here, λ_l and λ_r are parameters that are strictly positive. The intermediate states write $\mathbf{V}_s^\star = (1/h_s^\star, u_s^\star)$ for $s = l, r$. Following the work of Gallice (2002*a*), the consistency of the approximate Riemann solver, is ensured provided that the following equation is satisfied,

$$-\lambda_l(\mathbf{V}_l^\star - \mathbf{V}_l) + \lambda_r(\mathbf{V}_r - \mathbf{V}_r^\star) = [\![\mathbf{G}]\!] - [\![m\mathbf{P}]\!], \tag{5.2.6}$$

where $[\![\mathbf{G}]\!] = \mathbf{G}_r - \mathbf{G}_l$ is the variable jump and $[\![m\mathbf{P}]\!]$ the jump of the source term. Replacing each component in (5.2.6) yields

$$-\lambda_l\left(\frac{1}{h_l^\star} - h_l\right) + \lambda_r\left(h_r - \frac{1}{h_r^\star}\right) \quad = \quad -(u_r - u_l), \tag{5.2.7}$$

$$-\lambda_l\left(u_l^\star - u_l\right) + \lambda_r\left(u_r - u_r^\star\right) = \left(p_r - p_l\right) + g\overline{h[\![B]\!]}, \tag{5.2.8}$$

where $\overline{h[\![B]\!]} = [\![m]\!]\overline{h\partial_m B}$ and $\overline{h\partial_m B}$ is an arbitrary approximation of $h\partial_m B$ that we shall determine later. Analogically to the gas dynamics equation and for the sake of simplicity, we represent $\overline{[\![p]\!]} = [\![p]\!] + \overline{h[\![B]\!]}$ where $\overline{[\![p]\!]}$ is the total pressure jump and equation (5.2.8) becomes

$$-\lambda_l\left(u_l^\star - u_l\right) + \lambda_r\left(u_r - u_r^\star\right) = \overline{[\![p]\!]}. \tag{5.2.9}$$

Now, assuming that the hypothesis (H_1) is satisfied, we can proceed to write

$$u_l^\star - \lambda_l\frac{1}{h_l^\star} = u_l - \lambda_l\frac{1}{h_l}, \tag{5.2.10}$$

$$u_l^\star = u_r^\star, \tag{5.2.11}$$

$$u_r^\star + \lambda_r\frac{1}{h_r^\star} = u_r + \lambda_r\frac{1}{h_r} \tag{5.2.12}$$

We can define $u^\star = u_l^\star = u_r^\star$ thanks to (5.2.11), and with equations (5.2.7), (5.2.8), (5.2.10), (5.2.12), the intermediate states yield

$$u^\star = \frac{1}{\lambda_l + \lambda_r}\left(\lambda_l u_l + \lambda_r u_r - \overline{[\![p]\!]}\right), \tag{5.2.13}$$

$$\frac{1}{h_l^\star} = \frac{1}{h_l} - \frac{1}{\lambda_l(\lambda_l + \lambda_r)}\left(\overline{[\![p]\!]} - \lambda_r(u_r - u_l)\right), \tag{5.2.14}$$

$$\frac{1}{h_r^\star} = \frac{1}{h_r} + \frac{1}{\lambda_r(\lambda_l + \lambda_r)}\left(\overline{[\![p]\!]} + \lambda_l(u_r - u_l)\right). \tag{5.2.15}$$

From the previous equations, it is clear that the positivity is ensured provided that λ_l and λ_r are large enough because the denominator is quadratic in λ and can be made as small as necessary to compensate the sum of the jumps in pressure, velocity and source term. Moreover, for sufficiently large wave speeds, the solver is also entropic in the sense that

$$\left(p_r u_r - p_l u_l\right) + u^\star g\overline{h[\![B]\!]} \leq \lambda_l\left(E_l^\star - E_l\right) + \lambda_r\left(E_r^\star - E_r\right). \tag{5.2.16}$$

We are now in position to determine explicit conditions on the wave speeds to ensure positivity preserving properties.

5.2.2.a Positivity-preserving and entropy consistent approximate of the Lagrangian Riemann solver

Substituting (5.2.13) into (5.2.10-5.2.11), $h_l^\star$ and $h_r^\star$ can be written in a convex combination of the wave speeds,

$$\begin{aligned} h_l^\star &= \frac{\lambda_l}{\lambda_l+\lambda_r}\left(h_l - \frac{\overline{[\![p]\!]}}{(\lambda_l)^2}\right) + \frac{\lambda_r}{\lambda_l+\lambda_r}\left(h_l + \frac{[\![u]\!]}{\lambda_l}\right), \\ h_r^\star &= \frac{\lambda_r}{\lambda_l+\lambda_r}\left(h_r - \frac{\overline{[\![p]\!]}}{(\lambda_r)^2}\right) + \frac{\lambda_l}{\lambda_l+\lambda_r}\left(h_r + \frac{[\![u]\!]}{\lambda_r}\right), \end{aligned} \tag{5.2.17}$$

with $\overline{[\![p]\!]} = [\![p]\!] + g\overline{h}[\![B]\!]$. The positivity of the water height holds true provided that the terms between parentheses are positive. Therefore, similar condition the conditions on λ_l and λ_r as mentioned in 2.2.4.b is recovered and allows to ensure the positivity of $h_l^\star$ and $h_r^\star$, such that

$$\lambda_l = \max\left(h_l a_l, \sqrt{\overline{[\![p]\!]}_+ h_l}, -[\![u]\!]h_l\right), \quad \lambda_r = \max\left(h_r a_r, \sqrt{-\overline{[\![p]\!]}_+ h_r}, -[\![u]\!]h_r\right). \tag{5.2.18}$$

Where, for $x \in \mathbb{R}$, $(x)_+$ denotes the positive part of x, *i.e.*, $(x)_+ = \frac{1}{2}(x+|x|)$. In order to ensure the well-balanced property of the scheme, the lake at rest solution has to be maintained and must satisfy the following solution :

$$u_l = u_r = 0, \quad h_l + B_l = h_r + B_r = \text{constant} > 0.$$

Note that the Riemann solver produces $u^\star = 0$ if and only if

$$g\overline{h}[\![B]\!] = -(p_r - p_l), \tag{5.2.19}$$

which further implies that $\frac{1}{h_l^\star} = \frac{1}{h_l}$ and $\frac{1}{h_r^\star} = \frac{1}{h_r}$. Moreover, if we assure that $\overline{h[\![B]\!]} = \overline{h}(B_r - B_l)$, $\overline{h}$ can be defined as such

$$\overline{h} = \begin{cases} \dfrac{[\![p]\!]}{[\![f]\!]} & \text{if} \quad [\![h]\!] \neq 0 \\ h_l = h_r & \text{if} \quad [\![h]\!] = 0 \end{cases} \tag{5.2.20}$$

We are now in position to express the numerical flux in terms of the states of the Riemann solver

$$\mathbf{G}^\star = \frac{1}{2}(\mathbf{G}_l^\star + \mathbf{G}_r) - \frac{\lambda_l}{2}(\mathbf{V}_l^\star - \mathbf{V}_l) - \frac{\lambda_r}{2}(\mathbf{V}_r - \mathbf{V}_r^\star). \tag{5.2.21}$$

This concludes the design of our positivity preserving and well-balanced Lagrangian Riemann solver and the entropy stability will be studied in future work. This solver will be used next as a building block for the Eulerian counter-part.

5.2.3 Simple approximate Riemann solver for Eulerian shallow water

The extension of the Eulerian Riemann solver is obtained by means of a Lagrangian to Eulerian mapping, see Gallice (2003) and Chan *et al.* (2021). Here, the approximate Riemann solver associated to a Godunov-type scheme for the Eulerian system (5.2.1) is given by

$$\mathbf{W}^{Eul}\left(\mathbf{U}_l, \mathbf{U}_r, \frac{m}{t}\right) = \begin{cases} \mathbf{U}_l & \text{if} \quad \dfrac{x}{t} \leq -\Lambda_l, \\ \mathbf{U}_l^\star & \text{if} \quad -\Lambda_l \leq \dfrac{x}{t} < \Lambda_0, \\ \mathbf{U}_r^\star & \text{if} \quad \Lambda_0 < \dfrac{x}{t} \leq \Lambda_r, \\ \mathbf{U}_r & \text{if} \quad \dfrac{x}{t} \geq \Lambda_r. \end{cases} \tag{5.2.22}$$

Here, $-\Lambda_l, \Lambda_0, \Lambda_r$ are the Eulerian wave speeds and the intermediate states write $\mathbf{U}_s^\star = (h_s^\star, h_s^\star u_s^\star)$ for $s = l, r$. The Eulerian Riemann solver is simply deduced from the Lagrangian one by considering

$$\mathbf{U}_{l/r}^\star = \mathbf{U}(\mathbf{V}_{l/r}^\star), \tag{5.2.23}$$

and the Eulerian wave speeds are obtained using the Lagrange-Euler relation from Gallice (2003),Chan *et al.* (2021),

$$\Lambda_l = u_l - \lambda_l \frac{1}{h_l}, \qquad \Lambda_0 = u^\star, \qquad \Lambda_r = u_r + \lambda_r \frac{1}{h_r}. \tag{5.2.24}$$

Remark that the properties of the Eulerian Riemann solver are inherited from the Lagrangian one. Namely, the solver is positive under the explicit control of the Lagrangian wave speeds. Moreover, the Eulerian wave speeds are ordered by construction, thanks to the positivity of h: $-\Lambda_l \leq \Lambda_0 = u^\star \leq \Lambda_r$. The interface flux $\mathbf{F}^\star$ at $\dfrac{x}{t} = 0$ expressed in terms of the wave speeds and intermediate states yields

$$\mathbf{F}^\star = \frac{1}{2}(\mathbf{F}_l + \mathbf{F}_r) - \frac{|\lambda_l|}{2}(\mathbf{U}_l^\star - \mathbf{U}_l) - \frac{|\Lambda_0|}{2}(\mathbf{U}_r^\star - \mathbf{U}_l^\star) - \frac{|\Lambda_r|}{2}(\mathbf{U}_r - \mathbf{U}_r^\star). \tag{5.2.25}$$

5.2.4 Well-balanced Godunov-type scheme

Let us consider a one-dimensional mesh of the computational domain $\Omega = [x_{\min}, x_{\max}]$ made of $N > 0$ cells $\omega_i = [x_{i-1/2}, x_{i+1/2}]$, of size $\Delta x = x_{i+1/2} - x_{i-1/2}$. Built upon the Eulerian Riemann solver, the one-dimensional well-balanced first-order Finite Volume numerical scheme simply writes

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\left[\mathbf{U}_{i+1/2} - \mathbf{U}_{i-1/2}\right] - \Delta t\, \mathbf{S}(\mathbf{U}_i)^n, \tag{5.2.26}$$

here $\mathbf{S}(\mathbf{U})_i^n$ is an averaged value of the source term that we shall compute later. The generic numerical flux is given by

$$\mathbf{F}^\star_{i+\frac{1}{2}} = \frac{1}{2}\left(\mathbf{F}^n_i + \mathbf{F}^n_{i+1}\right) + \frac{|\Lambda_{l,i+\frac{1}{2}}|}{2}(\mathbf{U}^\star_{i+\frac{1}{2}} - \mathbf{U}_i) + \frac{|u^\star_{1+\frac{1}{2}}|}{2}(\mathbf{U}^\star_{r+\frac{1}{2}} - \mathbf{U}^\star_{l+\frac{1}{2}}) + \frac{|\Lambda_{r+\frac{1}{2}}|}{2}(\mathbf{U}_{i+1} - \mathbf{U}^\star_{r+\frac{1}{2}}). \tag{5.2.27}$$

The Eulerian wave speeds $\Lambda_{l,r}$ are determined for each interface thanks to (5.2.24), and they are derived naturally from the Lagrangian ones (5.2.18).
It remains to determine the discretization of the source terms along with a CFL condition ensuring the positivity preservation and the well-balance property. Recall that the well-balanced property is fulfilled if $u_l = u_r = 0$ and $h_l + B_l = h_r + B_r =$ constant, then $u^\star_l = u^\star_r = 0$, $h^\star_l = h_l$ and $h^\star_r = h_r$. Substituting these intermediate states into (5.2.27), the well-balance property for the momentum component means

$$-\frac{1}{2}\left(p^n_{i+1} - p^n_{i-1}\right) - \int_{\omega_i} g h^n(x)\partial_x B(x)\,\mathrm{d}x = 0, \tag{5.2.28}$$

because $u^n_i = 0$. We are looking for a well-balanced solution, that is $u^{n+1}_i = 0$. Replacing the pressure by its expression in terms of h yields

$$-\frac{1}{2}\left(\frac{1}{2}g(h^n_{i+1})^2 - \frac{1}{2}g(h^n_{i-1})^2\right) = -g\left(\frac{h^n_{i+1} + h^n_{i-1}}{2}\right)\left(\frac{h^n_{i+1} - h^n_{i-1}}{2}\right), \tag{5.2.29}$$

therefore the source term can be approximated as follows

$$-\int_{\omega_i} g h^n(x)\partial_x B(x)\mathrm{d}x \equiv -g\overline{h}_i \int_{\omega_i} \partial_x B(x)\mathrm{d}x = -g\overline{h}_i(B_{i+1/2} - B_{i-1/2}). \tag{5.2.30}$$

If we set $B_{i\pm1/2} = \frac{1}{2}(B_{i\pm1} + B_i)$ and $\overline{h}_i = \frac{1}{2}(h^n_{i+1} + h^n_{i-1})$, then in the case of a lake at rest solution, equation (5.2.28) is fulfilled. Indeed we have $B_i =$ constant $- h_i$ for all cell, therefore $(B_{i+1/2} - B_{i-1/2}) = \dfrac{B_{i+1} - B_{i-1}}{2} = -\dfrac{h^n_{i+1} - h^n_{i-1}}{2}$.
A well-balanced discretization of the source term gives

$$-\int_{\omega_i} g h^n(x)\partial_x B(x)\,\mathrm{d}x = -g\left(\frac{h^n_{i+1} + h^n_{i-1}}{2}\right)\left(\frac{B_{i+1} - B_{i-1}}{2}\right). \tag{5.2.31}$$

5.2.5 Time step control

The time step is controlled by a classical CFL condition based on the computed Eulerian wave speeds from (5.2.24) that depend on the Lagrangian ones (5.2.18) as

$$\Delta t \leq \min_i \left(\frac{\Delta x}{|u_i \pm c^\pm_i|}\right). \tag{5.2.32}$$

Further work will require a more established CFL condition such as the work done for the gas dynamics equation. This indicates one needs to express a convex combination in terms of the state variables and at the same time taking into account the source term. This way, the positivity of the Lagrangian and Eulerian intermediate states should guaranteed.

5.2.6 Second order extension

The second order temporal extension of the previously described Eulerian scheme is achieved by a classical predictor-corrector (P/C) scheme. The space discretization is brought to second order by means of piecewise linear reconstruction ($\mathbb{P}_1$) with Minmod or Van Leer slope limiter, that is a classical MUSCL-type scheme. Each inter-facial value is replaced by its piecewise linear limited reconstruction at intermediate half-time step. In our approach we reconstruct the variables h to ensure the positivity of the water height thanks to the use of a slope limiter, and the constant $= h + B$. The reconstructed bathymetry is deduced by subtraction of $h + B$ and h, hence the reconstruction of B is piecewise linear per cell and $\partial_x B$ is then constant per cell. Note that this choice ensures both, positivity preserving of the reconstructed states and the well-balanced character of the scheme as the reconstruction operator exactly preserves the lake at rest solutions. Concerning the source term (5.2.31), h being reconstructed, we can compute the reconstructed value of $h_{i+1/2}$ as the average values of the two reconstructions from cell i and $i+1$ (likewise for $h_{i-1/2}$), then $\int_{\omega_i} g h^n(x) \partial_x B(x)\,\mathrm{d}x \simeq g(\partial_x B)\dfrac{h_{i+1/2} + h_{i-1/2}}{2}$. Other formula are possible.
Such extensions via MUSCL and P/C are genuinely classical, thus we omit detailed description here and proceed to classical 1D shallow water test cases to assess the properties of the scheme.

5.2.7 One-dimensional numerical validation

In this section we present several classical test cases to assess the properties of the 1D scheme based on the approximate Riemann solver developed in section 5.2.3. The main purpose is to observe the well-balanced property of the numerical scheme and the positivity preservation of the specific volume. The CFL is set to 0.9, $N = 100$ unless otherwise stated, and the gravity constant $g = 9.81$. We may use the shorthand notation $q = hu$.

Accuracy test We consider the test proposed in Xing & Shu (2005) to measure the accuracy of the numerical scheme. The computational domain is $L = 1$ with periodic boundary conditions. The bottom function B and the initial conditions are:

$$B(x) = \sin^2(\pi x),\ h(x) = 5 + e^{\cos(2\pi x)},\ q(x) = hu(x) = \sin(\cos(2\pi x)).$$

The exact solution is unknown explicitly for this case. A reference solution is computed using the second order scheme with $N = 10000$. The simulation runs up to $t = 0.1$ since

shocks develop later in time for this test case. Table 5.1 presents the L^2 errors of the water height h and the associated orders of accuracy. We observe a plain first order and an almost second one.

Number of cells (N)	$\mathbb{P}_0$ scheme		$\mathbb{P}_1$ scheme	
	L^2 error	L^2 order	L^2 error	L^2 order
50	1.79×10^{-2}	—	3.12×10^{-2}	—
100	8.97×10^{-3}	0.99	1.09×10^{-2}	1.5
200	4.49×10^{-3}	0.99	3.89×10^{-3}	1.51
400	2.26×10^{-3}	0.99	1.35×10^{-3}	1.53
800	1.16×10^{-3}	0.97	4.58×10^{-4}	1.56
1600	5.96×10^{-4}	0.97	1.48×10^{-4}	1.63

Table 5.1: Accuracy test. L^2 errors and L^2 orders for variable h for both first-order $\mathbb{P}_0$ and second order $\mathbb{P}_1$ schemes.

Flows over a bump This series of test case from Goutal & Maurel (1997) allows to challenge the numerical scheme against well-balanced solutions of flows over a bump. The domain is $\Omega = [0 : 25]$ of length $L = 25$ m and the bathymetry is defined by

$$B(x) = \max(0, 0.2 - 0.05(x - 10)^2). \tag{5.2.33}$$

Different flow regimes are initialized following the initial conditions and boundary conditions given in table 5.2.

Still water over a bump The still water test case, also known as lake at rest, is a classical test case to validate the well-balanced property of a scheme. Figure 5.2 represents the numerical solution of the free surface at $t = 50$ s against the exact solution. The scheme obviously preserves the stationary solution and the L^1 error obtained for the water height h is of the order of machine precision, 5.55×10^{-17}, as expected.

Subcritical and supercritical flows over a bump A subcritical flow is determined when the velocity of the flow is smaller than the acoustic wave velocity, meaning that the

Flow regime	**Initial conditions**		**Boundary conditions**	
	$h + B$ (m)	hu (m^2/s)	$(hu)_{in}$ (m^2/s)	h_{out} (m)
Still water	0.5	0	-	-
Subcritical flow	2	0	4.42	2
Supercritical flow	2	0	2505.67	2 (h_{in})
Transcritical flow	0.66	0	1.53	0.66
Transcritical flow (shock)	0.33	0	0.18	0.33

Table 5.2: Initial and boundary conditions for the $1D$ flow over bump test cases.

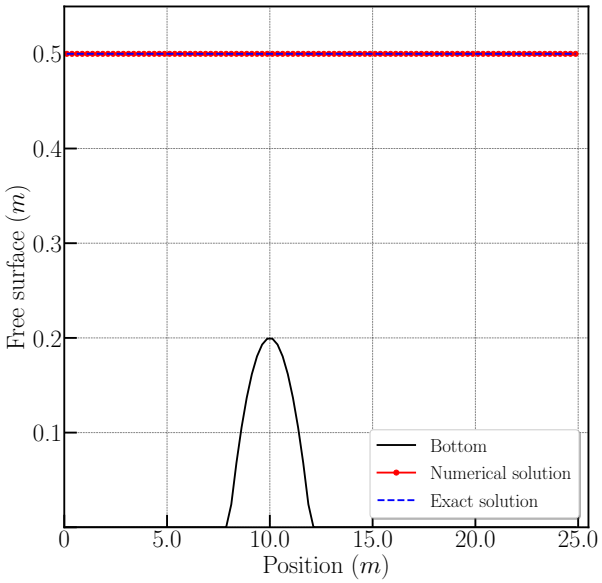


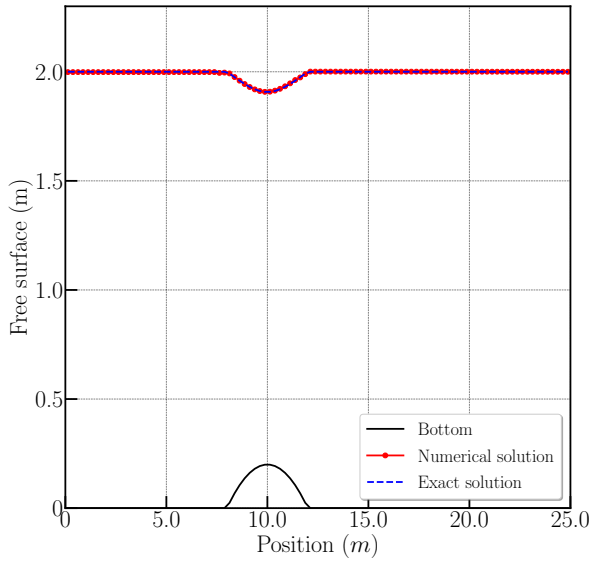
Figure 5.2: Still water — Bottom bathymetry (black) and free surface elevation at time $t = 50$ s for the numerical solution (red symbol) against the exact solution (blue dashed line).

Froude number, $\mathrm{Fr} = u/\sqrt{gh}$ is less than one. Contrarily a supercritical flow is observed when the Froude number is greater than one. The numerical free surfaces at $t = 100$ s for both of such flows are represented in figure 5.3 against the exact solution. The numerical solution is in well agreement with the exact solution in both cases.

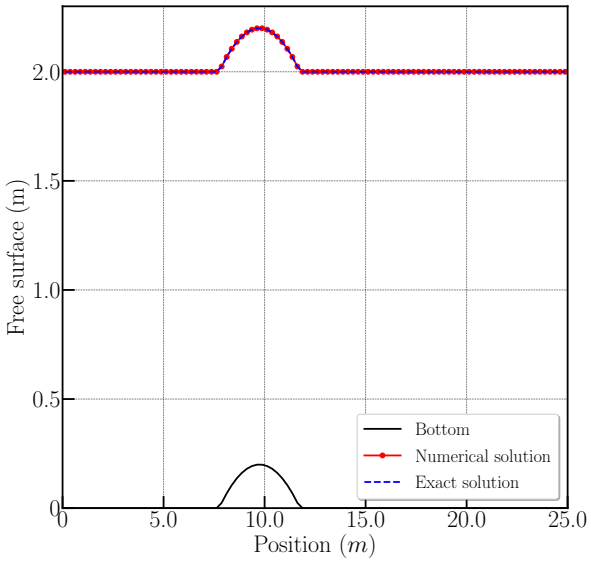
Transcritical flows over a bump A transcritical flow regime exists when certain parts of the flow are subcritical while others supercritical. For the first transcritical test, the flow is subcritical before the bump and supercritical after. The second test corresponds to a transcritical flow with shock, the flow is subcritical everywhere except for the supercritical point at $x = 10$ above the bump, creating a stationary hydraulic shock wave. Figure 5.4 illustrates the free surface of these flows at $t = 100$ s against the exact solution. While for the first test the agreement is almost perfect, the presence of the shock wave pollutes the numerical solution. Indeed an overshoot is observed just after the shock.

Dry bed generation The dry bed generation test case from Gallardo *et al.* (2007) is essential to verify the preservation of the positivity of the water height. In this test a vacuum state (i.e. dry bed) is formed in-between two rarefaction waves travelling in opposite directions. The computational domain is $\Omega = [0 : 10]$ of length $L = 10$ m and we observe the results at final time $t = 1$ s. Outflow boundary conditions are considered, and the initial conditions are

$$h(x) = 0.1\mathrm{m}, \quad q(u) = (hu)(x) = \begin{cases} -0.3\mathrm{m}^2/\mathrm{s} & \text{if} \quad x \leq 5, \\ 0.3\mathrm{m}^2/\mathrm{s} & \text{if} \quad \text{else.} \end{cases}$$

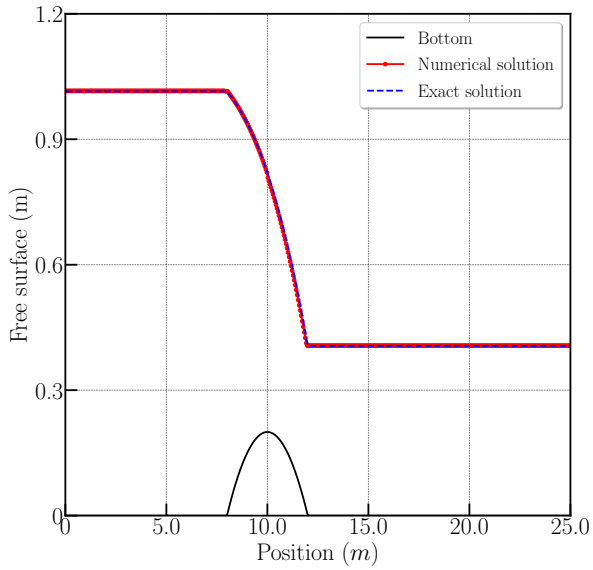


(a) Subcritical flow over a bump.

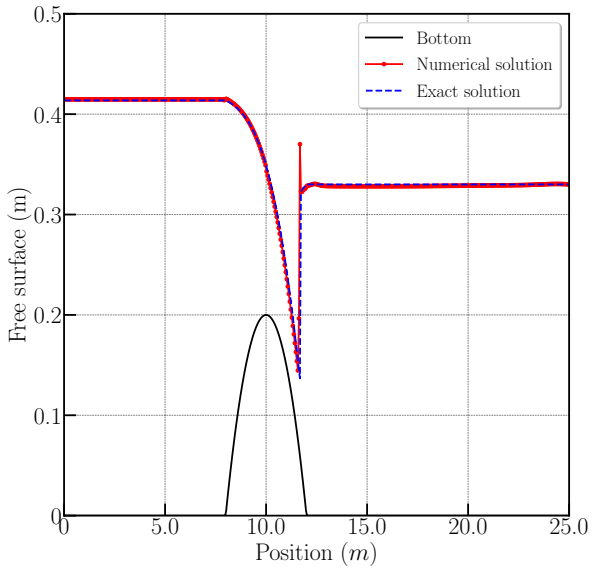


(b) Supercritical flow over a bump.

Figure 5.3: Subcritical and supercritical flow over a bump — Bottom bathymetry (black) and free surface elevation at time $t = 100$ s for the numerical solution (red symbol) against the exact solution (blue dashed line).



(a) Transcritical flow over a bump.

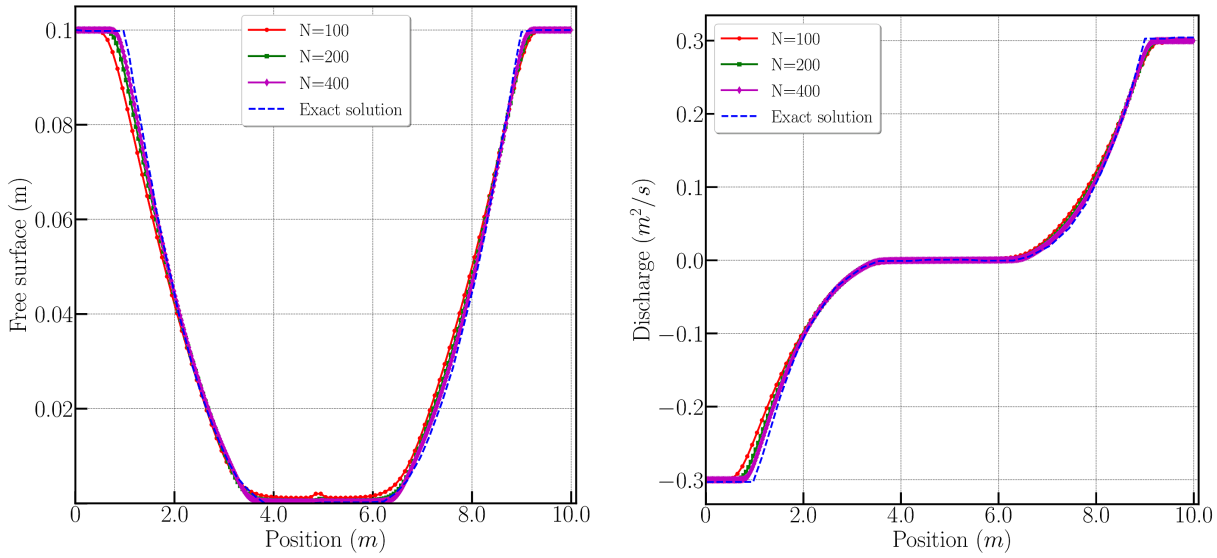


(b) Transcritical flow with shock over a bump.

Figure 5.4: Transcritical flow over a bump — Bottom bathymetry (black) and free surface elevation at time $t = 100$ s for the numerical solution (red symbol) against the exact solution (blue dashed line).

Notice that this test case involves a flat bottom. Figure 5.5 shows the results of a mesh convergence for both water height and discharge respectively for the first order scheme against the exact solution (dashed blue line). The grid is refined under the sequence 100, 200 and 400 cells. The numerical solution converges towards the exact one without

any positivity problem as expected since small amount of water remains on the dry area (here $h \approx 10^{-16}$). A modification of this test case, as proposed in Gallouet *et al.* (2003),

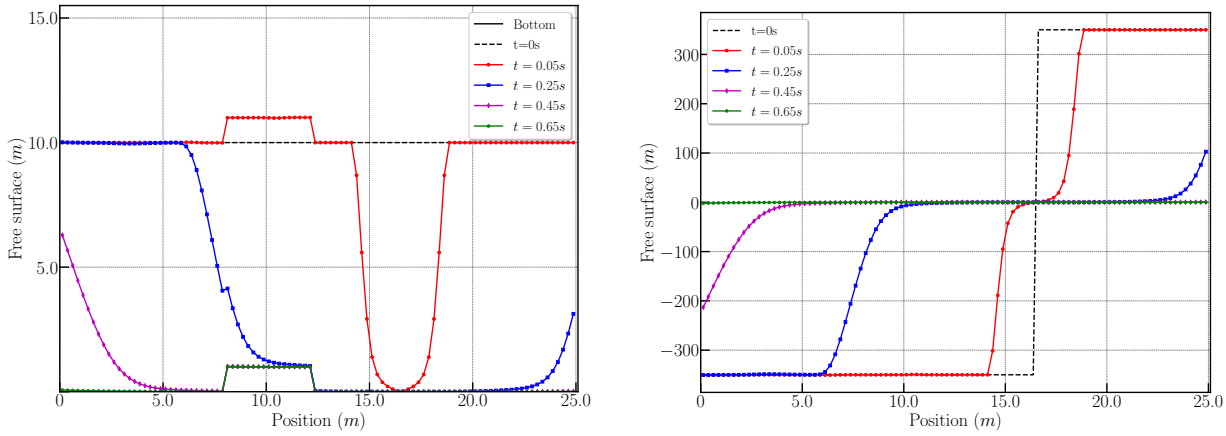


(a) Free surface elevation. (b)Discharge.

Figure 5.5: Dry bed generation test case — Grid convergence for $N = 100$, 200 and 400 cells.

introduced a non-flat bottom bathymetry. The computational domain length is $L = 25$ m and the bathymetry is defined as

$$B(x) = \begin{cases} 1 \text{ m} & \text{if} \quad 25/3 \leq x \leq 25/2, \\ 0 \text{ m} & \text{else.} \end{cases}$$



(a)Free surface. (b) Discharge.

Figure 5.6: Modified dry bed generation —Free surface (left) and discharge (right) at times $t = 0s, 0.05s, 0.25s, 0.45s$ and $0.65s$.

In Figure 5.6 we present the numerical results obtained with $N = 100$ cells and several

intermediate times before the final time $t = 0.65$ s. The free surface and discharge are presented respectively on the left and right panels. No visible problem is observed. As expected because our scheme does not deal yet with wet-dry front, a small amount of water is still present on the bathymetry.

In this section, we presented a well-balanced Godunov-type scheme for the one-dimensional shallow water equations. This scheme is a re-interpretation of the work established by Gallice (2002*a*). A simple approximate Lagrangian Riemann solver is built in the first place, and the well-balance property as well as the positivity preservation property is studied. The Eulerian counterpart is then achieved with the Lagrange-to-Euler mapping and by construction, the Eulerian scheme inherits the properties of the Lagrangian ones. Numerical validations demonstrate the well-balanced property of the scheme and the positivity-preserving property of the scheme is also assessed with a few near-dry test cases. Nonetheless, an in-depth study and analysis on the positivity-preserving property with an appropriate time step condition and the entropy stability of the scheme is yet to worked on.

5.3 Two-dimensional well-balanced Godunov-type scheme

Now that the one-dimensional well-balanced scheme is established, we are in position to design the two-dimensional scheme solving the shallow-water system (5.1.1). We will be implementing the multidimensional Godunov-type scheme described in chapter 3, this time on the the two-dimensional shallow-water equations where a source term is present. Let us recall a few important notations before proceeding.

Notations The computational domain is a polygonal portion of $\mathbb{R}^2$ paved with a set of non overlapping polygonal cells ω_c with c is the generic label of the cell. The generic label of a point is p, $\mathbf{x}_p$ denotes its vector position and $\mathcal{P}(c)$ the set of vertices of ω_c. The vertices of cell ω_c are counter-clockwise ordered, and p^- and p^+ are respectively the previous and the next points with respect to p, refer to figure 5.7. The subcell ω_{pc} is the quadrangle formed by joining the cell centroid, $\mathbf{x}_c$, to the midpoints of $[\mathbf{x}_{p^-}, \mathbf{x}_p]$, $[\mathbf{x}_p, \mathbf{x}_{p^+}]$ and to $\mathbf{x}_p$. The set of subcells ω_{pc} for $p \in \mathcal{P}(c)$ constitutes a partition of the cell ω_c, that is,

$$\omega_c = \bigcup_{p \in \mathcal{P}(c)} \omega_{pc}. \tag{5.3.1}$$

We also introduce the set of faces of cell ω_c and denote it $\mathcal{F}(c)$. Each face f of cell c is decomposed into subfaces by means of the partition of c induced by the subcells pc for $p \in \mathcal{P}(c)$. As such we define $\mathcal{SF}(pc)$ the set of subfaces attached to the corner pc, which is nothing but the set of faces of subcell ω_{pc} impinging at point p. We denote respectively by l_{pcf} and $\mathbf{n}_{pcf} = (n_x, n_y)_{pcf}$ the measure and the unit outward normal of the subface f. We observe that the set of subfaces $\mathcal{SF}(pc)$ for $p \in \mathcal{P}(c)$ constitutes a partition of the

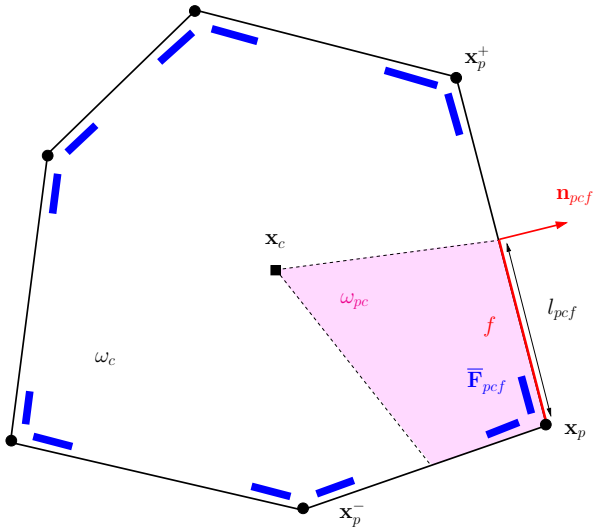


Figure 5.7: Geometrical entities attached to the polygonal cell ω_c.

set of faces of ω_c, that is,

$$\mathcal{F}(c) = \bigcup_{p\in\mathcal{P}(c)} \mathcal{SF}(pc). \tag{5.3.2}$$

Given a cell c and one of its face f, a unique 'neighbor' cell can be associated and we refer to it as $d(c, f)$ or d to shorten the notation. The set of neighbors of cell c is denoted by $\mathcal{N}(c)$.

In what follows, we will be recalling the subface-based Finite Volume scheme presented in chapter 3 in its generic form, this time taking into account the source term.

5.3.1 Subface-based Finite-Volume scheme with source term

We start by integrating the system of conservation laws (5.1.1) over ω_c and employing Green formula yields

$$|\omega_c|\frac{\mathrm{d}\mathbf{U}_c}{\mathrm{d}t} + \int_{\partial\omega_c} \mathbb{F}(\mathbf{U})\mathbf{n}\,\mathrm{d}s = \int_{\omega_c} \mathbf{S}(\mathbf{U})\,\mathrm{d}v, \tag{5.3.3}$$

where $\mathbf{U}_c(t) = \dfrac{1}{|\omega_c|}\displaystyle\int_{\omega_c} \mathbf{U}(\mathbf{x}, t)\,\mathrm{d}v$ is the cell-averaged value of $\mathbf{U}$ over ω_c and $\mathbf{U}_c^n \equiv \mathbf{U}_c(t^n)$ for any discrete time t^n. Employing a classical first-order explicit time integration turns (5.3.3) into

$$\mathbf{U}_c^{n+1} - \mathbf{U}_c^n + \frac{\Delta t}{|\omega_c|}\int_{\partial\omega_c} \mathbb{F}(\mathbf{U}^n)\mathbf{n}\,\mathrm{d}s = \frac{\Delta t}{|\omega_c|}\int_{\omega_c} \mathbf{S}(\mathbf{U}^n)\,\mathrm{d}v. \tag{5.3.4}$$

The Finite Volume scheme (5.3.4) requires the construction of an approximation of the normal flux integral. Following Gallice *et al.* (2022), this integral term is a node-based

approximation that relies on the partition of ω_c into subcells

$$\int_{\partial\omega_c} \mathbb{F}(\mathbf{U}^n)\mathbf{n}\,\mathrm{d}s = \sum_{p\in\mathcal{P}(c)} \int_{\partial\omega_{pc}\cap\partial\omega_c} \mathbb{F}(\mathbf{U}^n)\mathbf{n}\,\mathrm{d}s. \tag{5.3.5}$$

The surface integral term at the right-hand side of (5.3.5) is approximated along the subfaces by means of subface flux $\overline{\mathbf{F}}_{pcf}$ as follows

$$\int_{\partial\omega_{pc}\cap\partial\omega_c} \mathbb{F}(\mathbf{U}^n)\mathbf{n}\,\mathrm{d}s = \sum_{f\in\mathcal{SF}(pc)} l_{pcf}\overline{\mathbf{F}}_{pcf},$$

and the numerical approximation of the subface flux will be done by means of approximate Riemann solver that will be described in section 5.3.3 onwards.
As for the source term, its integral is defined as

$$\frac{1}{|\omega_c|}\int_{\omega_c} \mathbf{S}(\mathbf{U}^n)\,\mathrm{d}v = \sum_{f\in\mathcal{F}(c)} \overline{\mathbf{S}}_f, \tag{5.3.6}$$

and its definition will be presented later in 5.3.5. Substituting the above forms into (5.3.4) gives the generic subface-based Finite Volume scheme

$$\mathbf{U}_c^{n+1} - \mathbf{U}_c^n + \frac{\Delta t}{|\omega_c|}\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} l_{pcf}\overline{\mathbf{F}}_{pcf} = \Delta t \sum_{f\in\mathcal{F}(c)} \overline{\mathbf{S}}_f. \tag{5.3.7}$$

With the aim of defining an approximate Riemann solver to characterize the subface flux $\overline{\mathbf{F}}_{pcf}$, we emulate the strategy from Chapter 3 and 4 (or see G. Gallice & Maire (2022)), where the subface-based Finite Volume scheme for Eulerian gas dynamics equations was built. The key feature of this scheme includes:

- **Node-based conservation condition** (refer to section 3.2.6). The subface-based Finite Volume scheme is considered conservative if the summation of the subface fluxes attached to subfaces impinging at p is equal to zero, such as

  $$\sum_{c\in\mathcal{C}(p)}\sum_{f\in\mathcal{SF}(pc)} l_{pcf}\overline{\mathbf{F}}_{pcf} = \mathbf{0}.$$

  This condition has to be fulfilled for all nodes.

- **Fundamental relation between Lagrangian and Eulerian framework**. (refer to section 3.3). The Eulerian Finite Volume scheme is built using the Lagrangian counterpart. The Lagrangian simple approximate Riemann solver is built in the first place. The **Lagrange-to-Euler mapping** then allows to express the Eulerian variables and fluxes in terms of the Lagrangian ones. The Eulerian wave speeds can be estimated directly in terms of the Lagrangian ones thanks to the $(\mathbf{H_1})$ hypothesis. and lastly, the **fundamental relation** $\overline{\mathbf{F}}_{\mathbf{n}}^{+} - \overline{\mathbf{F}}_{\mathbf{n}}^{+} = \overline{\mathbf{G}}_{\mathbf{n}}^{+} - \overline{\mathbf{G}}_{\mathbf{n}}^{+}$ that demonstrates the equivalence of the Lagrangian and Eulerian numerical fluxes.

In what follows, we present the systems of balanced laws in both Eulerian and Lagrangian framework. We then establish the Lagrangian simple approximate Riemann solver and the node-based conservation condition in the Lagrangian framework. This will then be used as a building block to construct the Eulerian counterpart.

5.3.2 Description of the one-dimensional Eulerian and Lagrangian systems of balanced laws

Let us consider the subface characterized by the unit outward $\mathbf{n}$. The normal and tangential components of the velocity vector $\mathbf{u}$ write respectively $u_{\mathbf{n}} = \mathbf{u}\cdot\mathbf{n}$ and $u_{\mathbf{t}} = \mathbf{u}\cdot\mathbf{t}$, and, obviously, $\mathbf{u} = u_{\mathbf{n}}\mathbf{n} + u_{\mathbf{t}}\mathbf{t}$. The vector of conservative variables and the flux projected onto the normal direction $\mathbf{n}$ write

$$\mathbf{U} = \begin{pmatrix} h \\ hu_{\mathbf{n}} \\ hu_{\mathbf{t}} \end{pmatrix}, \qquad \mathbf{F}_{\mathbf{n}} = \mathbb{F}\mathbf{n} = \begin{pmatrix} hu_{\mathbf{n}} \\ hu_{\mathbf{n}}^2 + p(h) \\ hu_{\mathbf{n}}u_{\mathbf{t}} \end{pmatrix}. \tag{5.3.8}$$

Therefore, the one-dimensional Eulerian system associated to the SW system in the $\mathbf{n}$ direction, where $x_{\mathbf{n}} = \mathbf{x}\cdot\mathbf{n}$, reads

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_{\mathbf{n}}(\mathbf{U})}{\partial x_{\mathbf{n}}} = \mathbf{S}_{\mathbf{n}}(\mathbf{U}), \tag{5.3.9}$$

where $\mathbf{S}_{\mathbf{n}}(\mathbf{U})$ is the associated source term $\mathbf{S}_{\mathbf{n}}(\mathbf{U}) = (0, q_{\mathbf{n}}, 0)^t$ with $q_{\mathbf{n}} = -gh(\nabla B)_{\mathbf{n}}$ and $(\nabla B)_{\mathbf{n}} = \nabla B\cdot\mathbf{n}$ to simplify the notation in what follows. Employing the Lagrange-to-Euler mapping we can derive the corresponding one-dimensional Lagrangian system

$$\frac{\partial \mathbf{V}}{\partial t} + \frac{\partial \mathbf{G}_{\mathbf{n}}(\mathbf{V})}{\partial m} = \mathbf{P}_{\mathbf{n}}(\mathbf{V}), \tag{5.3.10}$$

where m is the Lagrangian mass coordinate related to the Eulerian coordinate $x_{\mathbf{n}}$ and $\tau = 1/h$ is called the specific volume by identification with the gas dynamics. The Lagrangian vector of conservative variables, the Lagrangian flux and source term write

$$\mathbf{V} = \begin{pmatrix} \tau \\ u_{\mathbf{n}} \\ u_{\mathbf{t}} \end{pmatrix}, \qquad \mathbf{G}_{\mathbf{n}} = \begin{pmatrix} -u_{\mathbf{n}} \\ p \\ 0 \end{pmatrix}, \qquad \mathbf{P}_{\mathbf{n}}(\mathbf{V}) = \begin{pmatrix} 0 \\ q_{\mathbf{n}} \\ 0 \end{pmatrix}.$$

5.3.3 Lagrangian simple Riemann solver

This Lagrangian simple approximate Riemann solver structure mimics the continuous structure of the one-dimensional Lagrangian system of balance laws. It is naturally composed of four states $\mathbf{V}_l$, $\mathbf{V}_l^\star$, $\mathbf{V}_r^\star$ and $\mathbf{V}_r$ separated respectively by discontinuities of

speeds $-\lambda_l$, 0 and λ_r in the (m,t) plane.

$$\mathbf{W}_l\left(\mathbf{V}_l,\mathbf{V}_r,\frac{m}{t}\right)=\begin{cases}\mathbf{V}_l & \text{if} \quad \dfrac{m}{t}\leq -\lambda_l,\\ \mathbf{V}_l^\star & \text{if} \quad -\lambda_l<\dfrac{m}{t}\leq 0,\\ \mathbf{V}_r^\star & \text{if} \quad 0<\dfrac{m}{t}\leq \lambda_r,\\ \mathbf{V}_r & \text{if} \quad \lambda_r<\dfrac{m}{t}.\end{cases}$$

λ_l and λ_r are positive real parameters which must be constrained as to ensure the positivity and entropy stability properties of the Riemann solver following the methodology introduced in Gallice (2002*a*, 2003) and revised in Chan *et al.* (2021); Gallice *et al.* (2022). The states components write $\mathbf{V}_s=(\tau_s,u_{\mathbf{n},s},u_{\mathbf{t},s})^t$ and the intermediate states components $\mathbf{V}_s^\star=(\tau_s^\star,u_{\mathbf{n},s}^\star,u_{\mathbf{t},s}^\star)^t$ for $s=l,r$. Considering the jump condition for τ across the second wave, which is also equivalent to the (H_1) hypothesis, we get

$$0(\tau_r^\star-\tau_l^\star)-(u_{\mathbf{n},r}^\star-u_{\mathbf{n},l}^\star)=0 \implies u_{\mathbf{n},r}^\star=u_{\mathbf{n},l}^\star\equiv u_{\mathbf{n}}^\star.$$

Therefore, the jump conditions for τ boils down to

$$\lambda_l(\tau_l^\star-\tau_l)-(u_{\mathbf{n}}^\star-u_{\mathbf{n},l})=0, \tag{5.3.11a}$$
$$\lambda_r(\tau_r^\star-\tau_r)+(u_{\mathbf{n}}^\star-u_{\mathbf{n},r})=0. \tag{5.3.11b}$$

Now, following Chan *et al.* (2021) we complete the Lagrangian Riemann solver characterization introducing the intermediate fluxes for $s=l,r$

$$\overline{\mathbf{G}}_{\mathbf{n},s}=(-\overline{u}_{\mathbf{n},s},\ \overline{p}_s,\ 0)^t, \tag{5.3.12}$$

which satisfy the system

$$\lambda_l(\mathbf{V}_l^\star-\mathbf{V}_l)+\overline{\mathbf{G}}_{\mathbf{n},l}-\mathbf{G}_{\mathbf{n},l}=\overline{\mathbf{P}}_{\mathbf{n},l}, \tag{5.3.13a}$$
$$-\lambda_r(\mathbf{V}_r-\mathbf{V}_r^\star)+\mathbf{G}_{\mathbf{n},r}-\overline{\mathbf{G}}_{\mathbf{n},r}=\overline{\mathbf{P}}_{\mathbf{n},r}, \tag{5.3.13b}$$

where $\mathbf{G}_{\mathbf{n},s}=\mathbf{G}_{\mathbf{n}}(\mathbf{V}_s)$ and $\overline{\mathbf{P}}_{\mathbf{n},s}=\mathbf{P}_{\mathbf{n}}(\mathbf{V}_s)$ is an appropriate mean value of the source term for $s=l,r$. Combining the first components of (5.3.13a), (5.3.13b) with (5.3.11a), (5.3.11b) leads to $\overline{u}_{\mathbf{n},l}=u_{\mathbf{n}}^\star=\overline{u}_{\mathbf{n},r}$ because the source term is null for its first component. We arrive at the following expressions of the intermediate states, fluxes and source terms for $s=l,r$

$$\mathbf{V}_s^\star=\begin{pmatrix}\tau_s^\star\\ u_{\mathbf{n}}^\star\\ u_{\mathbf{t},s}^\star\end{pmatrix},\qquad \overline{\mathbf{G}}_{\mathbf{n},s}=\begin{pmatrix}-u_{\mathbf{n}}^\star\\ \overline{p}_s\\ 0\end{pmatrix},\qquad \overline{\mathbf{P}}_{\mathbf{n},s}=\begin{pmatrix}0\\ \overline{q}_{\mathbf{n},s}\\ 0\end{pmatrix},$$

As such, the system of 6 equations for 7 unknowns is given by (5.3.13a) and (5.3.13b), that is

$$(\mathcal{S}_l)\begin{cases} \lambda_l(\tau_l^\star-\tau_l)-(u_{\mathbf{n}}^\star-u_{\mathbf{n},l})=0,\\ \lambda_l(u_{\mathbf{n}}^\star-u_{\mathbf{n},l})+\overline{p}_l-p_l=\overline{q}_{\mathbf{n},l},\\ \lambda_l(u_{\mathbf{t},l}^\star-u_{\mathbf{t},l})=0, \end{cases}\quad (\mathcal{S}_r)\begin{cases} \lambda_r(\tau_r^\star-\tau_r)+u_{\mathbf{n}}^\star-u_{\mathbf{n},r}=0,\\ \lambda_r(u_{\mathbf{n}}^\star-u_{\mathbf{n},r})-(\overline{p}_r-p_r)=-\overline{q}_{\mathbf{n},r},\\ \lambda_r(u_{\mathbf{t},r}^\star-u_{\mathbf{t},r})=0. \end{cases}$$

We can express 6 unknowns in terms of the 7^{th} unknown, for instance $u_{\mathbf{n}}^\star$ in a parametric form. The left and right tangential velocities are simply given by solving the third equations to get $u_{\mathbf{t},s}^\star=u_{\mathbf{t},s}$ because $\lambda_s\neq 0$ for $s=l,r$. Moreover, we can introduce the 'new' pressures as

$$\widetilde{p}_l=p_l+\overline{q}_{\mathbf{n},l},\quad \text{and}\quad \widetilde{p}_r=p_r+\overline{q}_{\mathbf{n},r}. \tag{5.3.14}$$

Next the intermediate normal velocity $u_{\mathbf{n}}^\star$ is viewed as a parameter to express the 4 remaining unknowns, $\tau_{l/r}^\star$ and $\overline{p}_{l/r}$, solution of the 4 remaining equations:

$$(\mathcal{S}_l)\begin{cases} \lambda_l(\tau_l^\star-\tau_l)-(u_{\mathbf{n}}^\star-u_{\mathbf{n},l})=0,\\ \lambda_l(u_{\mathbf{n}}^\star-u_{\mathbf{n},l})+\overline{p}_l-\widetilde{p}_l=0, \end{cases}\quad (\mathcal{S}_r)\begin{cases} \lambda_r(\tau_r^\star-\tau_r)+u_{\mathbf{n}}^\star-u_{\mathbf{n},r}=0,\\ \lambda_r(u_{\mathbf{n}}^\star-u_{\mathbf{n},r})-(\overline{p}_r-\widetilde{p}_r)=0, \end{cases} \tag{5.3.15}$$

In other words, given $\lambda_s>0$ and $u_{\mathbf{n}}^\star$, we compute

$$\begin{cases} \tau_l^\star=\tau_l+\dfrac{(u_{\mathbf{n}}^\star-u_{\mathbf{n},s})}{\lambda_l},\\ \overline{p}_l=\widetilde{p}_l-\lambda_l(u_{\mathbf{n}}^\star-u_{\mathbf{n},s}) \end{cases}\qquad \begin{cases} \tau_r^\star=\tau_r-\dfrac{(u_{\mathbf{n}}^\star-u_{\mathbf{n},r})}{\lambda_r},\\ \overline{p}_r=\widetilde{p}_r+\lambda_r(u_{\mathbf{n}}^\star-u_{\mathbf{n},r}) \end{cases} \tag{5.3.16}$$

The value of $\lambda_s>0$ is restricted from below because to ensure the positivity of the specific volumes, from the first equations of $(\mathcal{S}_l)$ and $(\mathcal{S}_r)$ we deduce the conditions still parametrized by the normal velocity $u_{\mathbf{n}}^\star$ as

$$\lambda_l\geq-\frac{u_{\mathbf{n}}^\star-u_{\mathbf{n},l}}{\tau_l},\qquad \text{and}\qquad \lambda_r\geq\frac{u_{\mathbf{n}}^\star-u_{\mathbf{n},r}}{\tau_r}. \tag{5.3.17}$$

The last unknown is the velocity $u_{\mathbf{n}}^\star$ that we shall determine in the next subsection. Next, we invoke the consistency of the Lagrangian Riemann solver with its underlying balance law by summing (5.3.13a) and (5.3.13b) to get

$$\overline{\mathbf{G}}_{\mathbf{n},r}-\overline{\mathbf{G}}_{\mathbf{n},l}=\lambda_l(\mathbf{V}_l^\star-\mathbf{V}_l)-\lambda_r(\mathbf{V}_r-\mathbf{V}_r^\star)+\mathbf{G}_{\mathbf{n},r}-\mathbf{G}_{\mathbf{n},l}-\overline{\mathbf{P}}_{\mathbf{n},r}-\overline{\mathbf{P}}_{\mathbf{n},l}. \tag{5.3.18}$$

On the other hand, utilizing the expression of the components of $\overline{\mathbf{G}}_{\mathbf{n},r}$ and $\overline{\mathbf{G}}_{\mathbf{n},l}$ yields

$$\overline{\mathbf{G}}_{\mathbf{n},r}-\overline{\mathbf{G}}_{\mathbf{n},l}=(\overline{p}_r-\overline{p}_l)\begin{pmatrix}0\\1\\0\end{pmatrix}. \tag{5.3.19}$$

The consistency of the simple Lagrangian Riemann solver with its underlying balanced law amounts to study the solutions of $\overline{\mathbf{G}}_{\mathbf{n},r} - \overline{\mathbf{G}}_{\mathbf{n},l} = \mathbf{0}$. To this end, we compute the difference $\overline{p}_r - \overline{p}_l$ summing the second equations of $(\mathcal{S}_l)$ and $(\mathcal{S}_r)$

$$\overline{p}_r - \overline{p}_l = (\lambda_l + \lambda_r)\left\{u^\star_{\mathbf{n}} - \left[\frac{\lambda_l u_{\mathbf{n},l} + \lambda_r u_{\mathbf{n},r}}{\lambda_l + \lambda_r} - \frac{(\widetilde{p}_r - \widetilde{p}_l)}{\lambda_r + \lambda_l}\right]\right\}. \tag{5.3.20}$$

This equation incites us to introduce the normal velocity

$$\overline{u}_{\mathbf{n}} = \frac{\lambda_l u_{\mathbf{n},l} + \lambda_r u_{\mathbf{n},r}}{\lambda_l + \lambda_r} - \frac{(\widetilde{p}_r - \widetilde{p}_l)}{\lambda_r + \lambda_l}. \tag{5.3.21}$$

From this definition and (5.3.20) we conclude that if $u^\star_{\mathbf{n}} = \overline{u}_{\mathbf{n}}$ (resp. $u^\star_{\mathbf{n}} \neq \overline{u}_{\mathbf{n}}$), then the simple Lagrangian Riemann solver is (resp. is not) consistent with its underlying balance law and induces (resp. does not induce) a classical conservative Godunov-type Finite Volume scheme. Before moving on, we to the Eulerian solver, we demonstrate the Lagrangian equivalence of the node-based conservation condition.

Lagrangian node-based conservation condition Thanks to the fundamental property, the Eulerian node-based conservation condition is equivalent to the Lagrangian one and can be expressed as such

$$\sum_{f\in\mathcal{SF}(p)} l_{pf}\left\{-\left[\sum_{k=1}^{m}\lambda_k\left(\mathbf{V}_{k+1}-\mathbf{V}_k\right)\right]_{l,r} + \mathbf{G}_{\mathbf{n}_{pf},r} - \mathbf{G}_{\mathbf{n}_{pf},l}\right\} = \mathbf{0}. \tag{5.3.22}$$

This condition involves all faces f impinging at point p. By substituting the components in the above equation, the generic term between curly brackets yields

$$\begin{aligned} -\left[\sum_{k=1}^{m}\lambda_k\left(\mathbf{V}_{k+1}-\mathbf{V}_k\right)\right]_{l,r} + \mathbf{G}_{\mathbf{n}_{pf},r} - \mathbf{G}_{\mathbf{n}_{pf},l} &= \lambda_l(\mathbf{V}^\star_l - \mathbf{V}_l) - \lambda_r(\mathbf{V}_r - \mathbf{V}^\star_r) \\ &+ \mathbf{G}_{\mathbf{n},r} - \mathbf{G}_{\mathbf{n},l} - \overline{\mathbf{P}}_{\mathbf{n},l} - \overline{\mathbf{P}}_{\mathbf{n},r} \end{aligned} \tag{5.3.23}$$

The left and the right states correspond respectively to the cells c and d since the unit normal $\mathbf{n}_{pcf}$ is pointing from cell c towards cell d. The right-hand side of the foregoing equation coincides precisely with the right-hand side of (5.3.19) which has been derived to study the consistency of the simple Lagrangian solver with its underlying conservation laws. This means that here, we have arbitrarily employed the same conservation condition than the one derived for gas dynamics. Thus, by virtue of (5.3.19), equation (5.3.23) turns

successively into

$$
\begin{aligned}
-\left[\sum_{k=1}^{m}\lambda_k\left(\mathbf{V}_{k+1}-\mathbf{V}_k\right)\right]_{l,r}+\mathbf{G}_{\mathbf{n}_{pf},r}-\mathbf{G}_{\mathbf{n}_{pf},l}=\overline{\mathbf{G}}_{\mathbf{n},r}-\overline{\mathbf{G}}_{\mathbf{n},l} &= (\overline{p}_r-\overline{p}_l)\begin{pmatrix}0\\1\\0\end{pmatrix}\\
&= (\overline{p}_{r,f}-\overline{p}_{l,f})\begin{pmatrix}0\\1\\0\end{pmatrix}.
\end{aligned}
\tag{5.3.24}
$$

Therefore, the Finite Volume scheme is conservative provided that the following node-based condition is satisfied

$$
\sum_{f\in\mathcal{SF}(p)} l_{pf}(\overline{p}_{r,f}-\overline{p}_{l,f})\begin{pmatrix}0\\1\\0\end{pmatrix}=\mathbf{0}. \tag{5.3.25}
$$

Note that the second and third components of the foregoing vector correspond to $\mathbf{n}_{pf}$, and condition (5.3.25) implies

$$
\sum_{f\in\mathcal{SF}(p)} l_{pf}(\overline{p}_{r,f}-\overline{p}_{l,f})\mathbf{n}_{pf}=\mathbf{0}. \tag{5.3.26}
$$

At this point, $u^{\star}_{\mathbf{n}_{pf}}$ is still an unknown parameter attached to each subface impinging at node p. To close this system of equations, we assume that $u^{\star}_{\mathbf{n}_{pf}}$ corresponds to the projection of the unknown nodal vector $\mathbf{u}_p$ onto the unit normal $\mathbf{n}_{pf}$, that is for all p and f

$$
u^{\star}_{\mathbf{n}_{pf}}=\mathbf{v}_p\cdot\mathbf{n}_{pf}. \tag{5.3.27}
$$

This fundamental assumption reduces the number of unknowns to the vectorial unknown $\mathbf{u}_p$, which can be interpreted as an approximation of the nodal velocity. Thanks to (5.3.20) and (5.3.21) we are able to express the interface pressure differences such that

$$
\sum_{f\in\mathcal{SF}(p)} l_{pf}(\lambda_{l,f}+\lambda_{r,f})(\mathbf{v}_p\cdot\mathbf{n}_{pf}-\overline{u}_{\mathbf{n}_{pf}})\mathbf{n}_{pf}=\mathbf{0}. \tag{5.3.28}
$$

Finally, the node-based condition (5.3.26) (or equivalently (5.3.28)) boils down to the system

$$
\mathbb{M}_p\,\mathbf{u}_p=\mathbf{w}_p, \tag{5.3.29}
$$

where $\mathbf{u}_p$ is the unknown and

$$
\mathbb{M}_p=\sum_{f\in\mathcal{SF}(p)} l_{pf}(\lambda_{l,f}+\lambda_{r,f})(\mathbf{n}_{pf}\otimes\mathbf{n}_{pf}),\quad \mathbf{w}_p=\sum_{f\in\mathcal{SF}(p)} l_{pf}(\lambda_{l,f}+\lambda_{r,f})\overline{u}_{\mathbf{n}_{pcf}}\mathbf{n}_{pf}. \tag{5.3.30}
$$

This system is non other than the Lagrangian **nodal solver** that provides an approximation of the nodal velocity $\mathbf{v}_p$.

5.3.4 Corresponding Eulerian simple Riemann solver

Let us deduce the corresponding Eulerian simple Riemann solver. Relying on hypothesis (H_1) we can deduce the Eulerian wave speeds Λ_l, Λ_0 and Λ_r from their Lagrangian counterparts λ_l, $\lambda_0 = 0$ and λ_r as

$$\Lambda_l = u_{\mathbf{n},l} - \lambda_l \tau_l = u^\star_{\mathbf{n}} - \lambda_l \tau^\star_l, \qquad \Lambda_0 = u^\star_{\mathbf{n}}, \qquad \Lambda_r = u^\star_{\mathbf{n}} + \lambda_r \tau^\star_r = u_{\mathbf{n},r} + \lambda_r \tau_r. \quad (5.3.31)$$

Provided that the Lagrangian approximate Riemann solver preserves the positivity of specific volumes, *i.e.*, $\tau^\star_s \geq 0$, then these Eulerian wave speeds are ordered: $\Lambda_l \leq \Lambda_0 \leq \Lambda_r$. The positivity property holds true provided that the Lagrangian wave speeds satisfy condition (5.3.17). The Eulerian approximate Riemann solver is deduced from its Lagrangian counterpart as follows

$$\mathbf{W}_E\left(\mathbf{U}_L, \mathbf{U}_r, \frac{x_{\mathbf{n}}}{t}\right) = \begin{cases} \mathbf{U}_l & \text{if} \quad \frac{x_{\mathbf{n}}}{t} \leq \Lambda_l, \\ \mathbf{U}^\star_l = \mathbf{U}(\mathbf{V}^\star_l) & \text{if} \quad \Lambda_l < \frac{x_{\mathbf{n}}}{t} \leq \Lambda_0, \\ \mathbf{U}^\star_r = \mathbf{U}(\mathbf{V}^\star_r) & \text{if} \quad \Lambda_0 < \frac{x_{\mathbf{n}}}{t} \leq \Lambda_r, \\ \mathbf{U}_r & \text{if} \quad \Lambda_r < \frac{x_{\mathbf{n}}}{t}. \end{cases}$$

Here, $\mathbf{V} \mapsto \mathbf{U}(\mathbf{V})$ is the Lagrange-to-Euler mapping defined by

$$\mathbf{V} = \tau(\mathbf{U} - h\mathbf{e}_1) + \tau \mathbf{e}_1, \qquad \text{where} \quad \mathbf{e}_1 = (1,0,0)^t.$$

The Eulerian intermediate states read $\mathbf{U}^\star_s = (h^\star_s, h^\star_s u^\star_{\mathbf{n}}, h^\star_s u^\star_{\mathbf{t},s})^t$ knowing that $h^\star_s = (\tau^\star_s)^{-1}$ for $s = l, r$. We observe that the Eulerian approximate Riemann solver is also parametrized by the normal star-velocity. More importantly, by construction, the Eulerian approximate Riemann solver has the same properties as its Lagrangian counterpart.

In the development of the simple Lagrangian solver, we stated that the contribution of the bathymetry, $\overline{\mathbf{P}}_{\mathbf{n},s} = (0, \overline{q}_{\mathbf{n},s}, 0)^t$ is an appropriate mean value of the source term for $s = l, r$. To determine this term we enforce that the simple Lagrangian Riemann solver verifies the steady state Lake at Rest (LaR) solution. Indeed, let us assume that the left and right states verify $\dfrac{1}{\tau_s} + B_s = \text{constant}$ and $u_{\mathbf{n},s} = u_{\mathbf{t},s} = 0$ for $s = l, r$. Under this lake at rest solution and considering (5.3.21) we have

$$\overline{u}_{\mathbf{n}_{pcf}} = -\frac{(\widetilde{p}_{r,f} - \widetilde{p}_{l,f})}{\lambda_{r,f} + \lambda_{l,f}} = -\frac{(p_{r,f} + \overline{q}_{\mathbf{n}_{pcf},r} - p_{l,f} - \overline{q}_{\mathbf{n}_{pcf},l})}{\lambda_{r,f} + \lambda_{l,f}}. \quad (5.3.32)$$

It is natural to expect that the nodal solver produces $\mathbf{u}_p = \mathbf{0}$ to respect the lake at rest solution. To obtain such a solution then, for all face f impinging at point p, we must retrieve $\overline{u}_{\mathbf{n}_{pcf}} = 0$, see system (5.3.29). In other words

$$p_{r,f} + \overline{q}_{\mathbf{n}_{pcf},r} - p_{l,f} - \overline{q}_{\mathbf{n}_{pcf},l} = 0. \quad (5.3.33)$$

Moreover, if $\mathbf{u}_p = \mathbf{0}$, then $\mathbf{u}_p \cdot \mathbf{n}_{pf} \equiv u^\star_{\mathbf{n}_{pf}} = 0$ for all f impinging at point p, then systems

$(\mathcal{S}_l)$ and $(\mathcal{S}_r)$ reduce to $\tau^\star_{s,f} = \tau_{s,f}$, and, $\overline{p}_{s,f} - p_{s,f} = \overline{q}_{\mathbf{n}_{pcf},s}$, for $s = l, r$. The mean value of the source term can be approximated by

$$-\overline{q}_{\mathbf{n}_{pcf},l} = \overline{q}_{\mathbf{n}_{pcf},r} = \frac{1}{2}\frac{1}{\overline{\tau}_f}\left(B_r - B_l\right), \tag{5.3.34}$$

where the derivative of B in the normal direction $\mathbf{n}_{pcf}$ is assumed to be $B_r - B_l$, and a mean value of τ at face f remains to be defined. Recalling that $p = p(\tau) = \frac{1}{2}\frac{1}{\tau^2}$ and considering the lake at rest solution, $B = \text{constant} - \frac{1}{\tau}$, so that, $B_r - B_l = -\frac{1}{\tau_r} + \frac{1}{\tau_l}$, then we rewrite (5.3.33) as

$$\frac{1}{2}\left(\frac{1}{\tau_{r,f}} + \frac{1}{\tau_{l,f}}\right)\left(\frac{1}{\tau_{r,f}} - \frac{1}{\tau_{l,f}}\right) - \frac{1}{\overline{\tau}_f}\left(\frac{1}{\tau_{r,f}} - \frac{1}{\tau_{l,f}}\right) = 0, \tag{5.3.35}$$

which is true in the general case if we assume

$$\frac{1}{\overline{\tau}_f} = \frac{1}{2}\left(\frac{1}{\tau_{r,f}} + \frac{1}{\tau_{l,f}}\right), \quad \text{or}, \quad \overline{h}_f = \frac{1}{2}\left(h_{r,f} + h_{l,f}\right). \tag{5.3.36}$$

Notice that $\overline{h}_f$ is a coherent approximation of the water height at the interface f between states h_r and h_l.

Wave speeds Similarly to the study done in chapter 4 for the positivity of state variables, the positivity of the water heights can be ensured with a proper explicit condition on the Lagrangian wave speeds. Here, we mimic the Dukowicz (1984) approximation which is a convenient manner of ensuring the positivity of the specific volume for the Lagrangian Riemann solver. In this case, the Dukowicz approximation for gas dynamics with perfect gas law (with $\gamma = 2$ in our case) is defined by

$$\lambda_l = \frac{a_l}{\tau_l}\left[1 - \mathcal{G}\frac{u^\star_{\mathbf{n}} - u_{\mathbf{n},l}}{a_l}\right], \qquad \lambda_r = \frac{a_r}{\tau_r}\left[1 + \mathcal{G}\frac{u^\star_{\mathbf{n}} - u_{\mathbf{n},r}}{a_r}\right], \tag{5.3.37}$$

with $\mathcal{G} = \dfrac{\gamma+1}{2} = \dfrac{3}{2} > 1$. Consequently, we can ensure the positivity of the water heights, and the left and right discontinuities using (5.3.37) give

$$\lambda_l = \frac{a_l}{\tau_l} + \mathcal{G}\left(-\frac{u^\star_{\mathbf{n}} - u_{\mathbf{n},l}}{\tau_l}\right) \geq -\frac{u^\star_{\mathbf{n}} - u_{\mathbf{n},l}}{\tau_l}, \qquad \lambda_r = \frac{a_r}{\tau_r} + \mathcal{G}\left(\frac{u^\star_{\mathbf{n}} - u_{\mathbf{n},r}}{\tau_r}\right) \geq \frac{u^\star_{\mathbf{n}} - u_{\mathbf{n},r}}{\tau_r}, \tag{5.3.38}$$

which is nothing but the positivity conditions derived in (5.3.17).

Remark 5.3.1. *The matrix and the right-hand-side of the nodal solver depend on the wave-speeds in a non-linear way. Practically only one step of a fixed-point Newton method is computed with the initial wave speeds* $\lambda_s = \dfrac{a_s}{\tau_s}$ *for* $s = l, r$ *in the nodal solver. We have*

experienced that two steps of this Newton method does not bring any obvious advantage in terms of accuracy, while increasing the computational cost.

5.3.5 Summary of the Eulerian multidimensional Finite Volume scheme

We recall that the multidimensional Finite Volume scheme writes

$$\mathbf{U}_c^{n+1} - \mathbf{U}_c^n + \frac{\Delta t}{|\omega_c|} \sum_{p\in\mathcal{P}(c)} \sum_{f\in\mathcal{SF}(pc)} l_{pcf} \overline{\mathbf{F}}_{pcf} = \Delta t \sum_{f\in\mathcal{F}(c)} \overline{\mathbf{S}}_f, \tag{5.3.39}$$

where $\overline{\mathbf{F}}_{pcf}$ is the left-sided flux with respect to the subface f and the unit outward normal $\mathbf{n}_{pcf}$. The left-sided flux between cells c and $d \equiv d(c,f)$ (notice that $c \equiv c(d,f)$) for a simple Eulerian Riemann solver is

$$\overline{\mathbf{F}}_{pcf} \equiv \overline{\mathbf{F}}^-_{\mathbf{n}_{pcf}} = \mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_c) - \left[\sum_{k=1}^{m} \Lambda_k^{(-)} (\mathbf{U}_{k+1} - \mathbf{U}_k) \right]_{c,d}, \tag{5.3.40}$$

while the right-sided flux writes

$$\overline{\mathbf{F}}^+_{\mathbf{n}_{pcf}} = \mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_d) - \left[\sum_{k=1}^{m} \Lambda_k^{(+)} (\mathbf{U}_{k+1} - \mathbf{U}_k) \right]_{c,d}. \tag{5.3.41}$$

Recall that the Eulerian Riemann solver is deduced from its Lagrangian counterpart that allows to compute the $m = 4$ states that are separated by 3 discontinuities

Classical face-based Finite Volume method with two-point flux Taking the arithmetic average of the left and the right-sided fluxes allows us to define the averaged flux on the subface f

$$\overline{\mathbf{F}}_{\mathbf{n}_{pcf}} = \frac{1}{2} \left[\mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_c) + \mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_d) \right] - \frac{1}{2} \left[\sum_{k=1}^{m} |\Lambda_k| (\mathbf{U}_{k+1} - \mathbf{U}_k) \right]_{c,d}. \tag{5.3.42}$$

This flux is common in magnitude to face f but applied with a plus (resp. a minus) sign in cell c (resp. d). The conservation is simply retrieved by cancellation when summation over the cells is invoked.

Node-based Finite Volume method with multi-point flux The fundamental property demonstrates that the difference between the right and left-sided subface fluxes in the Eulerian framework is equivalent to the Lagrangian one, leading to

$$\overline{\mathbf{F}}^+_{\mathbf{n}_{pcf}} - \overline{\mathbf{F}}^-_{\mathbf{n}_{pcf}} = \overline{\mathbf{G}}^+_{\mathbf{n}_{pcf}} - \overline{\mathbf{G}}^-_{\mathbf{n}_{pcf}} = (\overline{p}_d - \overline{p}_c) \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}. \tag{5.3.43}$$

Combining the foregoing equation with (5.3.42) we arrive at

$$\begin{aligned}\overline{\mathbf{F}}^{-}_{\mathbf{n}_{pcf}} =& \frac{1}{2}[\mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_c) + \mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_d)] - \frac{1}{2}\left[\sum_{k=1}^{m} |\Lambda_k|(\mathbf{U}_{k+1} - \mathbf{U}_k)\right]_{c,d} \\ & - \frac{\lambda_c + \lambda_d}{2}\left[\mathbf{u}_p \cdot \mathbf{n}_{pcf} - \overline{u}_{\mathbf{n}_{pcf}}\right]\begin{pmatrix}0\\1\\0\end{pmatrix}.\end{aligned} \tag{5.3.44}$$

This explicit expression of the numerical flux shows that in the unlikely event $\mathbf{v}_p \cdot \mathbf{n}_{pcf} = \overline{u}_{\mathbf{n}_{pcf}}$ we retrieve the classical conservative face-based Finite Volume scheme. In general, there is no reason for this to happen. We recall that the value of $\mathbf{v}_p$ is the approximate solution of the nodal solver (5.3.29), while $\overline{u}_{\mathbf{n}_{pcf}}$ is explicitly given by (5.3.21). Note that the positivity of water height is ensured provided that the wave speeds are computed with (5.3.37), and that the lake at rest solution is retrieved provided that the source term contribution is considered as in (5.3.34) by the Riemann solver.

Source term treatment To complete the scheme (5.3.7), it remains to express the source term $\overline{\mathbf{S}}_f$. We recall that the discrete source term at the right-hand side approximates explicitly

$$\frac{\Delta t}{|\omega_c|}\int_{\omega_c} \mathbf{S}(x,y,t^n)\,dv \simeq \frac{\Delta t}{|\omega_c|}\int_{\omega_c} -g\begin{pmatrix}0\\ h\partial_x B\\ h\partial_y B\end{pmatrix}(x,y,t^n)\,dv. \tag{5.3.45}$$

Again we make the fundamental hypothesis of enough regularity to ensure that these integrals and products are well defined. The well-balanced discretization of the source term must ensure that the lake at rest steady-state solution be exactly preserved. Previously we have shown that the intermediate states of the Riemann solver are already well-balanced. As such the discrete source term must balance the remaining terms from the fluxes which are

$$\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} l_{pcf}\overline{\mathbf{F}}_{pcf} = \sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} l_{pcf}\frac{1}{2}\left[\mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_c) + \mathbf{F}_{\mathbf{n}_{pcf}}(\mathbf{U}_d)\right]. \tag{5.3.46}$$

More precisely, in the case of a steady state lake at rest solution we expect that

$$\mathbf{U}_c^{n+1} = \mathbf{U}_c^n$$

where $\mathbf{U}_c^n = (h_c^n, 0, 0)$ and $h_c^n + B_c = \text{constant} > 0$. Due to the fact that the velocity is null as well as the source term for the first component of $\mathbf{U}$, we have directly $h_c^{n+1} = h_c^n$. For the non-trivial second and third components, we first can revamp the double sums

into a single sum on the faces as:

$$\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)}\frac{1}{2}(p_d+p_c)\,l_{pcf}\mathbf{n}_{pcf}\equiv\sum_{f\in\mathcal{F}(c)}\frac{1}{2}(p_d+p_c)\,l_f\mathbf{n}_f, \tag{5.3.47}$$

where l_f, $\mathbf{n}_f$ are the face length and unit outward pointing normal respectively. Recalling the geometrical relation on a closed contour $\sum_{f\in\mathcal{F}(c)} l_f\mathbf{n}_f = \mathbf{0}$, we have the following equalities

$$\sum_{f\in\mathcal{F}(c)}\frac{1}{2}(p_d+p_c)\,l_f\mathbf{n}_f=\sum_{f\in\mathcal{F}(c)}\frac{1}{2}p_d\,l_f\mathbf{n}_f+\frac{1}{2}p_c\underbrace{\sum_{f\in\mathcal{F}(c)}l_f\mathbf{n}_f}_{=\mathbf{0}}=\sum_{f\in\mathcal{F}(c)}\frac{1}{2}(p_d-p_c)\,l_f\mathbf{n}_f. \tag{5.3.48}$$

Next, following the one-dimensional case we assume that the source term can be expressed as

$$\sum_{f\in\mathcal{F}(c)}\overline{\mathbf{S}}_f=\sum_{f\in\mathcal{F}(c)}-\frac{g}{2}h_f\,(B_d-B_c)\,l_f\mathbf{n}_f, \tag{5.3.49}$$

where h_f is an approximation of the averaged water height at the face f. At last in order to fulfill the well-balanced property, we must ensure that

$$\sum_{f\in\mathcal{F}(c)}\frac{1}{2}(p_d-p_c)\,l_f\mathbf{n}_f=\sum_{f\in\mathcal{F}(c)}-\frac{g}{2}h_f\,(B_d-B_c)\,l_f\mathbf{n}_f. \tag{5.3.50}$$

A sufficient condition to fulfill (5.3.50) yields that for each face f between cells c and d one verifies

$$\frac{1}{2}(p_d-p_c)=-\frac{g}{2}h_f\,(B_d-B_c), \tag{5.3.51}$$

which, considering the case of a steady state lake at rest solution (B = constant $-$ h), allows to define the face based water height as

$$h_f=\frac{p_d-p_c}{g(h_d-h_c)}=\frac{h_d+h_c}{2}. \tag{5.3.52}$$

Therefore, the source term approximated by

$$\sum_{f\in\mathcal{F}(c)}\overline{\mathbf{S}}_f=\sum_{f\in\mathcal{F}(c)}-\frac{g}{2}h_f\,(B_d-B_c)\,l_f\mathbf{n}_f, \tag{5.3.53}$$

yields a well-balanced Finite Volume numerical scheme by construction.

Remark 5.3.2. *Similarly to the one-dimensional case, if the bathymetry is flat then the source term vanishes as expected. Whereas if the water height is constant in space, $h_f = h$, then the source term in the cell yields $-\frac{g}{2}h\sum_{f\in\mathcal{F}(c)}(B_d-B_c)\,l_f\mathbf{n}_f$, which is indeed a consistent definition of a discrete gradient of B.*

Remark 5.3.3. *Notice that we could have used a different technique for the well-balanced*

property based on the reconstruction of the steady states Castro et al. *(2008, 2020). First the Riemann solver developed for the homogeneous system (Euler isentropic* Chan et al. *(2021)) is used, which de facto does not produce well-balanced intermediate states. But then the source term is discretized with a reconstruction technique in such a way that the source term discretization compensates exactly with the pressure term in the case of Lake at Rest solution.*

By construction, the proposed two-dimensional numerical scheme associated to the Eulerian simple Riemann solver and the discrete source term described in this section is well-balanced as it ensures that lake at rest solution is retrieved by construction. It remains to determine the CFL condition for the two-dimensional case that ensures positivity and entropy stability. A thorough theoretical study is required and this will be done in the future work. Let us proceed to the numerical validation of this scheme on a classical test suite for two-dimensional shallow-water equations.

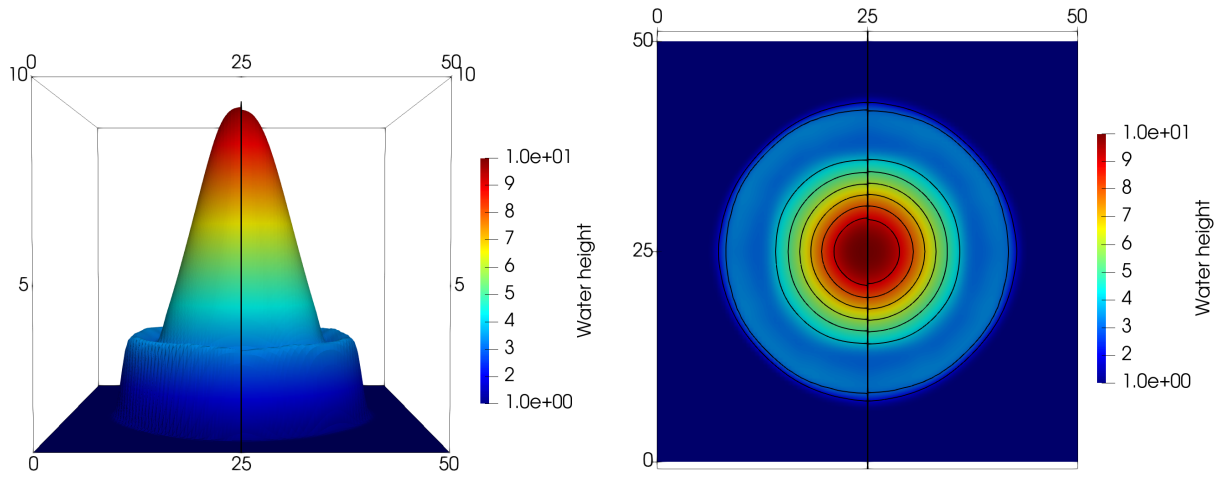
5.3.6 Two-dimensional numerical validation

In this section we present several classical test cases to assess the properties of the both two-point and multi-point schemes. The main purpose is not only observe the well-balanced and positivity-preserving property of the schemes, but also to study the difference between the two-point and multi-point schemes in terms of shock-generated spurious instabilities. The CFL is set to 0.5 and the gravity constant $g = 9.81$ unless otherwise stated.

Radial dam break The first benchmark test case is the circular dam-break problem on flat bottom from Alcrudo & Garcia-Navarro (1993). This problem involves the breaking of a circular dam allowing to verify the performance of the algorithm when solving symmetrical flow problems. The computational domain is a square $\Omega \in [0, 50] \times [0, 50]$ separating two regions by a cylindrical wall of radius $r_0 = 11$m, for which the initial conditions are

$$h = \begin{cases} 10 \text{ m} & \text{if } r \leq r_0, \\ 1 \text{ m} & \text{if } r > r_0, \end{cases} \qquad u = v = 0\text{m}^2/\text{s}, \qquad r = \sqrt{(x-25)^2 + (y-25)^2}.$$

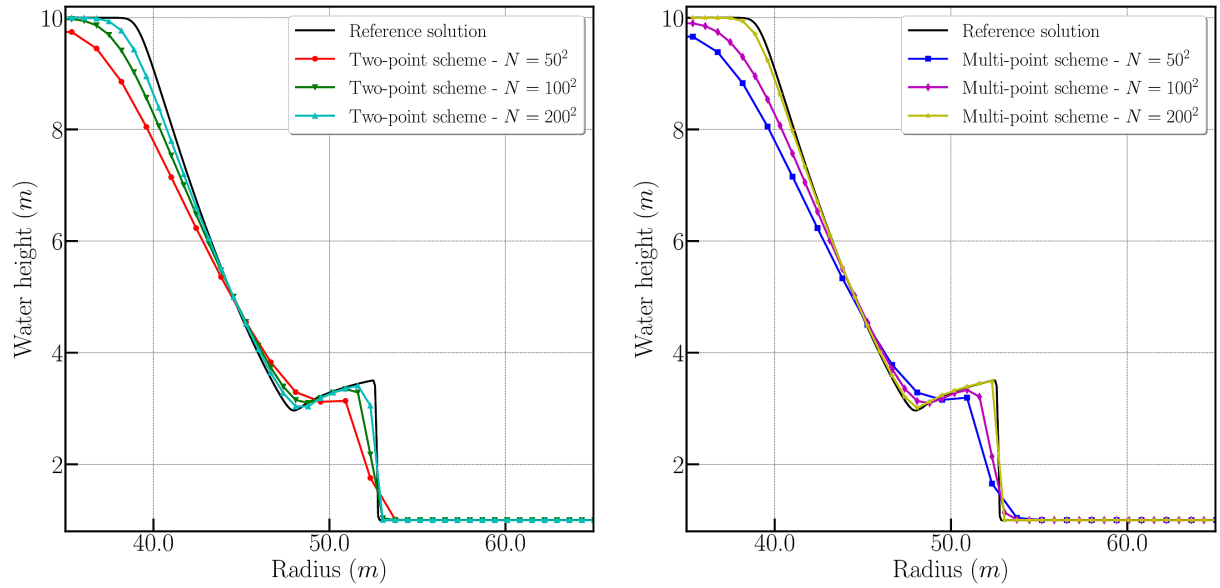
When the water starts to drain at $t = 0$, the circular dam-break bore waves will spread and propagate radially and symmetrically. The final solution is plotted at $t_{final} = 0.69s$. Figure 5.8(a) shows the 3D view of the water height h on a 50×50 structured and regular grid with the two-point scheme results on the left side and the multi-point one on the right. Likewise the contours of water height from $h = 1$m to $h = 10$m are illustrated in figure 5.8(b). Figures 5.9 is a mesh convergence of the water height along the radius on $N = 50^2, 100^2$ and 200^2, compared to a reference solution for the two-point scheme in 5.9(a) and the multi-point scheme in 5.9(b). For both schemes, the waves propagate uniformly and symmetrically. Next, the same simulation is ran on 6630 triangular cells and the results are presented in figure 5.10. The same conclusions occur.



(a) 3D view of water height h.

(b) 10 iso-lines of $1\text{m} < h < 10\text{m}$.

Figure 5.8: Circular Dam break — Water height at $t = 0.69s$ of the two-point (left) and multi-point (right) schemes on 50×50 grid.



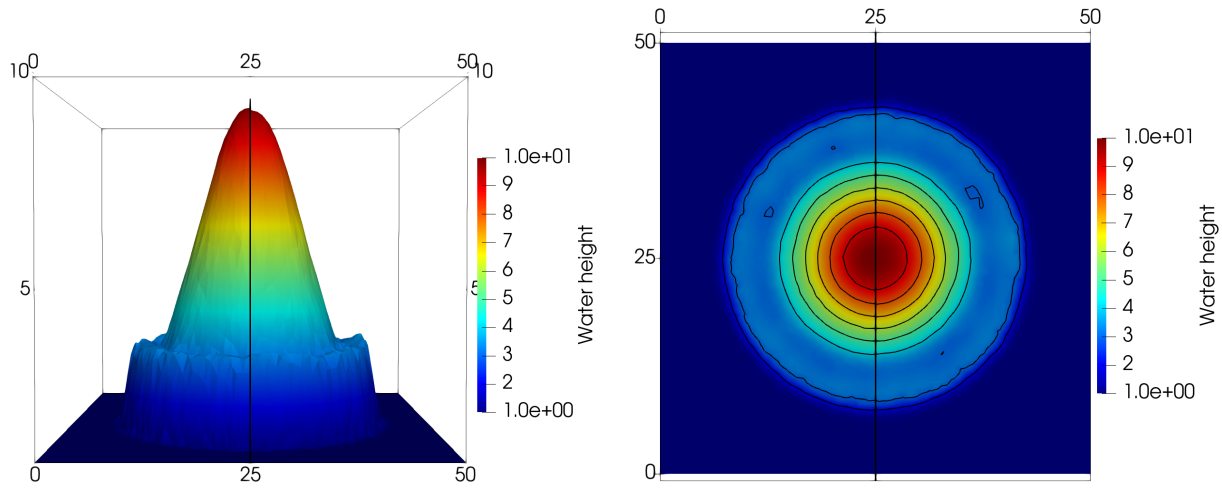
(a) Two-point scheme.

(b) Multi-point scheme.

Figure 5.9: Circular Dam break — Mesh convergence of the water height along the radius compared to a reference solution (black line).

Oblique hydraulic jump The oblique hydraulic jump is induced by means of an interaction between a super-critical flow and a converging wall deflected through an angle $\theta = 8.95°$, see Alcrudo & Garcia-Navarro (1993). The computational domain is set to $\Omega \in [0, 40] \times [0, 30]$. The initial conditions is a uniform flow given by

$$h = 1\text{m}, \quad u = 8.57\text{m/s}, \quad v = 0\text{m/s}.$$



(a) 3D view of water height h.

(b) 10 iso-lines of 1m $< h <$ 10m.

Figure 5.10: Circular Dam break — Water height at $t = 0.69s$ of the two-point (left) and multi-point (right) schemes on 6630 triangular cells.

The boundary conditions are set to super-critical flows upstream and downstream. The exact solution is given by $h = 1.5$m, $u = 7.9556$m/s for the downstream flow and an angle of 30° for the angle of the jump connecting down to upstream flow. The final time is set to $t_{final} = 10$s and the mesh is made of $N = 11892$ triangles.

In figure 5.11 we present the numerical water height obtained with the two- and multi-point schemes (top and bottom respectively) in color and for 5 isolines (1m $< h <$ 1.5m). Next in figure 5.12(a) are compared the water height along the line $y = 40$m with respect to the exact solution showing that the correct solution is captured by both schemes while the two-point seems slightly less dissipative. At last in figure 5.12(b) we present the comparison between the first and second order scheme showing an increase in the accuracy with which the discontinuity is captured.

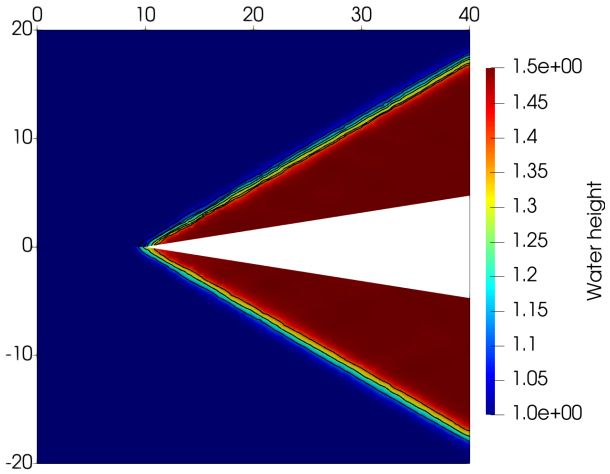
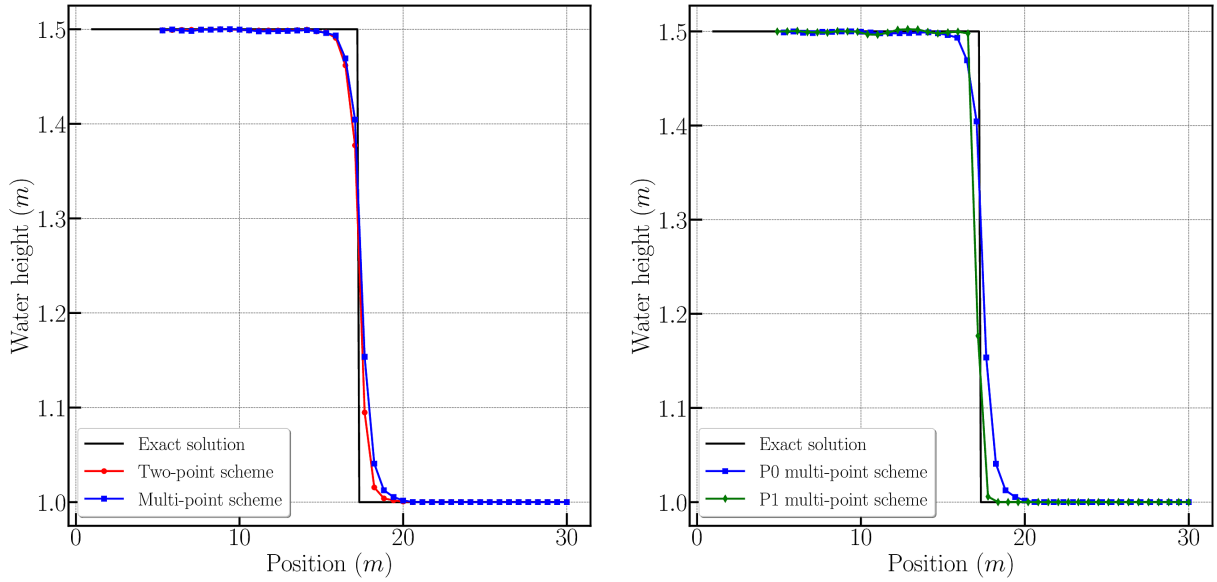


Figure 5.11: Oblique hydraulic jump — Numerical water height at $t_{final} = 10s$ on 11892 unstructured grids: Two-point (top) vs multi-point (bottom) water height in color and 5 iso-lines ($1 < h < 1.5$).



(a) Water heights along $y = 40$m. (b) 1st and 2nd order multi-point schemes.

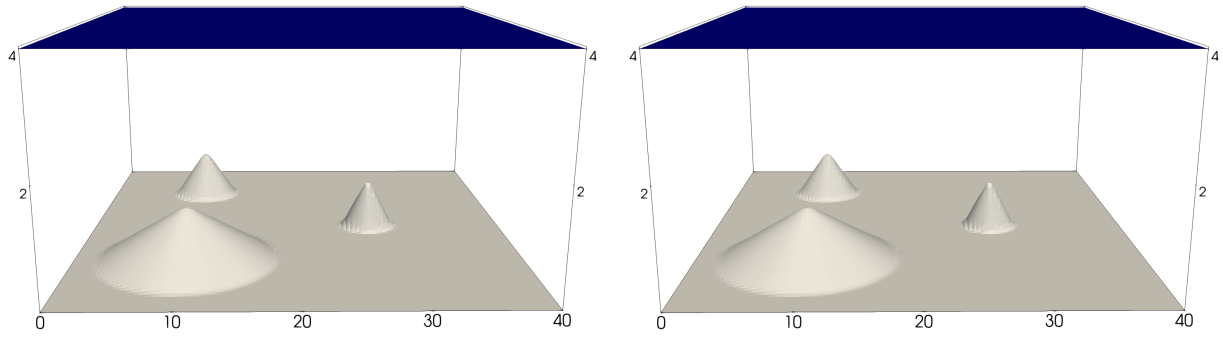
Figure 5.12: Oblique hydraulic jump — Comparison of numerical density with the exact solution (black line) along $y = 40m$.

Flow over three mounds This test case has a complex bottom configuration and allows to verify the well-balanced property of the schemes. The bottom configuration for a domain $\Omega = [0, 40\text{m}] \times [0, 40\text{m}]$ is defined by the function:

$$B(x, y) = \max\left[0,\ 1 - \frac{1}{8}R_{10,11},\ 1 - \frac{3}{10}R_{10,31},\ 1 - \frac{4}{10}R_{27,20}\right],\ R_{a,b} = \sqrt{(x-a)^2 + (y-b)^2}.$$

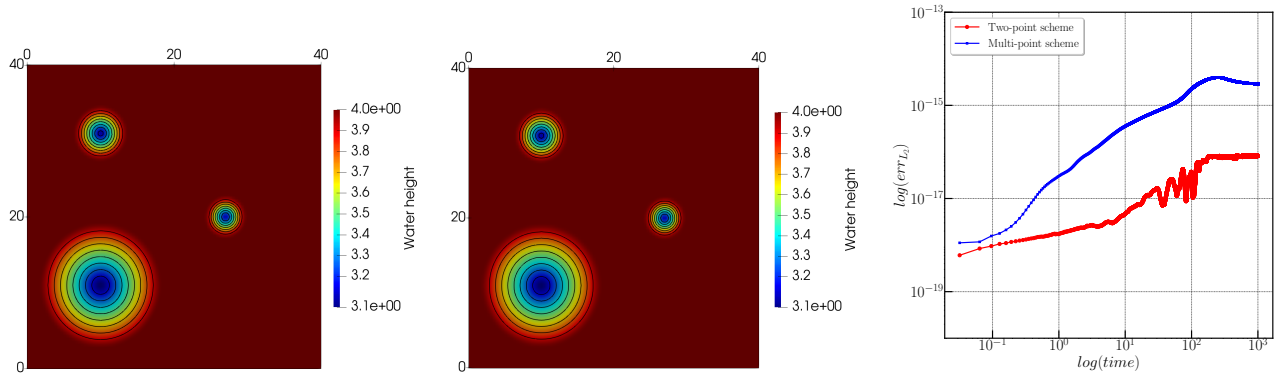
We initialize a lake at rest solution, that is the initial conditions are $h + B = 4$m and $u = v = 0$m/s and the final time is $t = 200$s. The mesh is constituted of 100×100 uniform squares and periodic boundary conditions are considered.
The free surface obtained by the two-point and multi-point schemes are presented in figure 5.13 along with the bathymetry. We clearly see that the initial state for both schemes is exactly preserved. The water height with 10 isolines from $h^{\min} = 3.15$m to $h^{\max} = 4$m are shown in figures 5.14(a) and 5.14(b) respectively, proving that the schemes reproduce the initial (and exact) solution. The L_2 error for the water height for both schemes are then presented in 5.14(c) in log scale to demonstrate that the still water steady state is maintained. Notice that the WB property is obtained independently of the bathymetry B.

Modified dry bed generation This test case simulates the generation of a dry bed on a non-flat bathymetry. While the test is 1D, we run it in 2D on a Cartesian grid. The



(a) Two-point scheme. (b) Multi-point scheme.

Figure 5.13: Flow over three mounds — Numerical solution at $t = 200s$ by the two-point and multi-point schemes on a 100×100 square grid, and log of error versus log of time.



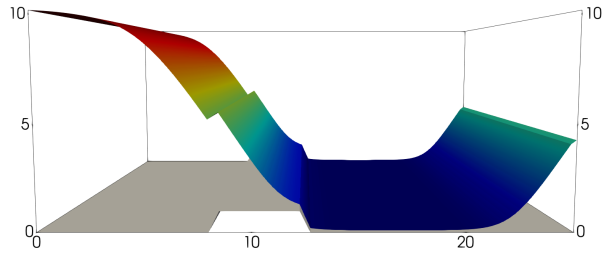
(a) Two-point scheme. (b) Multi-point scheme. (c) Error vs time.

Figure 5.14: Flow over three mounds — Isolines of h at $t = 200$s of both the two-point and multi-point schemes and plot of error *vs* time.

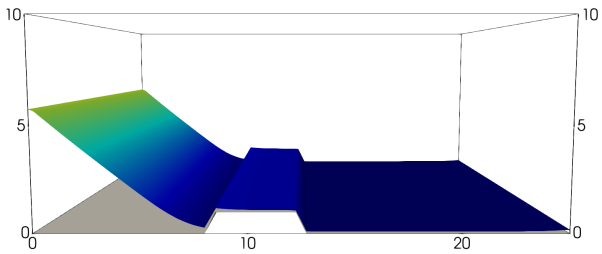
initial conditions on domain $\Omega = [0, 30] \times [0, 10]$ are set to

$$h = 10\text{m}, \quad u = \begin{cases} -35\text{m/s} & \text{if } x > 50/3 \\ 35\text{m/s} & \text{otherwise} \end{cases}, \quad v = 0\text{m/s}. \tag{5.3.54}$$

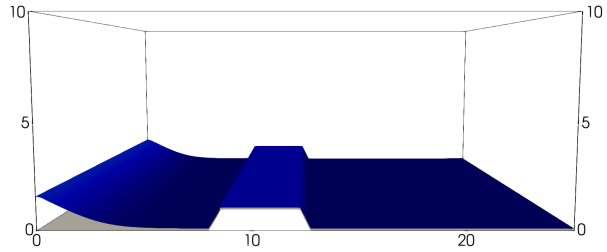
The step bathymetry is set to $B = 1$m if $25/3 < x < 25/2$ and $B = 0$m otherwise, see the gray shape in figure 5.15. Outflow boundary conditions are set on the left/right boundaries while periodic ones are considered on top/bottom. The velocity field creates two rarefaction waves moving in opposite directions, the left one further interacting with the bottom step and later escaping through the boundaries, as such drying the domain. The final time is set to $t = 1$s and several intermediate times are observed: $t = 0.05$s, 0.25s, 0.45s and 0.65s. The mesh is made of 100×100 quadrangles. In figure 5.15 we present the free surface elevation $h + B$ for the intermediate times. The numerical results of the multi-point scheme are in agreement with classical ones. Due to the fact that our scheme does not allow for a strict $h = 0$, then a small film of water always remains on the step and the bottom surface. Nonetheless no spurious effect is observed.



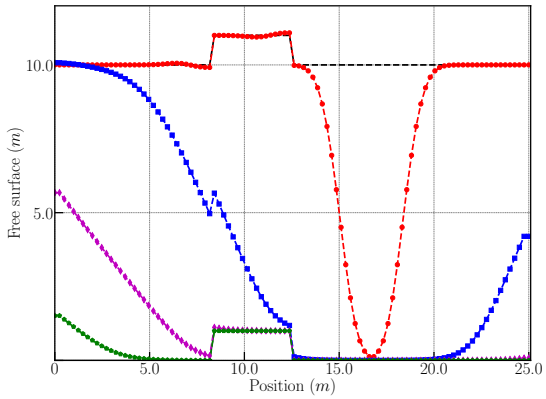
(a) $t = 0.25s$.



(b) $t = 0.45s$.



(c) $t = 0.65s$.



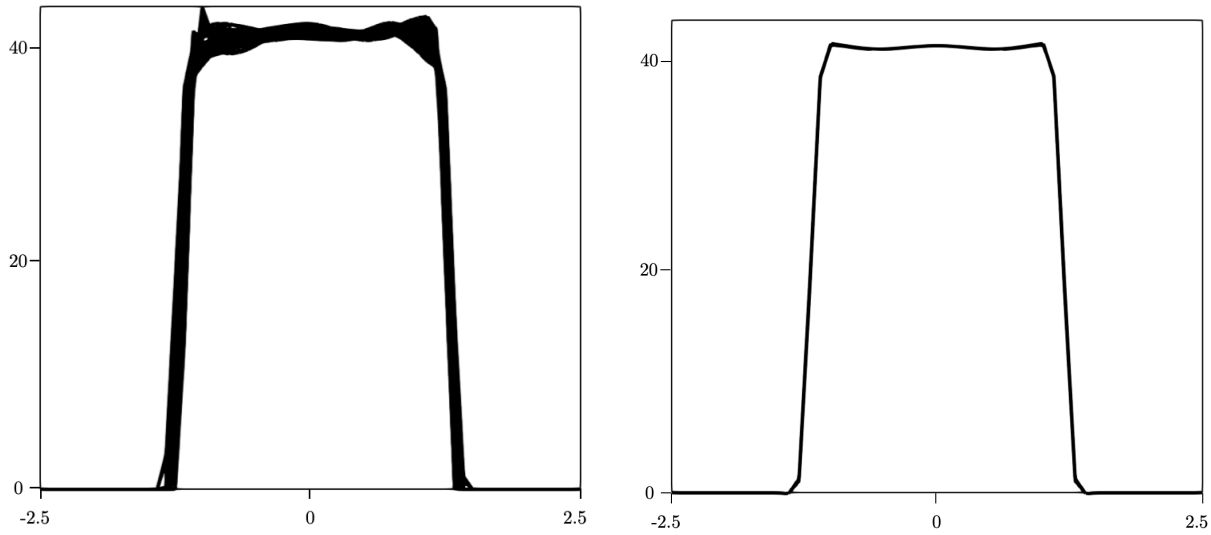
(d) Free surface $h + B$ for the multi-point scheme.

Figure 5.15: Modified dry bed generation — Numerical solution at $t = 0.05$s, 0.25s, 0.45s and 0.65s on 100×100 grid.

Colliding flow Next we test the sensibility of the numerical schemes towards shock instabilities, i.e. carbuncle. The first test case is a colliding flow on a flat bottom by Kemm (2018). The computational domain of this test case is $\Omega \in [-2.5\text{m}, 2.5\text{m}] \times [-2.5\text{m}, 2.5\text{m}]$ and we set the gravitational constant to $g = 1$. The initial condition is given as

$$h = 1m, \quad u = \begin{cases} 30\text{m/s} & \text{if} \quad x \leq 0, \\ -30\text{m/s} & \text{if} \quad \text{else.} \end{cases}, \quad v = 0\text{m/s}.$$

A small perturbation of the form $\delta(2z - 1)$, where z is a random number, $z \in [-1, 1]$ and $\mathbf{u} = \mathbf{u} + \delta(2 \cdot \text{RND} - 1)$ is introduced to the initial state variables with $\delta = 10^{-5}$ in order to trigger an instability. The domain is paved by 50×50 uniform quadrangular cells. Because the test is essentially 1D in x direction, this perturbation should not impact the numerical schemes.
In figure 5.16 we plot the scatter plot of the water height h, at time $t = 2$s for both two-point and multi-point schemes. Clearly the two-point scheme produces spurious instabilities whereas the multi-point scheme displays a symmetrical solution. This spurious instability is referred to as the 'carbuncle effect' Kemm (2018).



(a) Two-point scheme. (b) Multi-point scheme.

Figure 5.16: Colliding flow — Numerical water height at $t = 2$s for the two-point (left) and multi-point (right) schemes and a 50×50 quad grid.

5.4 Chapter summary

This chapter has presented an extension of the original numerical scheme for one- and two-dimensional shallow water balanced laws. The main characteristics of this numerical method is its direct construction based on the Lagrangian counterpart, allowing a more explicit strategy to assure good properties of a numerical scheme, including an appropriate wave speeds ordering and positivity preservation. This method in the one-dimensional case was established in Gallice (2002*a*) and reinterpreted in this present work. It is also constructed to assure that the steady-state solutions (Lake at Rest) are exactly retrieved by the Riemann solver, in other words the Riemann solver is well-balanced. Using this as a building block we construct a non-conservative Finite Volume scheme. The Lagrange-to-Euler mapping then allows to preserve and transfer these properties to the Eulerian ones and the conservation condition of this scheme is further retrieved on a node-based fashion. Several numerical results in both one- and two-dimensional are presented to assess the behavior of both two-point and multi-point numerical method. The multi-point scheme seems once again insensitive to the so-called carbuncle effect. The oblique hydraulic jump test case also shows that the second order scheme produces results with better accuracy as expected.
This work has been developed under the fundamental hypothesis that the domain is, and remains, wet. It remains to demonstrate rigorously the positivity-preserving and entropy stability properties of the numerical method under the constraint of a finite time step. A possible improvement would be to treat wet-dry transition area which would demand some technical adjustments on the implementation level. A second line of improvement is the extension to higher orders of accuracy using polynomial reconstructions and *a*

posteriori MOOD type limiting procedure, refer to Clain & Figueiredo (2017); Chan *et al.* (2021).

6 Conclusion

The main objective of this thesis is to develop a numerical scheme for multidimensional supersonic/hypersonic aerodynamics on unstructured mesh. Supersonic aerodynamics are flows with strong shock and rarefaction waves. Therefore, we aim at building a robust and precise numerical method in order to capture these waves correctly. The numerical method addressed is the Godunov-type Finite Volume method, associated with this scheme is a Riemann solver. In this thesis, the mathematical model used is the Euler equations and the cornerstone of this work is to forge a link between the Lagrangian and Eulerian framework, mainly inspired by the seminal work of Gallice (2001, 2002*a*, 2003), and also the work Shen *et al.* (2014) that was used as a starting point into developing the numerical scheme presented in this work.

The first stage of this project consists of building a robust scheme for one-dimensional system of conservation laws. The main feature of the one-dimensional simple approximate Riemann solver developed in this work lies on its properties: positivity-preserving, entropy stable and an explicit condition on the discontinuity velocities. To achieve these properties, the solver is first developed in the Lagrangian framework, offering a solver that preserves contact discontinuity. The positivity-preserving property and entropy control is accomplished by deriving explicit conditions on the Lagrangian wave speeds. Once the Lagrangian Riemann solver is validated, the Eulerian counterpart is established thanks to the Lagrange-to-Euler mapping. By construction, the Eulerian simple approximate Riemann solver inherits the appropriate properties from its Lagrangian counterpart. This solver is the cornerstone to build a positive and entropic Godunov-type scheme, provided that the time step condition is fulfilled. This scheme is then adopted as the parachute scheme for the high-order extension with MOOD limiting strategy, see Chan *et al.* (2021).

Following the one-dimensional study, we proceed to the multidimensional case. The multidimensional study is separated into two parts. The first part consists of the theoretical framework of the multidimensional numerical scheme. An original Godunov-type Finite Volume scheme for general hyperbolic systems of conservation laws on unstructured grids is demonstrated. The key feature of this method is that the discretization is subface-based, contrastingly to classical multidimensional Finite Volume schemes that are face-based. In this thesis, the subface-based discretization is named the multi-point scheme, as opposed of the classical discretization known as the two-point scheme. Suc-

cinctly, a generic mesh will be partitioned to subcells and the numerical flux approximation is subface-based. The subface flux is obtained from an approximate Riemann solver. A node-based sufficient condition is established on all nodes to ensure conservativity of the scheme. The scheme is considered $\mathcal{D}$-preserving with a specific time step condition. Then, implementing the same strategy and following Gallice (2003)'s work, the simple approximate Riemann solver is first built in the Lagrangian framework. The Lagrangian solver is then used as a building block for the Eulerian counterpart, thanks to the Lagrange-to-Euler mapping, (H_1) and (H_2) hypothesis and the fundamental relations that allow to obtain the Eulerian numerical fluxes from the Lagrangian ones.

After establishing the theoretical framework of the multidimensional scheme, the second part consists of the applications of the scheme. For an elementary demonstration, the multidimensional scheme is incorporated to the two-dimensional gas dynamics equations in the first place. By doing so, the node-based conservation condition boils down to the nodal solver, allowing to compute the nodal velocity of each nodes. The Eulerian numerical flux takes into account of the nodal velocity, building a node-based stencil where all the neighboring cells sharing a node are taken into account. This Riemann solver associated to the subface-based scheme carries the positivity preserving and entropy stability properties. A second-order extension is accomplished using classical methods: polynomial reconstruction (refer to Diot (2002)) plus the Venkatakrishnan (1995) limiter for space discretization and the Runge-Kutta method (refer to Gottlieb & Shu (1998)) for time discretization. When compared to classical two-point schemes, the multi-point scheme appears to be insensitive to numerical instabilities, i.e., the carbuncle phenomenon and the odd-even decoupling. A straightforward extension to three-dimensional cases is then achieved. The three-dimensional algorithm is validated with the 2D axisymmetric equations in the first place. A few challenging test cases involving re-entry vehicles are also exhibited. To complete this chapter, the implicit time-stepping scheme is presented. This scheme is useful mainly to study steady inviscid flows.

In the final chapter, we implement the subface-based scheme on system of conservation laws with source terms, namely the shallow-water equations, with the aim of building a well-balanced Godunov-type scheme. A consistent practice is to build a one-dimensional Eulerian approximate Riemann solver with its Lagrangian counterpart in the first place. Once the well-balanced property of the scheme is achieved, we proceed to the multidimensional subface-based scheme incorporated to the two-dimensional shallow water equations. Numerical validations allow us to come to the same conclusion where the multi-point scheme seems to be insensitive to numerical instabilities.

Further work will regard the extension of the presented multidimensional scheme for the Navier-Stokes equations on unstructured grids which was the main motivation of this thesis. Extension on real gas and ALE method is also considered in order to establish a complete aerodynamics module. An extensive numerical analysis of the existing scheme will be performed to better understand numerical instabilities such as the carbuncle phenomenon and also to characterize the multi-point scheme that seems insensitive to numerical instabilities. Other than that, for three-dimensional cases, a preliminary study on the main principles and basic design of waveriders (see appendix 1.4) was carried out. Subsequently, a detailed study on the flow over the object and eventually an improvement

on the design of the waverider is yet to be done. As for the implicit time marching scheme, the sequel is to develop the three-dimensional implicit scheme along with it a parallel simulation code. A fully multi-point linearization of the matrix form based on the nodal velocity and an extension to second order would also be in the lineup of perspectives. The last chapter on well-balanced Godunov-type Finite Volume scheme was more on an exploratory research, therefore it gives a heap of possible suite, including the study on the entropy stability of the well-balanced scheme, a convex combination CFL condition that ensures the positivity-preserving solution etc. Lastly, extension to more complex physics such as the MHD equations are also in view.

A Appendices

1.1 State-of-the-art Riemann solvers

1.1.1 Roe Riemann solver

The Roe (1981) Riemann solver is one of the earliest approximate Riemann solvers to improve the performance of the Godunov method. This solver is an exact Riemann solver applied to the linearized Riemann problem written as

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbb{A}(\mathbf{U}_l, \mathbf{U}_r)\frac{\partial \mathbf{U}}{\partial x} = \mathbf{0}, \tag{A.1}$$

where $\mathbb{A}$ is the Roe average Jacobian matrix and is required to satisfy the three following properties :

- Hyperbolicity. The Jacobian matrix $\mathbb{A}$ should have purely real eigenvalues $\lambda_i = \lambda_i(\mathbf{U}_l, \mathbf{U}_r)$ ordered as $\lambda_1, \lambda_2 \dots \lambda_n$ with its corresponding set of complete eigenvectors $\mathbf{v}_1, \mathbf{v}_2 \dots \mathbf{v}_n$;

- Consistency. The exact Jacobian yields $\mathbb{A}(\mathbf{U}, \mathbf{U}) = \mathbb{A}(\mathbf{U})$;

- Conservation across discontinuities. $\mathbf{F}_r - \mathbf{F}_l = \mathbb{A}(\mathbf{U}_r - \mathbf{U}_l)$.

Matrix $\mathbb{A}$ can be constructed by different methods that leads to the definition of the Roe averaged values with the Roe average values

$$\begin{aligned}
\hat{\rho} &= \sqrt{\rho_l \rho_r}, \\
\hat{u} &= \frac{\sqrt{\rho_l} u_l + \sqrt{\rho_r} u_r}{\sqrt{\rho_l} + \sqrt{\rho_r}}, \\
\hat{H} &= \frac{H_l \sqrt{\rho_l} + H_r \sqrt{\rho_l}}{\sqrt{\rho_l} + \sqrt{\rho_r}}, \\
\hat{c} &= \sqrt{(\gamma - 1)\left(\hat{H} - \frac{\hat{u}^2}{2}\right)},
\end{aligned} \tag{A.2}$$

where $\hat{H}$ is the total enthalpy. The eigenvalues, λ and the eigenvectors, $\mathbf{v}$ of the Jacobian matrix are

$$\begin{array}{rclcrcl}
\lambda_1 &=& \hat{u} - \hat{c} &,& \mathbf{v}_1 &=& [1, \hat{u} - \hat{c}, \hat{H} - \hat{u}\hat{c}]^t, \\
\lambda_2 &=& \hat{u} &,& \mathbf{v}_2 &=& [1, \hat{u}, \hat{u}^2/2]^t, \\
\lambda_3 &=& \hat{u} + \hat{c} &,& \mathbf{v}_3 &=& [1, \hat{u} + \hat{c}, \hat{H} + \hat{u}\hat{c}]^t.
\end{array} \tag{A.3}$$

To complete the Roe numerical flux $\mathbf{F}^{Roe}$, it remains to determine the wave strengths α_i by projecting the jump $\Delta\mathbf{U}$ onto the right, average eigenvectors, $\Delta\mathbf{U} = \sum_{i=1}^{3} \alpha_i \mathbf{v}_i$. With some algebra manipulation, the jumps α yield

$$\begin{aligned}
\alpha_1 &= \frac{1}{2\hat{c}^2}(\Delta p - \hat{c}\hat{\rho}\Delta u), \\
\alpha_2 &= \Delta\rho - \frac{1}{\hat{c}^2}\Delta p, \\
\alpha_3 &= \frac{1}{2\hat{c}^2}(\Delta p + \hat{c}\hat{\rho}\Delta u)
\end{aligned} \tag{A.4}$$

where $\Delta p = p_r - p_l$, $\Delta\rho = \rho_r - \rho_l$ and $\Delta u = u_r - u_l$. Finally the Roe flux is written as

$$\mathbf{F}^{Roe}(\mathbf{U}_l, \mathbf{U}_r) = \frac{1}{2}(\mathbf{F}_l + \mathbf{F}_r) - \frac{1}{2}\sum_{i=1}^{3} \alpha_i |\lambda_i| \mathbf{v}_i. \tag{A.5}$$

Note that the solution of the linearized Riemann problem consists only of discontinuous jumps, meaning that the approximation is correct for contact discontinuities and shocks, whereas for rarefaction waves, the approximation can lead to nonphysical phenomena due to violation of the second law of thermodynamics. An entropy fix is therefore introduced to the Roe approximate solver by modifying the modulus of the eigenvalues

$$|\lambda_i|^\star = \begin{cases} |\lambda_i|, & \text{if} \quad \lambda_i \geq \delta, \\ \dfrac{1}{2\delta}(|\lambda_i|^2 + \delta^2), & \text{if} \quad \lambda_i < \delta. \end{cases} \tag{A.6}$$

The value of δ is arbitrary as long as it is small relative to unity. By construction, the Roe approximate Riemann solver unfortunately has the drawback of creating nonphysical results, *i.e.* entropy violating discontinuous waves, or generating the infamous shock instabilities such as the carbuncle phenomenon. Note that the effect of this drawback can be reduced by implementing an appropriate entropy fix.

1.1.2 HLL Riemann solver

The HLL approximate Riemann solver is first proposed in Harten *et al.* (1983) where an approximate solution of the Riemann problem (2.2.93) is sought by computing the numerical flux directly. Given initial conditions $\mathbf{U}_l$ and $\mathbf{U}_r$ and assuming Λ_l and Λ_r the fastest left and right wave speeds emerging from the solution of the Riemann problem, the structure of the approximate solver is given in figure A.1. The states are represented as follows:

$$\mathbf{U}(x,t) = \begin{cases} \mathbf{U}_l & \text{if} \quad 0 \leq \Lambda_l, \\ \mathbf{U}^\star & \text{if} \quad \Lambda_l \leq 0 \leq \Lambda_r, \\ \mathbf{U}_r & \text{if} \quad 0 \geq \Lambda_r, \end{cases} \tag{A.7}$$

and the numerical flux is determined by

$$\mathbf{F}_{HLL}(\mathbf{U}_l, \mathbf{U}_r) = \begin{cases} \mathbf{F}(\mathbf{U}_l) & \text{if} \quad 0 \leq \Lambda_l, \\ \mathbf{F}^\star & \text{if} \quad \Lambda_l \leq 0 \leq \Lambda_r, \\ \mathbf{F}(\mathbf{U}_r) & \text{if} \quad 0 \geq \Lambda_r. \end{cases} \tag{A.8}$$

We want to determine the values of $\mathbf{U}^\star$ that allow us to deduce the numerical flux at the

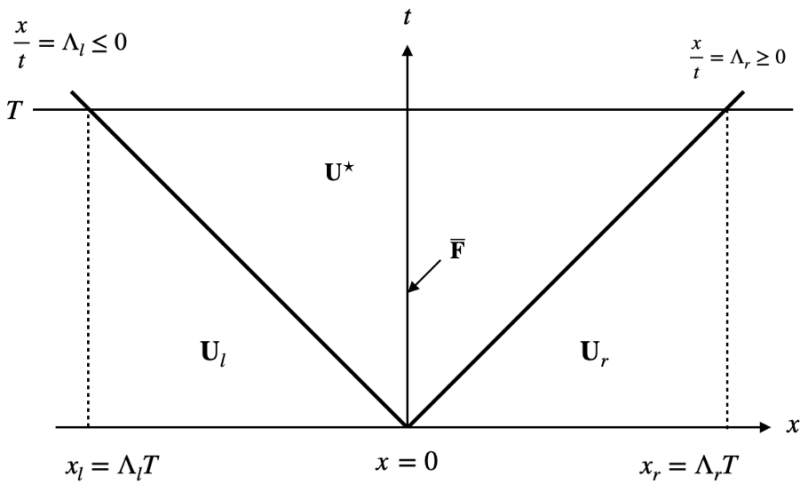


Figure A.1: Wave configuration of the HLL Riemann solver.

interface as $\overline{\mathbf{F}} = \frac{1}{2}(\mathbf{F}^- + \mathbf{F}^+)$. Recalling the definition of the left-sided and right-sided

fluxes for the HLL Riemann solver following 2.1.4, we arrive at

$$\begin{aligned} \mathbf{F}^- &= \mathbf{F}(\mathbf{U}_l) - \Lambda_l^-(\mathbf{U}^\star - \mathbf{U}_l) - \Lambda_r^-(\mathbf{U}_r - \mathbf{U}^\star), \\ \mathbf{F}^+ &= \mathbf{F}(\mathbf{U}_r) - \Lambda_l^+(\mathbf{U}^\star - \mathbf{U}_l) - \Lambda_r^+(\mathbf{U}_r - \mathbf{U}^\star), \end{aligned}$$

The conservative property of the Riemann solver allows to derive the intermediate state $\mathbf{U}^\star$ such that when $\mathbf{F}^+ = \mathbf{F}^-$,

$$\mathbf{F}(\mathbf{U}_l) - \mathbf{F}(\mathbf{U}_r) - \Lambda_l^-(\mathbf{U}^\star - \mathbf{U}_l) - \Lambda_r^-(\mathbf{U}_r - \mathbf{U}^\star) = \mathbf{0}.$$

and the intermediate state vector reads

$$\mathbf{U}^\star = \frac{\Lambda_r \mathbf{U}_r - \Lambda_l \mathbf{U}_l + \mathbf{F}(\mathbf{U}_l) - \mathbf{F}(\mathbf{U}_r)}{\Lambda_r - \Lambda_l}. \tag{A.9}$$

The average numerical flux at the cell-interface yields

$$\overline{\mathbf{F}} = \frac{1}{2}(\mathbf{F}^- + \mathbf{F}^+) = \frac{1}{2}(\mathbf{F}(\mathbf{U}_l) + \mathbf{F}(\mathbf{U}_r)) - \frac{1}{2}|\Lambda_l|(\mathbf{U}^\star - \mathbf{U}_l) - \frac{1}{2}|\Lambda_r|(\mathbf{U}_r - \mathbf{U}^\star). \tag{A.10}$$

Finally, to complete the HLL scheme, the wave speeds Λ_l and Λ_r are estimated with

$$\begin{cases} \Lambda_l &= \min(u_l - a_l, u_r - a_r), \\ \Lambda_r &= \max(u_l + a_l, u_r + a_r). \end{cases} \tag{A.11}$$

The HLL Riemann solver consists of only two waves and therefore is considered incomplete, resulting in numerical diffusion for intermediate waves.

1.1.3 HLLC Riemann solver

The HLLC (Harten, Lax, Van Leer, Contact) Riemann solver, see Toro (1999) is a modified version of the HLL solver. A crucial difference between the HLL and HLLC solvers is that the HLLC solver takes into account the presence of intermediate waves, such as contact discontinuities and shear waves. The structure of the solution as presented in figure A.2 composed of four constant states separated by three discontinuities Λ_l, Λ_0 and Λ_r.

$$\mathbf{F}(\mathbf{U}_l, \mathbf{U}_r)^{HLLC} = \begin{cases} \mathbf{F}(\mathbf{U}_l) & \text{if} \quad 0 \leq \Lambda_l, \\ \mathbf{F}_l^\star & \text{if} \quad \Lambda_l \leq 0 \leq \Lambda_0, \\ \mathbf{F}_r^\star & \text{if} \quad \Lambda_0 \leq 0 \leq \Lambda_r, \\ \mathbf{F}(\mathbf{U}_r) & \text{if} \quad 0 \geq \Lambda_r. \end{cases} \tag{A.12}$$

Once again following 2.1.4, the left-sided and right-sided fluxes for the HLLC Riemann solver writes

$$\begin{aligned} \mathbf{F}^- &= \mathbf{F}(\mathbf{U}_l) - \Lambda_l^-(\mathbf{U}_l^\star - \mathbf{U}_l) - \Lambda_0^-(\mathbf{U}_r^\star - \mathbf{U}_l^\star) - \Lambda_r^-(\mathbf{U}_r - \mathbf{U}_r^\star), \\ \mathbf{F}^+ &= \mathbf{F}(\mathbf{U}_r) - \Lambda_l^+(\mathbf{U}_l^\star - \mathbf{U}_l) - \Lambda_0^+(\mathbf{U}_r^\star - \mathbf{U}_l^\star) - \Lambda_r^+(\mathbf{U}_r - \mathbf{U}_r^\star), \end{aligned}$$

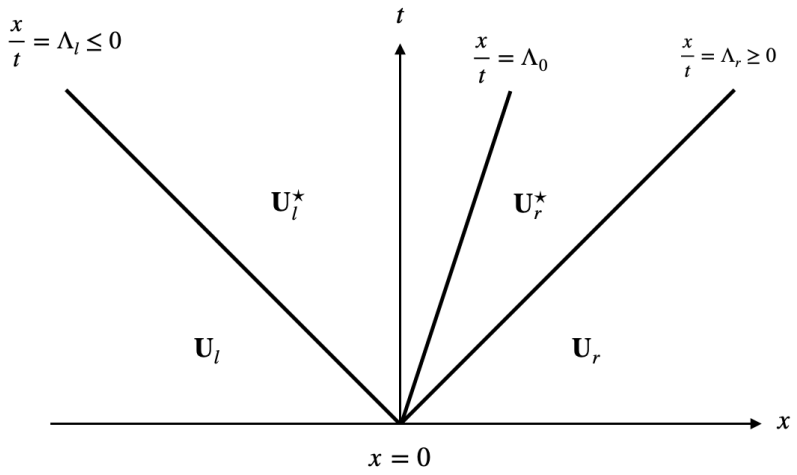


Figure A.2: Structure of the solution of the Riemann problem in the $x - t$ plane for the HLLC approximate Riemann solver..

The Rankine-Hugoniot conditions across the waves speeds allow to write

$$\mathbf{F}_l^\star = \mathbf{F}_l + \Lambda_l(\mathbf{U}_l^\star - \mathbf{U}_l), \quad , \quad \mathbf{F}_r^\star = \mathbf{F}_l^\star + \Lambda_0(\mathbf{U}_r^\star - \mathbf{U}_l^\star) \quad , \quad \mathbf{F}_r^\star = \mathbf{F}_r + \Lambda_r(\mathbf{U}_r^\star - \mathbf{U}_r).$$

The aim is to determine the vectors $\mathbf{U}_s^\star$ with $s = l, r$ in order to compute the average flux at an interface. The following conditions are imposed on the Riemann solver, where $u_l^\star = u_r^\star = u^\star = \Lambda_0$ and $p_l^\star = p_r^\star = p^\star$. Similarly to the HLL solver, the unknowns are obtained by implementing the integral form of the differential conservation laws. The intermediate states $\mathbf{U}_s^\star$ are given by

$$\mathbf{U}_s^\star = \rho_s \left(\frac{\Lambda_s - u_s}{\Lambda_s - \Lambda_0} \right) \begin{bmatrix} 1 \\ \Lambda_0 \\ \frac{E_s}{\rho_s} + (\Lambda_0 - u_l)\left(\Lambda_0 + \dfrac{p_l}{\rho_s(\Lambda_s - u_s)} \right) \end{bmatrix}, \tag{A.13}$$

with the intermediate wave speed $\Lambda_0 = \dfrac{p_r - p_l + \rho_l u_l(\Lambda_l - u_l) - \rho_r u_r(\Lambda_r - u_r)}{\rho_l(\Lambda_l - u_l) - \rho_r(\Lambda_r - u_r)}$. As for the left and right wave speeds, different choices to compute Λ_l and Λ_r are discussed in Batten *et al.* (1997) without a firm basis and clear agreement without practitioners. Finally, the average numerical flux at the cell-interface yields

$$\overline{\mathbf{F}} = \frac{1}{2}(\mathbf{F}^- + \mathbf{F}^+) = \frac{1}{2}(\mathbf{F}(\mathbf{U}_l) + \mathbf{F}(\mathbf{U}_r)) - \frac{1}{2}|\Lambda_l|(\mathbf{U}_l^\star - \mathbf{U}_l) - \frac{1}{2}|\Lambda_0|(\mathbf{U}_r^\star - \mathbf{U}_l^\star) - \frac{1}{2}|\Lambda_r|(\mathbf{U}_r - \mathbf{U}_r^\star). \tag{A.14}$$

1.2 Axisymmetric Euler equations

A three-dimensional flow is sometimes invariant under rotation along the axis of symmetry, making it possible to describe the problem with only two space coordinates. This approach has the advantage of transforming an initially 3D problem into a 2D axisymmetric problem, allowing to consider 3 velocity components at the same time conserving the initial shape of the case. The aim of this extension is to validate the 3D Euler equation with cylindrical symmetry when the azimuthal velocity is taken into account.

The vector form of the axisymmetric Euler equations can be written as

$$(y\mathbf{U})_t + (y\mathbf{F})_x + (y\mathbf{G})_y = \mathbf{S}, \tag{A.1}$$

with

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{pmatrix}, \qquad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{pmatrix}, \qquad \mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vH \end{pmatrix}, \qquad \mathbf{S} = \begin{pmatrix} 0 \\ 0 \\ p \\ 0 \end{pmatrix},$$

We implement a classical approach where a source term is derived from (A.1) and added to the right-hand side of the equation. The source term is defined as

$$\mathbf{RHS} = \frac{(\mathbf{S} - \mathbf{G})}{y}, \tag{A.2}$$

and the numerical scheme then becomes

$$\mathbf{U}_c^{n+1} - \mathbf{U}_c^n + \frac{\Delta t}{|\omega_c|} \sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{SF}(pc)} l_{pcf} \overline{\mathbf{F}}_{pcf} = \frac{\Delta t}{|\omega_c|} \sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{SF}(pc)} l_{pcf} \mathbf{RHS}_{pcf}, \tag{A.3}$$

with

$$\mathbf{RHS} = -\frac{1}{y} \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 \\ (\rho e + p)u \end{pmatrix}.$$

1.3 Some properties of the steady Euler equations

The governing equations for steady Euler equations are

$$\begin{array}{rcl} \nabla \cdot (\rho \mathbf{v}) & = & 0, \\ \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) + \nabla p & = & \mathbf{0}, \\ \nabla \cdot (\rho e \mathbf{v}) + \nabla \cdot (p \mathbf{v}) & = & 0, \end{array} \tag{A.1}$$

with $e = \varepsilon + \frac{1}{2}\mathbf{v}^2$, the total energy, $h = \varepsilon + \frac{p}{\rho}$ enthalpy and $H = h + \frac{1}{2}\mathbf{v}^2$ the total enthalpy. The total energy equation rewrites

$$\nabla \cdot (\rho H \mathbf{v}) = 0. \tag{A.2}$$

For smooth solutions, $\nabla \cdot (\rho H \mathbf{v}) = H \nabla \cdot (\rho \mathbf{v}) + \rho \mathbf{v} \cdot \nabla H$, and due to the first equation in (A.1), we arrive at

$$\mathbf{v} \cdot \nabla H = 0. \tag{A.3}$$

This means that for smooth flows, it is constant along streamlines. We proceed to study the jump across a discontinuity surface σ characterized by unit normal $\mathbf{n}$, such as $\mathbf{n}^2 = 1$. The jump across σ reads $[\phi] = \phi_+ - \phi_-$, and $\phi_+ = \lim_{h \to 0} \phi(\mathbf{x} + h\mathbf{n})$, $\phi_- = \lim_{h \to 0} \phi(\mathbf{x} - h\mathbf{n})$ for $h > 0$, and $\mathbf{x} \in \sigma$. Recall that

$$[\phi\psi] = \overline{\phi}[\psi] + \overline{\psi}[\phi], \tag{A.4}$$

where $\overline{\phi} = \frac{1}{2}(\phi_- + \phi_+)$. The Rankine-Hugoniot conditions for the governing equations therefore writes,

$$\begin{array}{rcl} [\rho \mathbf{v} \cdot \mathbf{n}] & = & 0, \\ [\rho (\mathbf{v} \otimes \mathbf{v}) \mathbf{n}] + [p]\, \mathbf{n} & = & \mathbf{0}, \\ [\rho H \mathbf{v} \cdot \mathbf{n}] & = & 0. \end{array} \tag{A.5}$$

The first equation of (A.5) can be rewritten as a mass flux

$$\rho_- \mathbf{v}_- \cdot \mathbf{n} = \rho_+ \mathbf{v}_+ \cdot \mathbf{n} = \mu.$$

Employing (1.3), the mass flux turns into

$$(\overline{\mathbf{v} \cdot \mathbf{n}})[\rho] + \overline{\rho}[\mathbf{v} \cdot \mathbf{n}] = 0. \tag{A.6}$$

We also have $\frac{\mu}{\rho^+} = \mathbf{v}_+ \cdot \mathbf{n}$ and $\frac{\mu}{\rho^-} = \mathbf{v}_- \cdot \mathbf{n}$, implying

$$\mu \left[\frac{1}{\rho}\right] - [\mathbf{v} \cdot \mathbf{n}] = 0. \tag{A.7}$$

For the moment equation, inserting (1.3) into it leads to $\overline{\rho(\mathbf{v}\cdot\mathbf{n})}\,[\mathbf{v}]+\overline{\mathbf{v}}\,[\rho\mathbf{v}\cdot\mathbf{n}]+[p]\,\mathbf{n}=\mathbf{0}$. Employing the same arguments, we arrive at

$$\mu\,[H]=0. \tag{A.8}$$

To study the steady Rankine-Hugoniot relations, we reuse $\mu=\rho_-\mathbf{v}_-\cdot\mathbf{n}=\rho_+\mathbf{v}_+\cdot\mathbf{n}$. The jump relations then become

$$\begin{array}{rcl}\mu\left[\dfrac{1}{\rho}\right]-[\mathbf{v}\cdot\mathbf{n}] &=& 0,\\ \mu\,[\mathbf{v}]+[p]\,\mathbf{n} &=& \mathbf{0},\\ [H] &=& 0.\end{array} \tag{A.9}$$

Projecting the second jump relation on to the normal direction leads to

$$\mu\,[\mathbf{v}\cdot\mathbf{n}]+[p]=0. \tag{A.10}$$

It is clear that the tangential component for the velocity is continuous across the shock wave, and that the total enthalpy is conserved across the shock wave. Substituting the first jump relations into (A.10) yields

$$[p]=-\mu^2\left[\frac{1}{\rho}\right]. \tag{A.11}$$

Recalling that $H=\varepsilon+\left[\dfrac{p}{\rho}\right]+\dfrac{1}{2}\mathbf{v}^2$, and employing (1.3), we recover the Hugoniot relation

$$[\varepsilon]-\overline{p}\left[\frac{1}{\rho}\right]=0. \tag{A.12}$$

Now, let us investigate the steady supersonic flow in front of a half cylinder where we will observe the presence of a curved detached shock in front of the body such as illustrated in figure A.3. The incident flow is a uniform flow defined by $\rho_\infty, p_\infty, \mathbf{v}_\infty=v_\infty\mathbf{e}_x, Ma_\infty=\dfrac{\mathbf{v}_\infty}{a_\infty}>1, H_\infty=\varepsilon_\infty+\dfrac{p_\infty}{\rho_\infty}+\dfrac{1}{2}\mathbf{v}_\infty^2$ and $a^2=-\left(\dfrac{\partial p}{\partial\tau}\right)_\eta$ the isentropic sound speed. For a gamma gas law $p=(\gamma-1)\rho\varepsilon$ and $a^2=\gamma\dfrac{p}{\rho}$. The total enthalpy is conserved not only along streamlines but also across the shock wave and thus $H=H_\infty$ throughout the steady flow. For a gamma gas law,

$$H=\frac{a^2}{\gamma-1}+\frac{1}{2}\mathbf{v}^2=H_\infty. \tag{A.13}$$

The physical quantities at the wall are denoted using the subscript w, and 0 is the stagnation point belonging to the symmetry axis $\mathbf{v}_0=v_0\mathbf{e}_x$ and $v_0=0$ due to the wall boundary condition. Writing (A.13) between the incident flow and the stagnation point

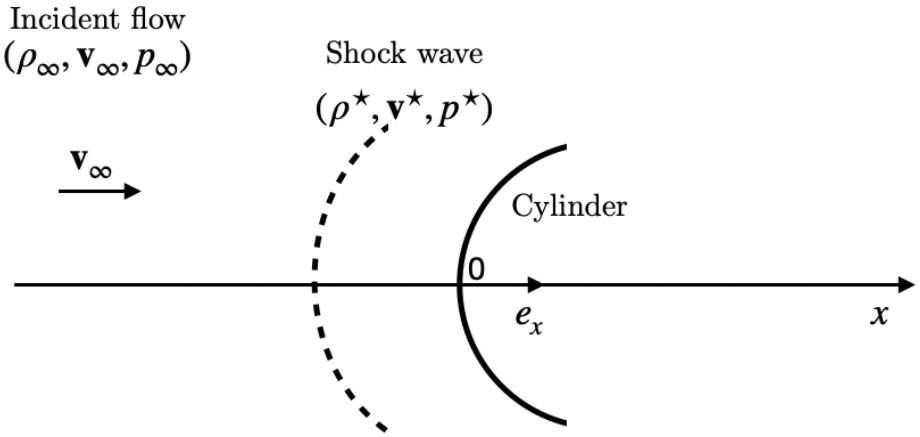


Figure A.3: Illustration of a detached curved shock in front of a cylinder body.

along the symmetry axis leads to $\dfrac{a_\infty}{\gamma-1} + \dfrac{1}{2}v_\infty^2 = \dfrac{a_w^2}{\gamma-1}$, which allows to express the sound speed at the stagnation point as

$$a_w^2 = a_\infty^2 \left[1 + \frac{(\gamma-1)}{2} M_\infty^2\right]. \tag{A.14}$$

Behind the shock wave, the physical quantities are denoted by means of the superscript $\star$. The shocked quantities are computed by means of the Rankine-Hugoniot relations. Let us rewrite it along the symmetry axis

$$\mu = \rho_\infty v_\infty > 0, \quad \rho_\infty v_\infty \left(\frac{1}{\rho^\star} - \frac{1}{\rho_\infty}\right) - (v^\star - v_\infty) = 0,$$

where $v^\star$ is the normal component of $\mathbf{v}^\star$. This leads to

$$\frac{\rho^\star}{\rho_\infty} = \frac{v_\infty}{v^\star}, \tag{A.15}$$

which is nothing but the mass conservation along the streamline.

From Hugoniot relation (A.12), we have

$$\varepsilon^\star - \varepsilon_\infty + \frac{1}{2}(p^\star + p_\infty)\left(\frac{1}{\rho^\star} - \frac{1}{\rho_\infty}\right) = 0.$$

Substituting $\varepsilon = \dfrac{p}{(\gamma-1)}\rho$ into the foregoing relation yields

$$\frac{1}{\rho^\star} = \frac{1}{\rho_\infty}\frac{(\gamma+1)\rho_\infty + (\gamma-1)p^\star}{(\gamma-1)\rho_\infty + (\gamma+1)p^\star}. \tag{A.16}$$

Inserting the above equation into (A.11), we arrive at

$$p^\star = p_\infty \left[1 + \frac{2\gamma}{\gamma+1}(M_\infty^2 - 1)\right]. \tag{A.17}$$

From the mass conservation (A.15), we deduce

$$\rho^\star = \rho_\infty \left[1 - \frac{2}{\gamma+1}\left(\frac{M_\infty - 1}{M_\infty}\right)\right]^{-1}. \tag{A.18}$$

Asymptotically, $\dfrac{\rho^\star}{\rho_\infty} \to \dfrac{\gamma+1}{\gamma-1}$ when $M_\infty \to +\infty$. The flow is isentropic between the wall and the shock wave along the streamline, therefore $\dfrac{p_w}{p^\star} = \left(\dfrac{\rho_w}{\rho^\star}\right)^\gamma$, and, on the other hand, $\dfrac{a_w^2}{(a^\star)^2} = \dfrac{p_w}{p^\star}\dfrac{\rho^\star}{\rho_w}$. This then leads to

$$\begin{aligned} \frac{p_w}{p^\star} &= \left[\frac{a_w^2}{(a^\star)^2}\right]^{\frac{\gamma}{\gamma-1}}, \\ \frac{\rho_w}{\rho^\star} &= \left[\frac{a_w^2}{(a^\star)^2}\right]^{\frac{1}{\gamma-1}}. \end{aligned} \tag{A.19}$$

Asymptotically, $\dfrac{a_w^2}{a^{2\star}} \to \dfrac{(\gamma+1)^2}{4\gamma}$ when $M_\infty \to +\infty$. With some algebraic manipulation, the stagnation pressure $\dfrac{p_w}{p_\infty}$ is given By

$$\frac{p_w}{p_\infty} = \left(\frac{\gamma+1}{2}M_\infty^2\right)^{\frac{\gamma}{\gamma-1}} \left[1 + \frac{2\gamma}{\gamma+1}(M_\infty^2 - 1)\right]^{-\frac{1}{\gamma-1}}, \tag{A.20}$$

and accordingly the stagnation density $\dfrac{\rho_w}{\rho_\infty}$ reads

$$\frac{\rho_w}{\rho_\infty} = \left[1 + \frac{\gamma-1}{2}M_\infty^2\right]^{\frac{1}{\gamma-1}} \left[1 + \frac{2\gamma}{\gamma+1}(M_\infty^2 - 1)\right]^{-\frac{1}{\gamma-1}} \left[1 - \frac{2}{\gamma+1}\frac{(M_\infty^2 - 1)}{M_\infty^2}\right]^{-\frac{\gamma}{\gamma-1}}. \tag{A.21}$$

Asymptotically, $\dfrac{\rho_w}{\rho_\infty} \to \left[\dfrac{(\gamma+1)^2}{4\gamma}\right]^{\frac{1}{\gamma-1}} \left(\dfrac{\gamma+1}{\gamma-1}\right)$ when $M_\infty \to +\infty$. In our case, $\gamma = \dfrac{7}{5}$, therefore the stagnation density value is about $\dfrac{\rho_w}{\rho_\infty} \approx 6.4378$.

1.4 Waveriders

Waverider is a hypersonic atmospheric entry vehicle design that improves its supersonic lift-to-drag ratio using the shock waves generated by its own flight as a compression lift. The waverider concept has been of great interest for high-speed vehicle design. An introductory study was carried out during this thesis mainly to understand the flight concept of different type of waveriders and designing one. Here, we present a brief history on the evolution of waveriders, followed by a step-by-step design of a conical waverider. For a few introductory literatures on waveriders, refer to Kuchemann (2012), Lunan (2015), Ferguson *et al.* (2015) and Bowcutt (1986).

1.4.1 Brief history of waveriders

The first design concept of a waverider was described in 1959 when Nonweiler (1959) designed the caret wing waverider. It is a delta-winged aircraft with its rear that resemble the caret symbol, hence it's name. The caret wing waverider uses the flow field behind a planar oblique shock formed on a two-dimensional wedge and the stream surfaces behind the shock to generate its body shape. The caret wing rides on top of the attached shock wave at the designed Mach number, however, the flow beneath the vehicle has a high pressure due to the shock, resulting in a compression lift being generated on the lower surface.

Inspired by Nonweiler's work, waverider designs has since then evolved with various study on waverider designs using different shockwave shapes. One of the simplest earliest design was the conical flow waverider by Jones *et al.* (1967). The aircraft is generated by the conical shock generated by a cone and the waverider is designed to keep the rounded shockwave attached to its wings in order to increase lift. An extension of the conical waverider are those of osculating cone waverider, see Chauffour & Lewis (2004), where the three-dimensional flow field is discretized into several osculating planes and the flow is assumed to be conical within each plane.

Another principal category of the waverider is the viscous optimized waverider. The complex interaction between the flow field and the boundary layer instigated research on optimized waverider designs by taking into account the effects of the boundary layer. For instance, in the work of Corda & Jr. (1988), the viscous optimized waveriders is designed from general axisymmetric flow fields to shape the lower surface of the vehicle. The shape of the viscous optimizer waverider is similar to the conical ones when the shock wave on the nose is beyond a certain critical angle. An iterative process is implemented for the optimization of the design.

Today, hypersonic flight is becoming a common goal in aerospace technology and are on the quest on coming up with better optimized designs based on different aspects for hypersonic vehicles. To cite a few, various methods are summarized in Zao *et al.* (2020) to improve its aerodynamic performance for wide-speed range waveriders. In Liu *et al.* (2018), a new osculating flow field method is developed based on variable shock angles to give greater flexibility in design and increase the lift-to-drag ratio of the waverider.

This section provided a glance off the history and different categories of waveriders. Optimization of waverider design will progress along with the fast-growing development in aerospace technology research. The following section presents a step-by-step process for the construction of a conical waverider that was done during this thesis. A thorough numerical simulation is in view for future work, along with it an analysis on the flow around the object and its lift-to-drag ratio and eventually improving the design of waveriders.

1.4.2 Design of conical waverider

This subsection present the process implemented to construct a conical waverider during this thesis. We start by introducing a few vocabularies of the waverider. Figure A.4 is an illustration of a conical waverider and the key vocabulary to construct a waverider.

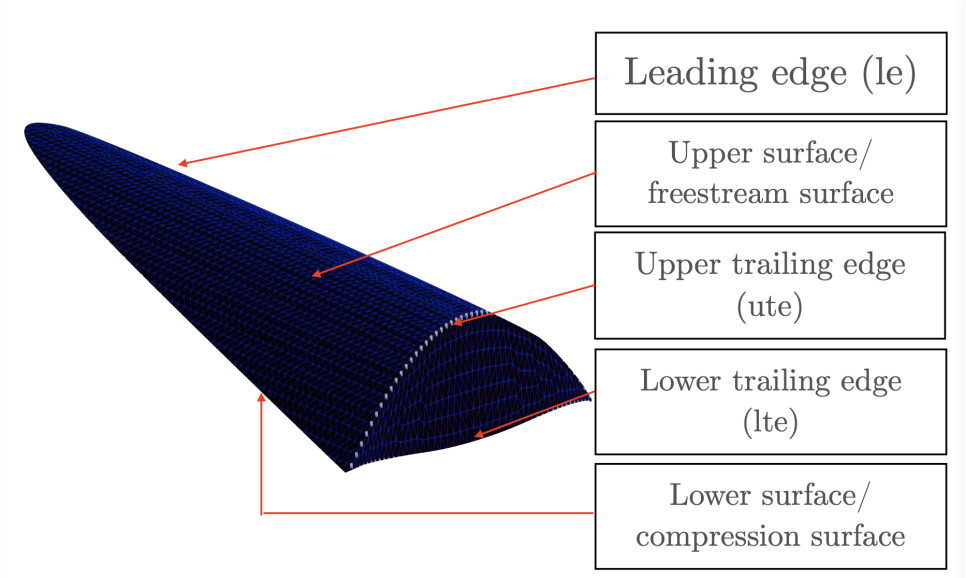


Figure A.4: Waverider vocabulary.

To begin, we fix the following parameters:

$$Ma = 10,\ \delta = 5.5°,\ l = 1m,\ \phi_l = 30°,$$

and the following are the procedure to construct a conical waverider.

Step 1. Compute a Taylor Maccoll analysis to generate a flow past a cone. The Taylor Maccoll (1937) analysis allows to guess a shock angle β and velocity (V_r, V_s) downstream of the shock, for a given Mach number Ma and cone angle δ. Start by numerically integrating the Taylor Maccoll differential equation for V_r and V_s from the shock wave until $V_\theta = 0$. When $V_\theta = 0$ at $\theta = \theta_{surf}$, an iterative process is implemented to adjust the shock angle until $\theta_{surf} = \beta$.

Step 2. Generate a conical flow field. Generating the conical field requires geometrical parameters of the base cone and the length of the cone l. The parameters and the equation required to generate the conical flow field is given in table A.1 and figure A.5 illustrates the flow field.

	Shock cone	Cone cone
Base radius at $z = l(R)$	$l = tan(\beta)$	$l = tan(\delta)$
X - Coordinate	$x = R_\beta cos\Theta_\beta$	$x = R_\delta cos\Theta_\delta$
Y - Coordinate	$y = R_\delta sin\Theta_\beta$	$y = R_\delta sin\Theta_\delta$
Z - Coordinate	$z = R_\beta / tan_\beta$	$x = R_\delta / tan_\delta$

Table A.1: Generating a conical flow field.

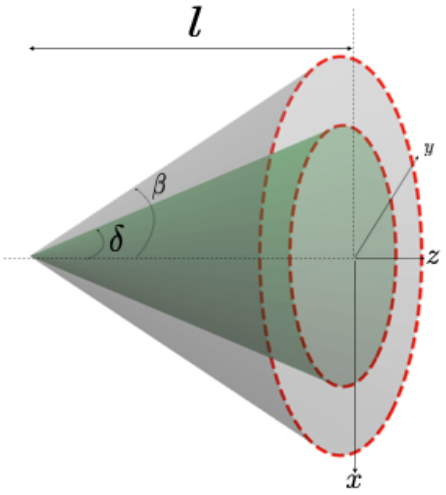


Figure A.5: Defining the upper trailing edge.

Step 3. Define the upper trailing edge and the upper surface. Assuming that the upper trailing edge at $z = l$ is to be a parabola that can be described in the Cartesian coordinate with

$$\mathbf{X} = R_0 + A\mathbf{Y}^2,$$

with

$$\mathbf{X} = x/(l \cdot \delta) \text{and} \mathbf{Y} = y/(l \cdot \delta).$$

We define a dimensionless quantity

$$\sigma = \beta/\delta.$$

At the shock wave,

$$X_s = \sigma cos(\phi_l),$$
$$Y_s = \sigma sin(\phi_l).$$

Thus, $A = (X_s - R_0)/Y_x^2 \text{and} R_0/X_s = 0.75$ in general. Note that the lower the values of R_0/X_s, the larger the drag and vice versa. The upper trailing edges are then defined by

$$x_{ute} = X \cdot l \cdot \delta,$$

$$y_{ute} = Y \cdot l \cdot \delta,$$

$$z_{ute} = l.$$

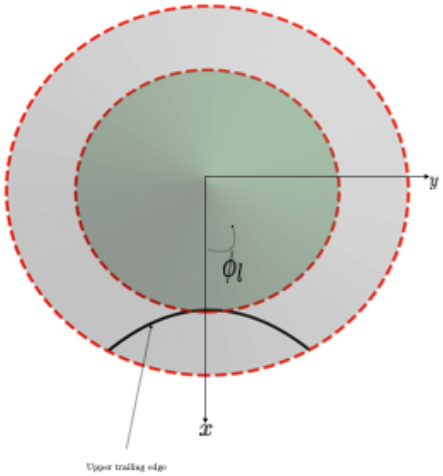


Figure A.6: Defining the upper trailing edge.

Step 4. Define the leading edge. The upper surface or also known as the free stream surface is defined by assuming that the surface is parallel to the free stream flow. As for the leading edge, we define along the free flow direction until the intersection with the shock wave such as

$$x_{le} = x_{ute},$$

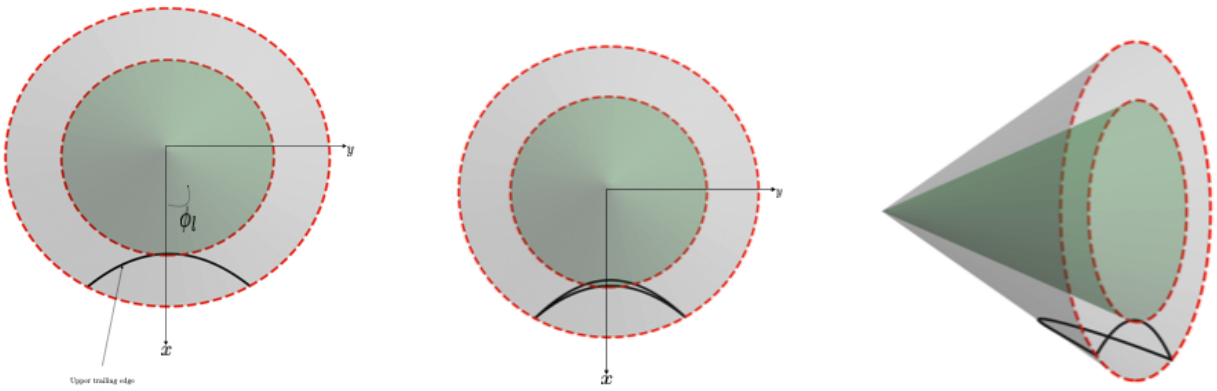
$$y_{le} = y_{ute},$$

$$z_{le} = r/tan(\beta),$$

with $r = \sqrt{x_{ute}^2 + y_{ute}^2}$.

Step 5. Define the lower trailing edge. The last step consists of defining the lower surface, also known as the compression surface of the waverider. The streamline tracking method is a classical way to trace the streamlines from leading edge. Therefore, in this work, we applied the simplified tracing method, also known as the geometric relations method (see Ding *et al.* (2013)). The geometric relations to obtain the lower surface are the followings :

$$x_{lte} = x_{ute} + x_{le} \cdot (x_{ute} - zle) \cdot tan\delta/\sqrt{x_{le}^2 + y_{le}^2},$$



Upper surface Leading edge.

Figure A.7: Defining the upper surface and leading edge of the conical waverider.

$$y_{lte} = y_{ute} + y_{le} \cdot (z_{ute} - zle) \cdot tan\delta / \sqrt{x_{le}^2 + y_{le}^2},$$
$$z_{lte} = z_{ute}.$$

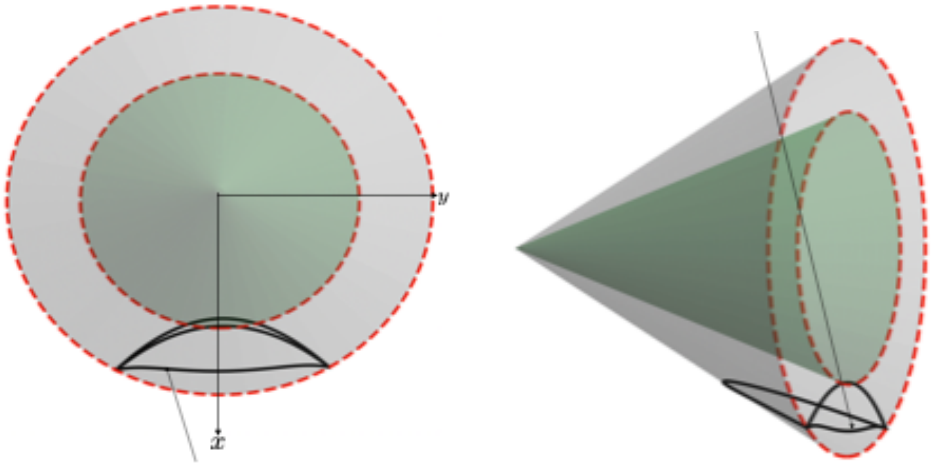


Figure A.8: Defining the lower surface.

A python script was developed to construct the conical waverider as described. The script is then called in the GMSH mesh generator in order to generate the domain mesh and a quick simulation was done on the three-dimensional simulation code. As mentioned earlier, further studies is required to analyze the flow and study its lift-to-drag ratio. Improvements are yet to be made on the design of the waverider in order to achieve a better aerodynamic efficiency.

Bibliography

Abgrall, R. 2018 Some remarks about conservation for residual distribution schemes. *Computational Methods for applied Mathematics* **81** (3), 327 – 351. (Cited on page 83.)

Alcrudo, F. & Garcia-Navarro, P. 1993 A high resolution Godunovov-type scheme in finite volumes for the 2d shallow water equations. *Int. J. Numer. Methods Fluids* **16**, 489–505. (Cited on pages 184 and 185.)

Anderson, J.D. 2006 *Hypersonic and high-temperature gas dynamics*. AIAA education series. (Cited on page 119.)

Annaloro, J., Galera, S., Karrang, P., Prevereaud, Y., Verant, J-L., Spel, M., Hauwaert, P. Van & Omaly, P. 2017 Space debris atmospheric entry prediction with spacecraft-oriented tools. *7th European Conference on Space Debris* . (Cited on pages 137, 138, 140, and 141.)

Balay, S., Abhyankar, S., Adamsand, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W.-D., Karpeyev, D., Kaushik, D., Knepley, M.G., May, D.-A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B.-F., Zampini, S. & Zhang, H. 2021*a* PETSc users manual. *Tech. Rep.* ANL-95/11 - Revision 3.15. Argonne National Laboratory. (Cited on page 130.)

Balay, S., Abhyankar, S., Adamsand, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W.-D., Karpeyev, D., Kaushik, D., Knepley, M.G., May, D.-A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B.-F., Zampini, S. & Zhang, H. 2021*b* PETSc Web page. https://www.mcs.anl.gov/petsc. (Cited on page 130.)

Balay, S., Gropp, W. D., McInnes, L. C. & Smith, Barry F. 1997 Efficient management of parallelism in object oriented numerical software libraries. *Modern Software Tools in Scientific Computing* pp. 163–202. (Cited on page 130.)

Balsara, D. & Dumbser, M. 2015 Self-similar internal structure-part 2-Application to hyperbolic conservation laws on unstructured meshes. *Journal of Computational Physics* **287**, 269–292. (Cited on page 2.)

Barth, T.J. & Jespersen, D.C. 1989 The design and application of upwind schemes on unstructured meshes. *AIAA Paper 89-0366* pp. 1–12. (Cited on page 110.)

Batten, P., Clarke, N., Lambert, C. & Causon, D.M. 1997 On the choice of wave speeds for the hllc riemann solver. *SIAM Journal on Scientific Computing (SISC)* **18** (6), 1553–1570. (Cited on pages 2, 45, and 201.)

Boscheri, W., Dumbser, M., Loubère, R. & Maire, P.-H. 2018 A second-order cell-centered lagrangian ADER-MOOD finite volume scheme on multidimensional unstructured meshes for hydrodynamics. *Journal of Computational Physics* **358**, 103 – 129. (Cited on page 57.)

Boscheri, W., Loubère, R. & Dumbser, M. 2015 Direct Arbitrary-Lagrangian-Eulerian ADER-MOOD finite volume schemes for multidimensional hyperbolic conservation laws. *Journal of Computational Physics* **292**, 56 – 87. (Cited on page 57.)

Bouchut, F. 2004 *Nonlinear Stability of Finite Volume Methods for Hyperbolic Conservation Laws*, 1st edn. Birkhauser. (Cited on pages 12, 15, 77, 103, and 160.)

Bouchut, F. & de Luna, T. Morales 2009 Semi-discrete entropy satisfying approximate riemann solvers. the case of the suliciu relaxation approximation. *Journal of Scientific Computing* **41**. (Cited on page 24.)

Bowcutt, K.G. 1986 Optimization of hypersonic waveriders derived from cone flows—including viscous effects. PhD thesis, University of Maryland, College Park, phD Dissertation. (Cited on page 207.)

Castro, M., Gallardo, J. M., López-García, J. A. & Parés, C. 2008 Well-balanced high order extensions of godunov's method for semilinear balance laws. *SIAM Journal on Numerical Analysis* **46** (2), 1012–1039. (Cited on page 184.)

Castro, M., J. & Parés, C. 2020 Well-balanced high-order Finite Volume methods for systems of balance laws. *Journal of Scientific Computing* **82** (2), 48. (Cited on page 184.)

Chalons, C. 2010 Nonlinear hyperbolic systems of conservation laws. Lecture notes available at https://chalons.perso.math.cnrs.fr/hyperbolic.htm. (Cited on page 50.)

Chalons, C., Girardin, M. & Kokh, S. 2013 Large time step and asymptotic preserving numerical schemes for the gas dynamics equations with source terms. *SIAM Journal on Scientific Computing (SISC)* **35**, A2874–A2902. (Cited on page 28.)

Chan, A., Gallice, G., Loubère, R. & Maire, P.-H. 2021 Positivity preserving and entropy consistent approximate Riemann solvers dedicated to the high-order MOOD-based Finite Volume discretization of Lagrangian and Eulerian gas dynamics. *Computers & Fluids* **229**. (Cited on pages i, iii, 3, 9, 55, 84, 100, 101, 103, 106, 164, 175, 184, 191, and 193.)

Chauffour, M.L. & Lewis, M.J. 2004 Corrected waverider design for inlet applications. *40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit* . (Cited on page 207.)

Cheng, J.C. & Shu, C.W. 2014 Positivity-preserving lagrangian scheme for multimaterial compressible scheme. *Journal of Computational Physics* **257**, 143–168. (Cited on page 3.)

Clain, S., Diot, S. & Loubère, R. 2011 A high-order finite volume method for systems of conservation laws – multi-dimensional optimal order detection (MOOD). *Journal of Computational Physics* **230** (10), 4028 – 4050. (Cited on pages 57 and 59.)

Clain, S. & Figueiredo, J. 2017 The mood method for the non-conservative shallow-water system. *Computers & Fluids* **145**, 99 – 128. (Cited on pages 58 and 191.)

Corda, S. & Jr., D. Anderson 1988 Viscous optimized hypersonic waveriders designed from axisymmetric flow fields. *AIAA 26th Aerospace Sciences Meeting* . (Cited on page 207.)

Deconinck, H., Roe, P.L. & Struijs, R. 1993 A multidimensional generalization of Roe's flux difference splitter for the Euler equations. *Computers & Fluids* **22**, 215–222. (Cited on page 2.)

Després, B. 2001 Lagrangian systems of conservation laws. *Numerische Mathematik* **89** (1), 99–134. (Cited on page 90.)

Després, B. & Mazeran, C. 2005 Lagrangian gas dynamics in two dimensions and Lagrangian systems. *ARMA* **178**, 327–372. (Cited on pages 3 and 75.)

Desveaux, V. & C.Berthon 2013 An entropic MOOD scheme for the euler equations. *International Journal of Finite Volumes* . (Cited on page 57.)

Dieudonne, W., Boerrigter, H.L. & Charbonnier, J.M. 1999 Hypersonic flow on a blunted cone-flare in the vki-h3 Mach 6 wind tunnel. *Tech. Rep.*. Von Karman Institute. (Cited on page 136.)

Ding, F., liu, J., Jin, L. & Luo, S. B. 2013 Comparison between methods of generation of waveriders derived from streamline tracing and simplified method. *Applied Mechanics and Materials* **390**, 134–140. (Cited on page 210.)

Diot, S. 2002 La méthode MOOD, Multi-dimensional Optimal Order Detection, la première approche a posteriori aux méthodes Volumes Finies d'ordre tres élevé. PhD thesis, University of Bordeaux, France, habilitation à diriger des recherches. (Cited on pages 109 and 194.)

Dukowicz, J. K. 1984 A general, non-iterative Riemann Solver for Godunov's method. *Journal of Computational Physics* **61**, 119–137. (Cited on page 180.)

Einfeldt, B. 1988 On Godunov-type methods for gas dynamics. *SIAM Journal on Numerical Analysis* **25** (2), 294–318. (Cited on pages 1 and 2.)

Engquist, B. & Osher, S. 1981 One-sided difference approximations for nonlinear conservation laws. *Mathematics of Computation* **36** (154), 321–321. (Cited on page 1.)

Fambri, F., Dumbser, M. & Zanotti, O. 2017 Space–time adaptive ADER-DG schemes for dissipative flows: Compressible Navier–Stokes and resistive MHD equations. *Computer Physics Communications* **220**, 297 – 318. (Cited on page 58.)

Farmakis, P. S., Tsoutsanis, P. & Nogueira, X. 2020 Weno schemes on unstructured meshes using a relaxed a posteriori MOOD limiting approach. *Computer Methods in Applied Mechanics and Engineering* **363**, 112921. (Cited on page 57.)

Ferguson, F., Dasque, N. & Dhanasar, M. 2015 Waverider design and analysis. *International Space planes and hypersonic systems and technologies conferences* . (Cited on page 207.)

Fleischmann, N., Adami, S. & Adams, N. A. 2020 A shock-stable modification of the HLLC riemann solver with reduced numerical dissipation. *Journal of Computational Physics* **423**. (Cited on page 4.)

G. Gallice, A. Chan, R. Loubère & Maire, P.-H. 2022 Entropy stable and positivity preserving Godunov-type schemes for multidimensional hyperbolic systems on unstructured grid. *Journal of Computational Physics* **468**. (Cited on pages 71 and 173.)

Gaitonde, D. 1992 High-speed viscous flows past blunt bodies and compression corner with flux-split methods. *Tech. Rep.*. Wright Laboratory. (Cited on page 118.)

Gallardo, J., Pares, C. & Castro, M. 2007 On a well-balanced high-order Finite Volume scheme for shallow water equations with topography and dry areas. *Journal of Computational Physics* **227**. (Cited on pages 160 and 168.)

Gallice, G. 1997 Système d'Euler-Poisson, magnétohydrodynamique et schémas de Roe. PhD thesis, Université Bordeaux I. (Cited on page 16.)

Gallice, G. 2000 Schémas de type Godunov entropiques et positifs préservant les discontinuités de contact. *C.R. Acad. Sci. Paris, Ser. I* **331** (2), 149 – 152. (Cited on pages 5, 15, and 84.)

Gallice, G. 2001 Schémas de type Godunov entropiques et positifs pour la dynamique des gaz et la magnétohydrodynamique lagrangiennes. *C.R. Acad. Sci. Paris, Ser. I* **332** (11), 1037 – 1040. (Cited on page 193.)

Gallice, G. 2002*a* Approximation numérique de Systèmes Hyperboliques Non-linéaires Conservatifs ou Non-conservatifs . PhD thesis, University of Bordeaux, France, habilitation à diriger des recherches. (Cited on pages i, iii, 3, 5, 22, 24, 41, 70, 84, 88, 90, 92, 100, 161, 171, 175, 190, and 193.)

GALLICE, G. 2002*b* Solveurs simples positifs et entropiques pour les systèmes hyperboliques avec terme source. *C.R. Acad. Sci. Paris, Ser. I* **334** (8), 713 – 716. (Cited on pages 157 and 160.)

GALLICE, G. 2003 Positive and Entropy Stable Godunov-Type Schemes for Gas Dynamics and MHD Equations in Lagrangian or Eulerian Coordinates. *Numerische Mathematik* **94** (4), 673–713. (Cited on pages i, iii, 3, 4, 5, 9, 15, 20, 22, 25, 41, 68, 70, 71, 84, 88, 90, 92, 95, 100, 164, 175, 193, and 194.)

GALLICE, G., CHAN, A., LOUBÈRE, R. & MAIRE, P.-H. 2022 Entropy stable and positivity preserving godunov-type schemes for multidimensional hyperbolic systems on unstructured grid. *Journal of Computational Physics* . (Cited on pages 172 and 175.)

GALLOUET, T., HÉRARD, J.-M & SEGUIN, N. 2003 Some approximate godunov schemes to compute shallow-water equations with topography. *Computers & Fluids* **32**, 479–513. (Cited on page 170.)

GEUZAINE, C. & REMACLE, J.-F. 2009 Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering* **79** (11), 1309–1331. (Cited on page 110.)

GODLEWSKI, E. & RAVIART, P. A. 1996 *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer. (Cited on pages 1, 12, 19, 38, 50, 72, 75, and 98.)

GODUNOV, S.K. 1999 Reminiscences about difference schemes. *Journal of Computational Physics* **153** (1), 6 – 25. (Cited on page 1.)

GODUNOV, S.K., ZABRODINE, A., IVANOV, M., KRAIKO, A. & PROKOPOV, G. 1979 *Résolution numérique des problèmes multidimensionnels de la dynamique des gaz*. Mir. (Cited on page 1.)

GOTTLIEB, S. & SHU, C.W. 1998 Total variation diminishing Runge-Kutta schemes. *Mathematics of Computation* **67**, 73–85. (Cited on pages 57, 61, 109, and 194.)

GOTTLIEB, SIGAL, SHU, CHI-WANG & TADMOR, EITAN 2001 Strong stability-preserving high-order time discretization methods. *SIAM Review* **43** (1), 89–112. (Cited on page 57.)

GOUTAL, N. & MAUREL, F. 1997 Proceedings of the 2nd workshop on dam-break wave simulation. *Tech. Rep.*. EDF-DER. (Cited on page 167.)

HARTEN, A. & LAX, P.D. 1981 A Random Choice Finite Difference Scheme for Hyperbolic Conservation Laws. *SIAM Journal on Numerical Analysis* **18** (2), 289–315. (Cited on page 87.)

HARTEN, A., LAX, P.D. & VAN LEER, B. 1983 On upstream Differencing and Godunov-Type schemes for Hyperbolic Conservation Laws. *SIAM Review* **25** (1), 35–61. (Cited on pages 1, 2, 10, 11, 12, 13, 14, 15, 76, 87, and 199.)

Jacq, P., Maire, P.-H. & Abgrall, R. 2014 A Nominally Second-Order Cell-Centered Finite Volume Scheme for Simulating Three-Dimensional Anisotropic Diffusion Equations on Unstructured Grids. *Communications in Computational Physics* **16**, 841–891. (Cited on page 75.)

Jensen, J. L. 1906 Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica* **30**, 175 – 193. (Cited on page 13.)

Jiang, G.-S. & Shu, C.W. 1996 Efficient implementation of weighted ENO schemes. *Journal of Computational Physics* **126**, 202–228. (Cited on page 63.)

Jiang, Z. H., Y., C. & Yu, J. 2018 Efficient methods with higher order interpolation and MOOD strategy for compressible turbulence simulations. *Journal of Computational Physics* **371**, 528 – 550. (Cited on page 58.)

Jones, J.G., Pike, J. & Roe, P.I. 1967 A method for designing lifting configurations for high supersonic speeds, using axisymmetric flow fields. *Tech. Rep.*. Royal Aircraft Establishment of England. (Cited on page 207.)

Kamm, J.R. & Timmes, F.X. 2007 On efficient generation of numerically robust sedov solutions. *Tech. Rep.* LA-UR-07-2849. Los Alamos National Laboratory. (Cited on page 122.)

Kemm, F. 2018 Heuristical and numerical considerations for the carbuncle phenomenon. *Applied Mathematics and Computation* **320**, 596–613. (Cited on pages 4 and 189.)

Kemm, F., Gaburro, E., Thein, F. & Dumbser, M. 2020 A simple diffuse interface approach for compressible flows around moving solids of arbitrary shape based on a reduced Baer-Nunziato model. *Computers & Fluids* **204**. (Cited on page 58.)

Kitamura, K. & Hashimoto, A. 2017 Simple a posteriori slope limiter (post limiter) for high resolution and efficient flow computations. *Journal of Computational Physics* **341**, 313 – 340. (Cited on page 57.)

Knepley, M.G. 1997 Matt knepley's homepage. (Cited on page 131.)

Kuchemann, D. 2012 *The Aerodynamic Design of Aircraft*. AIAA. (Cited on page 207.)

Kulikowskii, A.G., Pogorelov, N.V. & Semenov, A. Yu. 2001 *Mathematical Aspects of Numerical Solution of Hyperbolic Systems*. Chapman & Hall/CRC. (Cited on page 85.)

Lange, M., Gorman, G.J. & Knepley, M. 2015 Flexible, Scalable Mesh and Data Management PETSc DMPLex . (Cited on page 131.)

Leveque, R. 1998 Balancing source terms and flux gradients in high-resolution Godunov methods: The quasi-steady wave-propagation algorithm. *Tech. Rep.*. (Cited on page 158.)

Liu, W., Liu, J., Ding, F., Li, K. & Xia, W.X. 2018 Novel osculating flow field methodology for hypersonic waverider vehicles based on variable shock angle. *Tech. Rep.*. American Society of Civil Engineers. (Cited on page 207.)

Loubère, R., Dumbser, M. & Diot, S. 2014 A new family of high order unstructured MOOD and ADER finite volume schemes for multidimensional systems of hyperbolic conservation laws. *Communication in Computational Physics* **16**, 718–763. (Cited on page 57.)

Loubère, R., Maire, P.-H. & Rebourcet, B. 2016 *Handbook of Numerical Methods for Hyperbolic Problems: Basic and Fundamental Issues, edited by R. Abgrall and C.W. Shu*, chap. 13 Staggered and colocated Finite Volume scheme for Lagrangian hydrodynamics, pp. 319–352. North Holland. (Cited on pages i, iii, and 75.)

Loubère, R., Maire, P.-H., Shashkov, M., Breil, J. & Galera, S. 2010 ReALE: A reconnection-based Arbitrary-Lagrangian-Eulerian method. *Journal of Computational Physics* **229** (12), 4724–4761. (Cited on pages 95 and 110.)

Lunan, D. 2015 *Waverider, a Revise Chronology* . (Cited on page 207.)

Maccoll, J.W. 1937 The conical shock wave formed by a cone moving at a high speed. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* pp. 459–472. (Cited on page 208.)

Maire, P.-H. 2009 A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *Journal of Computational Physics* **228**. (Cited on pages 3, 4, 75, 95, 97, and 105.)

Maire, P.-H. 2011 Contribution to the numerical modeling of Inertial Confinement Fusion . PhD thesis, University of Bordeaux, France, habilitation à diriger des recherches. (Cited on page 95.)

Maire, P.-H. & Breil, J. 2012 A nominally second-order accurate finite volume cell-centered scheme for anisotropic diffusion on two-dimensional unstructured grids. *Journal of Computational Physics* **231**, 2559–2299. (Cited on page 75.)

Menikoff, R. 1994 Errors when shock waves interact due to numerical shock width. *SIAM Journal on Scientific Computing* **15** (5), 1227–1242. (Cited on page 124.)

Menikoff, R. & B.J.Plohr 1989 The riemann problem for fluid flows of real materials. *Review of Modern Physics* **61**, 75–130. (Cited on pages 24 and 102.)

Munz, C. D. 1994 On godunov-type schemes for Lagrangian gas dynamics. *IAM Journal on Numerical Analysis* **31**, 17–42. (Cited on page 3.)

Munz, C. D., Einfeldt, B., Roe, P. L. & Sjögreen, B. 1991 On Godunov-type methods near low densities. *Journal of Computational Physics* **92**, 273–295. (Cited on page 1.)

Nishikawa, H. & Kitamura, K. 2008 Very simple, carbuncle-free, boundary-layer-resolving rotated-hybrid riemann solvers. *Journal of Computational Physics* **227**, 2560–2581. (Cited on page 147.)

Nogueira, X., Ramírez, L., Clain, S. & Loubère, R. 2016 High accurate SPH method with multidimensional optimal order detection. *Computer Methods in Applied Mechanics and Engineering* **310**, 134–155. (Cited on page 57.)

Noh, W.F. 1987 Errors for calculations of strong shocks using an artificial viscosity andan artificial heat flux. *Journal of Computational Physics* **72**, 78–120. (Cited on page 123.)

Nonweiler, T.R.F. 1959 Aerodynamic problems of manned space vehicles. *The Aeronautical Journal* **63**, 521–528. (Cited on page 207.)

Osher, S. 1984 Riemann solvers, the entropy condition and difference approximations. *SIAM Journal on Numerical Analysis* **21**, 217–235. (Cited on page 1.)

Pandolfi, M. & D'Ambrosio, D. 2001 Numerical Instabilities in Upwind Methods: Analysis and Cures for the Carbuncle Phenomenon. *Journal of Computational Physics* **166** (2), 271–301. (Cited on page 118.)

Peery, K. & Imlay, S. 1988 Blunt-body flow simulations. *24th Joint Propulsion Conference* . (Cited on page 118.)

Petsopoulos, T. & Regan, F. J. 1996 Moving-mass roll control system for fixed-trim re-entry vehicle. *Journal of Spacecraft and Rockets* **33**, 54 – 61. (Cited on page 136.)

Qu, F., Sun, D., Liu, Q. S. & Bai, J. Q. 2022 A Review of Riemann Solvers for Hypersonic Flows. *Computational Methods in Engineering* **29**, 1771–1800. (Cited on page 1.)

Quirk, J.J. 1994 A contribution to the great Riemann solver debate. *International Journal of Numerical Methods in Fluids* **18**, 555–574. (Cited on pages 4, 71, 115, and 154.)

Rider, W.J. 2000 Revisiting wall heating. *Journal of Computational Physics* **162** (2), 395–410. (Cited on page 124.)

Riemann, B. 1860 Uber die fortpflanzung ebener luftwellen von endlicher schwingungsweite.abhandlungen der k€oniglichen gesellschaft der wissenschaften zu g€ottingen. —- **8**, 43–65. (Cited on page 1.)

Rodionov, A.V. 2017 Artificial viscosity Godunov-type schemes to cure the carbuncle phenomenon. *Journal of Computational Physics* **345**, 308–329. (Cited on pages 4, 5, 117, 119, and 123.)

Roe, P.L. 1981 Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics* **43**, 357–372. (Cited on pages 1, 2, 15, and 197.)

Sangeet, S. & Manda, J. C. 2018 A cure for numerical shock instability in hllc riemann solver using anti diffusion control. *Computers & fluids* **174**, 144–166. (Cited on page 4.)

Savino, R. & Paterna, D. 2005 Blunted cone-flare in hypersonic flow. *Computers & Fluids* **34**, 859–875. (Cited on pages 135, 136, and 151.)

Sedov, L.I. 1959 *Similarity and Dimensional Methods in Mechanics*. New York: Academic Press. (Cited on pages 120 and 133.)

Shen, Z., Yan, W. & Yuan, G. 2014 A robust and contact resolving Riemann solver on unstructured mesh, Part I, Euler method. *Journal of Computational Physics* **268**, 432–455. (Cited on pages i, iii, 4, 75, 105, and 193.)

Shu, C.W. 1997 Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for hyperbolic Conservation Laws. *NASA/CR-97-206253 ICASE Report No.97-65* . (Cited on page 114.)

Toro, E.F. 1999 *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 2nd edn. Springer. (Cited on pages 1, 15, 34, 35, 37, 41, 44, 51, 52, 89, 103, 111, 144, and 200.)

Toro, E.F., Müller, L.O. & Siviglia, A. 2020 Bounds for Wave Speeds in the Riemann Problem: Direct Theoretical Estimates. *Computers & Fluids* **209**. (Cited on page 45.)

Toro, E.F., Spruce, M. & Speares, W. 1994 Restoration of the contact surface in the Harten-Lax-van Leer Riemann solver. *Journal of Shock Waves* **4**, 25–34. (Cited on pages 1, 2, and 44.)

Venkatakrishnan, V. 1995 Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. *Journal of Computational Physics* **118** (1), 120 – 130. (Cited on pages 110 and 194.)

Vilar, F. 2019 A posteriori correction of high-order discontinuous galerkin scheme through subcell finite volume formulation and flux reconstruction. *Journal of Computational Physics* **387**, 245 – 279. (Cited on page 57.)

Vilar, F., Shu, C.W. & Maire, P.-H. 2016 Positivity-preserving cell-centered Lagrangian schemes for multi-material compressible flows: From first-order to high-orders. Part I: The one-dimensional case. *Journal of Computational Physics* **312**, 385–415. (Cited on pages 3, 24, and 102.)

Walpot, L. 2002 Development and application of a hypersonic flow solver. PhD thesis, Deft Technical University. (Cited on page 138.)

Woodward, P. & Colella, P. 1984 The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics* **54**, 115–173. (Cited on pages 34 and 124.)

Xing, Y.L. & Shu, C.W. 2005 High order finite difference WENO schemes with the exact conservation property for the shallow water equations. *Journal of Computational Physics* **208**, 206–227. (Cited on page 166.)

Yee, H. C., Beam, R. M. & Warming, R. F. 1982 Boundary approximations for implicit schemes for one-dimensional inviscid equations of gas dynamics. *AIAA Journal* **20** (9). (Cited on page 140.)

Zanotti, O., Fambri, F., Dumbser, M. & Hidalgo, A. 2015 Space-time adaptive ADER Discontinuous Galerkin finite element schemes with a posteriori sub-cell finite volume limiting. *Computers & Fluids* **118**, 204–224. (Cited on pages 57 and 58.)

Zao, Z. T., huang, W., Yan, L. & Yang, Y. G. 2020 An overview of research on wide-speed range waverider configuration. *Progress in Aerospace Sciences* **113**. (Cited on page 207.)