Conception and Analysis of Very High Order Residual Distribution Schemes. Application to Fluid Mechanics.

Adam Larat

N° d'ordre : 3873

THÈSE

PRÉSENTÉE À

L'UNIVERSITÉ DE BORDEAUX I

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

Par **Adam LARAT**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : MATHÉMATIQUES APPLIQUÉES

Conception et analyse de schémas d'ordre très élevé distribuant le résidu. Application à la mécanique des fluides.

Soutenue le : 6 Novembre 2009

Après avis des rapporteurs :

Herman DECONINCK	Head of AR Department, VKI, Brussels
Frédéric COQUEL ...	CR, LJLL, Paris

Devant la commission d'examen composée de :

Luc MIEUSSENS	Professeur, IMB	Président du Jury
Herman DECONINCK	Head of AR Department, VKI, Brussels	Rapporteur
Frédéric COQUEL ...	CR, LJLL, Paris	Rapporteur
Rémi ABGRALL	Professeur, IMB	Directeur de Thèse
Angelo IOLLO	Professeur, IMB	Examinateur
Vincent COUAILLER	Ingénieur, ONERA	Examinateur
Mario RICCHIUTO ..	CR, INRIA	Invité

– 2009 –

Remerciements

Cette thèse a été réalisée au sein de l'équipe Bacchus de l'INRIA Bordeaux Sud-Ouest et à l'Institut de Mathématiques de Bordeaux. Elle a été financée par le contrat de recherche européen ADIGMA, sous le contrat numéro 030719 (AST5-CT-2006-030719). D'une manière générale, je tiens à remercier l'ensemble des gens qui ont rendu ce travail possible.

L'ensemble du travail présenté dans la suite doit beaucoup à Rémi Abgrall et Mario Ricchiuto. J'ai beaucoup appris grâce à eux durant ces trois ans.

Rémi, mon directeur de thèse, est une mine de connaissance en Mathématiques que je n'ai jamais hésité à consulter et qui m'a rarement refusé un instant. De plus, dans sa grande réserve, il m'a quelque fois encouragé avec quelques mots qui ont suffit à me redonner du cœur à l'ouvrage, comme : - *«débugger, c'est marrant, c'est comme un jeux...»*

Mario a toujours été là. J'apprécie énormement sa vision des problèmes scientifiques, le temps qu'il a pu consacrer pour m'en montrer les tenants et aboutissants, son approche intuitive des choses, sa manière d'être, et tous les excellents moments qu'on a passés ensemble.

Je tiens également à remercier tous les membres du jury d'avoir accepté d'assister à ma soutenance. En particulier, je remercie les rapporteurs pour le temps qu'ils ont passé à la lecture attentive de mon manuscrit. Notamment Frédérique Coquel qui a eu la gentillesse de me signaler très simplement des erreurs grossières dans ma rédaction, ce qui a permis d'améliorer notablemment la qualité du manuscrit.

Tout au long de la rédaction, j'ai régulièrement pensé à Marc Garbey, directeur du Computer Science Departement à l'Université de Houston (TX, USA). En fait, je me rends compte aujourd'hui que c'est au cours du stage que j'ai effectué dans son laboratoire que j'ai pris réellement goût à la recherche dans le domaine des mathématiques numériques. Je le remercie vivement de m'avoir accueilli pendant 4 mois durant l'été 2005.

Je tiens ensuite à remercier toutes les personnes du labo qui m'ont un jour filé un coup de main.

En particulier, mes remerciements vont à deux ingénieurs, Pascal Jacq et Rémi Butel, qui, par leur travail de fourmis, ont largement participé à la maintenance et à l'amélioration de la plateforme "Fluidbox" dans laquelle a été implémenté l'essentiel des schémas présentés dans la suite.

Je ne peux pas écrire cette page sans un grand "MERCI !" à tous les geeks de l'INRIA qui savent sauver une journée de travail en trois lignes de commandes. Merci à Mathieu, Jérémie, Nicolas, Abdou, Orel, Damien, Xavier, etc... pour le temps que vous avez passé à vous occuper de mon incapacité informatique.

Je souhaite aussi remercier Robin Huart et Christelle Wervaecke pour les bons moments qu'on a pu passer ensemble à travailler sur les mêmes problèmes et je vous souhaite bon vent pour la suite de votre thèse.

Enfin, un petit clin d'oeil à Josy Baron, l'assistante de l'équipe Bacchus qui m'a toujours très bien guidé dans les procédures administratives.

Je termine cette séance de remerciements par mes proches.

Merci à mon frère, ma soeur et mes parents de n'avoir toujours rien compris au sujet ; on a pu parler d'autre chose. Merci à Papa et Maman d'être venus à ma soutenance et de s'être parlé comme avant. Merci à Maman d'avoir pleuré.

Je ne pense pas que Bordeaux ait été une si belle ville à mes yeux s'il n'y avait pas eu La Grasse Bande. Malgré les quelques tourments que cette fanfare a pu me faire endurer, j'y ai là mes meilleurs amis, mes plus grands confidents et toute ma stupidité... Merci la Grasse Bande, vous êtes beaux et je vous aime !

Parmi les gens de la Grasse Bande, il y en a deux avec lesquels j'ai vécu. Merci à Antoine Barré pour l'année et demi de colocation avec un thésard. Et surtout, merci à Coralie avec qui je vis maintenant et qui connait la vie d'un thésard de A à Z. J'ai toujours beaucoup apprécié ses ingénus : - *«Whaa ! C'est beau ce que tu fais !»*, souvent accompagnés d'un bisou.

Bordeaux, le 16 novembre 2009

Za to, že mně naučil pít pivo a pak plno dalších věci,
venuji tuto tézu mému dědovi V. Rouckovi.

Conception et analyse de schémas d'ordre très élevé distribuant le résidu. Application à la mécanique des fluides.

Résumé :

La simulation numérique est aujourd'hui un outils majeur dans la conception des objets aérodynamiques, que ce soit dans l'aéronautique, l'automobile, l'industrie navale, *etc...* Un des défis majeurs pour repousser les limites des codes de simulation est d'améliorer leur précision, tout en utilisant une quantité fixe de ressources (puissance et/ou temps de calcul). Cet objectif peut être atteint par deux approches différentes, soit en construisant une discrétisation fournissant sur un maillage donné une solution d'ordre très élevé, soit en construisant un schéma compact et massivement parallèlisable, de manière à minimiser le temps de calcul en distribuant le problème sur un grand nombre de processeurs. Dans cette thèse, nous tentons de rassembler ces deux approches par le développement et l'implémentation de Schéma Distribuant le Résidu ($\mathcal{RDS}$) d'ordre très élevé et de compacité maximale.

Ce manuscrit commence par un rappel des principaux résultats mathématiques concernant les Lois de Conservation hyperboliques ($\mathcal{CL}$s). Le but de cette première partie est de mettre en évidence les propriétés des solutions analytiques que nous cherchons à approcher, de manière à injecter ces propriétés dans celles de la solution discrète recherchée. Nous décrivons ensuite les trois étapes principales de la construction d'un schéma $\mathcal{RD}$ d'ordre très élevé :

- la représentation polynomiale d'ordre très élevé de la solution sur des polygones et des polyèdres;
- la description de méthodes distribuant le résidu de faible ordre, compactes et conservatives, consistantes avec une représentation polynomiale des données de très haut degré. Parmi elles, une attention particulière est donnée à la plus simple, issue d'une généralisation du schéma de Lax-Friedrichs (LxF);
- la mise en place d'une procédure préservant la positivité qui transforme tout schéma stable et linéaire, en un schéma non linéaire d'ordre très élevé, capturant les chocs de manière non oscillante.

Dans le manuscrit, nous montrons que les schémas obtenus par cette procédure sont consistants avec la $\mathcal{CL}$ considérée, qu'ils sont stables en norme $\mathcal{L}^\infty$ et qu'ils ont la bonne erreur de troncature. Même si tous ces développements théoriques ne sont démontrés que dans le cas de $\mathcal{CL}$s scalaires, des remarques au sujet des problèmes vectoriels sont faites dès que cela est possible. Malheureusement, lorsqu'on considère le schéma LxF, le problème algébrique non linéaire associé à la recherche de la solution stationnaire est en général mal posé. En particulier, on observe l'apparition de modes parasites de haute fréquence dans les régions de faible gradient. Ceux-ci sont éliminés grâce à un terme supplémentaire de stabilisation dont les effets et l'évaluation numérique sont précisément détaillés. Enfin, nous nous intéressons à une discrétisation correcte des conditions limites pour le schéma d'ordre élevé proposé.

Cette théorie est ensuite illustrée sur des cas test scalaires bidimensionnels simples. Afin de montrer la généralité de notre approche, des maillages composés uniquement de triangles et des maillages hybrides, composés de triangles et de quandrangles, sont utilisés. Les résultats obtenus par ces tests confirment ce qui est attendu par la théorie et mettent en avant certains avantages des maillages hybrides. Nous considérons ensuite des solutions bidimensionnelles des équations d'Euler de la dynamique des gaz. Les résultats sont assez bons, mais on perd les pentes de convergence attendues dès que des conditions limite de paroi sont utilisées. Ce problème nécessite encore d'être étudié. Nous présentons alors l'implémentation parallèle du schéma. Celle-ci est analysée et illustrée à travers des cas test tridimensionnel de grande taille. Du fait de la relative nouveauté et de la complexité des problèmes tridimensionels, seuls des remarques qualitatives sont faites pour ces cas test : le comportement global semble être bon, mais plus de travail est encore nécessaire pour définir les propriétés du schémas en trois dimensions. Enfin, nous présentons une extension possible du schéma aux équations de Navier-Stokes dans laquelle les termes visqueux sont traités par une formulation de type Galerkin. La consistance de cette formulation avec les équations de Navier-Stokes est démontrée et quelques remarques au sujet de la précision du schéma sont soulevées. La méthode est validé sur une couche limite de Blasius pour laquelle nous obtenons des résultats satisfaisants.

Ce travail offre une meilleure compréhension des propriétés générales des schémas $\mathcal{RD}$ d'ordre très élevé et soulève de nouvelles questions pour des améliorations futures. Ces améliorations devrait faire des schémas $\mathcal{RD}$ une alternative attractive aux discrétisations classiques $\mathcal{FV}$ ou ENO/WENO, aussi bien qu'aux schémas Galerkin Discontinu d'ordre très élevé, de plus en plus populaires.

Mots clés:

Distribution du Résidu, Fluctuation Splitting, Schémas d'ordre très élevé, Lois de Conservation, Hyperbolicité, Équations d'Euler, Équations de Navier-Stokes, Maillages non structurés, Maillages Hybrides, Traitement Parallèle, Discrétisation Compacte.

Discipline :

Mathématiques Appliquées

Conception and analysis of very high order distribution schemes. Application to fluid mechanics.

Abstract:

Numerical simulations are nowadays a major tool in aerodynamic design in aeronautic, automotive, naval industry *etc...* One of the main challenges to push further the limits of the simulation codes is to increase their accuracy within a fixed set of resources (computational power and/or time). Two possible approaches to deal with this issue are either to contruct discretizations yielding, on a given mesh, very high order accurate solutions, or to construct compact, massively parallelizable schemes to minimize the computational time by means of a high performance parallel implementation. In this thesis, we try to combine both approaches by investigating the contruction and implementation of very high order Residual Distribution Schemes ($\mathcal{RDS}$) with the most possible compact stencil.

The manuscript starts with a review of the mathematical theory of hyperbolic Conservation Laws ($\mathcal{CL}$s). The aim of this initial part is to highlight the properties of the analytical solutions we are trying to approximate, in order to be able to link these properties with the ones of the sought discrete solutions. Next, we describe the three main steps toward the construction of a very high order $\mathcal{RDS}$:

- The definition of higher order polynomial representations of the solution over polygons and polyhedra;
- The design of low order compact conservative RD schemes consistent with a given (high degree) polynomial representation. Among these, particular accest is put on the simplest, given by a generalization of the Lax-Friedrich's (LxF) scheme;
- The design of a positivity preserving nonlinear transformation, mapping first-order linear schemes onto nonlinear very high order schemes.

In the manuscript, we show formally that the schemes obtained following this procedure are consistent with the initial $\mathcal{CL}$, that they are stable in $\mathcal{L}^\infty$ norm, and that they have the proper truncation error. Even though all the theoretical developments are carried out for scalar $\mathcal{CL}$s, remarks on the extension to systems are given whenever possible. Unortunately, when employing the first order LxFscheme as a basis for the construction of the nonlinear discretization, the final nonlinear algebraic equation is not well-posed in general. In particular, for smoothly varying solutions one observes the appearance of high frequency spurious modes. In order to kill these modes, a streamline dissipation term is added to the scheme. The analytical implications of this modifications, as well as its practical computation, are thouroughly studied. Lastly, we focus on a correct discretization of the boundary conditions for the very high order $\mathcal{RDS}$ proposed.

The theory is then extensively verified on a variety of scalar two dimensional test cases. Both triangular, and hybrid triangular-quadrilateral meshes are used to show the generality of the approach. The results obtained in these tests confirm all the theoretical expectations in terms of accuracy and stability and underline some advantages of the hybrid grids. Next, we consider

solutions of the two dimensional Euler equations of gas dynamics. The results obtained are quite satisfactory and yet, we are not able to obtain the desired convergence rates on problems involving solid wall boundaries. Further investigation of this problem is under way. We then discuss the parallel implementation of the schemes, and analyze and illustrate the performance of this implementation on large three dimensional problems. Due to the preliminary character and the complexity of these three dimensional problems, a rather qualitative discussion is made for these tests cases: the overall behavior seems to be the correct one, but more work is necessary to assess the properties of the schemes in three dimensions. In the last chapter, we consider one possible extension to the Navier-Stokes equations in which the viscous terms are discretized by a standard Galerkin approach. We formally show that the overall discretization is consistent with the Navier-Stokes equations. However some accuracy issues are highlighted and discussed. The method is tested on a flat plate laminar boundary layer flow. The results are satisfactory.

The work presented in this thesis allows a better understanding of the general properties of very high order $\mathcal{RDS}$, and contributes substantially to bring forward a number of open issues for future improvement. These improvements should make $\mathcal{RD}$ discretizations a very appealing alternative to now classical high order and very high order $\mathcal{FV}$ ENO/WENO schemes, and to the increasingly popular class of Discontinuous Galerkin schemes.

Keywords:

Residual Distribution, Fluctuation Splitting, Very High Order Schemes, Conservative Laws, Hyperbolicity, Euler Equations, Navier-Stokes Equations, Unstructured Meshes, Hybrid Meshes, Parallel treatment, Compact Discretization.

Discipline:

Applied Mathematics

Content

Chapter 1

Introduction

1.1 Motivation and Context

The development of high-order algorithms for the simulation of compressible flows in complex domains and on arbitrary meshes is one of the most important research topics in Computational Fluid Dynamics (CFD). The continuous growth of the available computing power allows to increase the complexity of the flow configurations, object of the simulations, and to run always bigger test cases usually to obtain an improved accuracy on the flow parameters. However, improvements in the efficiency, flexibility and robustness of the numerical algorithms are still needed to fully exploit this computational potential.

It is generally agreed that, when dealing with complex geometries and flow patterns, the use of unstructured grids is somewhat mandatory. Compared to structured and multi-block structured grids, the generation of unstructured meshes, or more generally hybrid unstructured/structured meshes, can in fact be highly automated. A considerably lower degree of *user-input* and, consequently, less time [12], are needed. Moreover, unstructured mesh generation lends itself very naturally to solution-dependent local refinement and adaptation, which are known to improve the simulation output, and at the same time reduce the number of elements/degrees of freedom needed to achieve a fixed level of accuracy [12, 15, 18]. As a consequence, the design of new numerical algorithms for the simulation of compressible flows is largely oriented to formulations well suited for unstructured grids (see *e.g.* the volumes [18, 17]).

An abstract model for the fluid-mechanics equations is given by a so-called *Conservation Law*: a Partial Differential Equation (PDE) stating the conservation of some unknowns over a given region of space and time. The design of new numerical schemes for compressible flow simulations often starts with the study of simple *Conservation Laws* for which one has more theoretical information on the properties of the exact solution. It is generally accepted that state of the art of numerical methods for conservation laws on unstructured grids is not entirely satisfactory. The need of more flexible, accurate and robust solution algorithms for the analysis of large and complex systems is what drives the development of new techniques. Accuracy, robustness and efficiency requirements lead to the following *design constraints*:

Accuracy: The accuracy of a numerical solution is measured as its mathematical distance to the exact solution. It is well known this error is often a power function of a characteristic size

of the used mesh. The power coefficient measuring the speed of convergence of the method is called *the order of accuracy*. It is actually possible to increase the order of accuracy of the approximation in a relatively simple way, without introducing expensive reconstruction steps. Moreover, due to the fact that unstructured grids can be quite irregular (especially in 3D), the accuracy of the method should be as insensitive as possible to the regularity of the mesh;

Stability: Conservation laws admit *weak solutions* containing discontinuities. These solutions are piecewise smooth without strong oscillations in correspondence of the singularities. The numerical method must be able to handle discontinuities without polluting the solution with spurious oscillations, what usually leads to a reduced order of accuracy. Additionally, weak solutions of *Conservation Laws* also verify additional constraints imposed by the existence of a (vanishing) dissipative mechanism[1]. This gives an additional stability requirement for the numerical method. Ideally, the stability of the scheme (non-oscillatory character and energy/entropy stability) should be *parameter free*, that is, it should not depend on constants which are difficult to optimize in a general way;

Efficiency: Since the beginning of this century, CPU designers are able to still fit the Moore law [70] only thanks to the increasing number of processor cores inside the CPUs. In order to go along with this computation distribution, the numerical method of the future should allow a fast and efficient implementation, particularly on parallel platforms. From this point of view, the main requirements are simplicity and *compactness*. A compact method is one that, to update the values of the unknowns in a certain mesh location, only uses information contained in the closest grid entities. In parallel implementations, this allows to minimize the overhead due to inter-processor communication. Compactness is equivalent to the *locality* of the discrete procedure.

1.2 Methods Overview

This section presents a brief overview of the main methods used to approximate the solutions of compressible flow problems.

1.2.1 Finite Volume Methods

Within these methods, the *Finite Volume Methods* [66, 117] are certainly the most mature and the most documented ones. The reason of this is that most of the industrial codes for CFD have started by implementing this kind of methods. At the difference of the two next presented methods, the *Finite Volume Methods* are based on *Cell-Centered* approximation of the spatial domain: to each node of the mesh is associated a small area in its vicinity. It is called *the cell*. The node interacts with its neighbors through the edges of this cell. Problem is that in multiple dimensions, most $\mathcal{FV}$ schemes are designed by applying only one dimensional formulations along particular mesh directions (edges, edge normals, etc...). This often reduces dramatically the accuracy on irregular meshes and it is why this type of scheme suffers of strong deficiencies as far as accuracy and efficiency are concerned. Moreover, the construction of high order formulation necessitates the local reconstruction of polynomials of the proper degree, what is done by looking

[1] The entropy inequality implied by the second principle of thermodynamics is an example

for enough neighbors such that the local polynomial coefficients are uniquely defined. For very high order polynomial approximation, one will then use the direct neighbors, the neighbors of the neighbors, *aso...* This renders the schemes non-compact, hence less efficient.

Even though there have been attempts to design truly multidimensional finite volume schemes ([67, 65]) and to improve high order $\mathcal{FV}$ schemes for unstructured meshes [20, 19, 21], the main deficiencies remain. These deficiencies are neither cured by the very high order extensions obtained using the ENO/WENO philosophy (see [110, 111]), which are based on even more complex polynomial reconstructions that are completely annihilating hopes of efficient parallelization.

1.2.2 Discontinuous Galerkin Methods

As you may guess from their name, the Discontinuous Galerkin ($\mathcal{DG}$) methods are based on the Galerkin Finite Element theory, but allow the numerical solution to be discontinuous [14, 13]. Each element of the grid has its own degrees of freedom and do not share them with others. Interactions between elements are computed by numerical fluxes that can be rather complex, often coming from the theory of the Riemann solvers. It is today a numerical method enjoying a very wide and very active community because of its promising character. The main advantage of the method is an easy and compact generalization to high order formulation [13]. This is due to the fact that high order polynomial representation of the data is not reconstructed but defined on the elements of the grid, all containing extra *degrees of freedom*. Impressive results have already been shown [45, 44].

Unfortunately, even if local energy stability properties can be easily proved [14], the design of non-oscillatory $\mathcal{DG}$ schemes relies either on the use of $\mathcal{FV}$ limiters, which can reduce dramatically their accuracy, or, as stabilized $\mathcal{FE}$ schemes, on the use of discontinuity capturing operators [61, 46, 16]. This technique basically reduces to adding strongly dissipative terms in localized regions where the gradient of the solution is large. This approach, if on one hand allows to prove the global L^∞ stability of the solution, on the other hand does not fully guarantee its local monotonicity. More importantly, these shock-capturing ($\mathcal{SC}$) terms depend on tunable constants which are difficult to determine in a general way.

Finally, the price to pay for this discontinuous approach is a quite expensive computational cost. On Figure 1.1 is represented for the same mesh the *conformal* approach that would be used by the continuous Residual Distribution schemes and the *non-conformal* discretization used in the $\mathcal{DG}$ framework. It is clear the $\mathcal{DG}$ discretization uses more degrees of freedom. To be more rational, let us consider a mesh composed of n vertices. We can roughly estimate the number of degrees of freedom needed by a $\mathcal{DG}$ scheme and by a $\mathcal{RD}$ one. This is done in Tabular 1.1. The Residual Distribution framework presents always much less unknowns than its $\mathcal{DG}$ equivalent, especially for low order of accuracy. For 4^{th} order, it is for example 3 to 4 times cheaper. But if we look at the asymptotic behaviour with respect to the polynomial order of representation of the data, we see that both schemes need approximately the same amount of unknowns. In 2D, if k is the polynomial order of representation of the solution, a $\mathcal{RD}$ scheme needs approximately $k^2 n$ degrees of freedom when $\mathcal{DG}$ needs $(k+1)(k+2)n$. The same in 3D, both scheme needing asymptotically $k^3 n$ degrees of freedom.

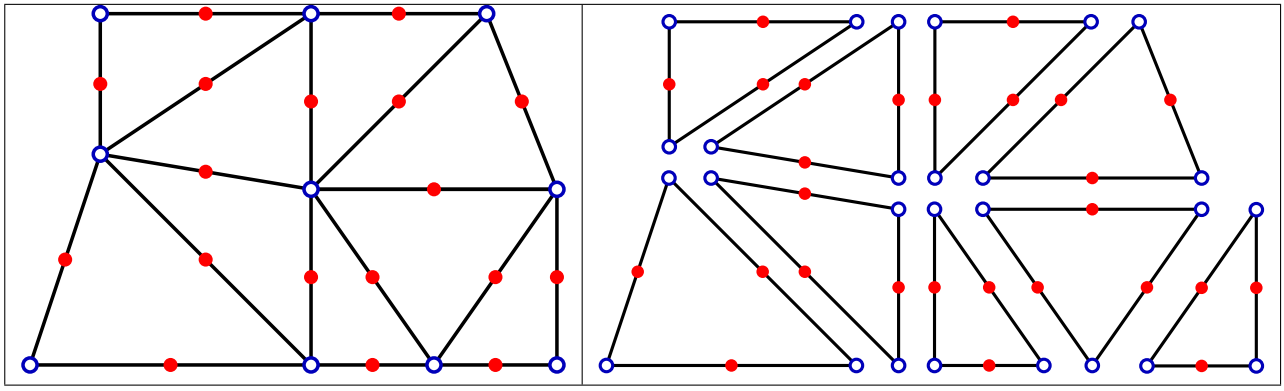


Figure 1.1: Third Order $\mathcal{RD}$ and $\mathcal{DG}$ meshes.

	2D		3D	
Order	$\mathcal{DG}$	$\mathcal{RD}$	$\mathcal{DG}$	$\mathcal{RD}$
2	$6n_s$	n_s	$24n_s$	n_s
3	$12n_s$	$4n_s$	$40n_s$	$8n_s$
4	$20n_s$	$9n_s$	$80n_s$	$27n_s$

Table 1.1: Comparison of the number of degrees of freedom needed for second, third and fourth order approximation in the case of a $\mathcal{DG}$ or a $\mathcal{RD}$ scheme.

1.2.3 Residual Distribution Schemes

The last class of methods we are presenting here is the one that is going to be used and developed through all this thesis. The *Residual Distribution Schemes* ($\mathcal{RDS}$), is a class of methods that uses a continuous representation of the variables, similarly to the standard Finite Element methods. It has been first studied by P.L. Roe in the early eighties [99] and was called at that time the *Fluctuation Splitting* methods. The ground entity is the *residual*, an integral quantity over each element, that represents the balance of information entering the element. Following some well defined rules, this residual is distributed to the nodes of the elements and by looping over this oversimplified scheme, we prove to converge toward an approximation of the exact solution of the *Conservation Law*. These methods allow to discretize all the operators of the equation at the same time and it is proved the global accuracy of the scheme is led by the *residual* computation accuracy. Furthermore, these methods can guarantee by construction the local monotonicity of the approximation. Solutions with discontinuities can then be computed without the help of any shock capturing or slope limiter term. Eventually, the distribution of the degrees of freedom used for the k^{th} order polynomial representation of the data being done inside the elements and therefore *maximum compact*, the update of the value of the solution in a given location of the mesh only uses the information stored in immediately adjacent mesh entities. This makes residual methods very compact and efficiently parallelizable.

1.3 Contribution of This Thesis

The objective of this thesis is to construct and analyze $\mathcal{RD}$ methods that are using high order polynomial representation of the data on hybrid unstructured grids. The work presented here describes an automatic non linear method allowing to build a k^{th} order ($k \in \mathbb{N}^*$) *Residual Distribution Scheme* from a first order one. In particular, the study mainly focuses on this automatic method applied on the linear *Lax-Friedrichs* scheme. This provides a monotonicity preserving k^{th} order conservative scheme that can be applied on unstructured hybrid grids for complex *Conservation Laws*. This work is widely illustrated with a large panel of test cases and the convergence order is often examined through mesh convergence curves. The efficiency of the higher order approximation is always discussed in term of accuracy as well as in term of computational time and effort. The main goal of the higher order schemes is to reach a given level of accuracy with a significantly reduced amount of nodes, such that the global computational cost is also drastically cut down. Hereafter we recall the background of our work and discuss its major contributions.

1.3.1 State of the Art at the Beginning of the Thesis

Historical Overview and Literature Survey in $\mathcal{RDS}$: An impressive bibliography is available in Mario Ricchuito's thesis [89] which have been published in May 2005. One can especially give a look at page 20 of this manuscript for an exhaustive list of the main publications on $\mathcal{RDS}$ at that time.

Since then, pretty much the same laboratories have carried on with this domain. The pioneering work of Roe has been continued by H. Nishikawa at *University of Michigan* [75, 74]. They are today focusing on solving the second order advection-diffusion equation as a first order system by introducing the gradients as additional variables. $\mathcal{RD}$ framework can then be applied to the diffusive terms, but the price to pay is a much bigger system to solve. In Italy, M. Napolitano and *al.* from *politecnico di Bari* are working on the theoretical and numerical analysis of the various $\mathcal{RD}$ schemes in their steady or unsteady version [101, 39, 102]. At *University of Leeds*, M.J. Hubbard and his team are studying both steady and unsteady cases [55, 58, 57] but are recently interested in particular in a discontinuous formulation of the Fluctuation Splitting Schemes [56, 59]. This gives even more flexibility to the formulation, but on the other hand asks to define some numerical fluxes along the edges of the mesh. At *ENSAM, Paris*, A. Lerat and his collaborators are designing a residual based scheme using a reconstructing stencil [31, 30, 32]. They are making the transition between the Finite Volume spirit and the pure compact Residual Distribution. At the *University of Wisconsin*, J.A. Rossmanith is showing interesting results [103, 104]. We can also notice that applications to more complex flow models of the $\mathcal{RD}$ framework are promising. $\mathcal{RDS}$ have been already used for multiphase flow problems [43], for the resolution of the shallow water equations [94, 92, 91, 103] and for the Magneto-Hydrodynamics (MHD) [38, 86]. Eventually, the von Karman Institute for Fluid Dynamics in Brussels, Belgium [52, 95, 63, 108, 41] under the lead of H. Deconinck and the project Scallaplix of INRIA Bordeaux-Sud-Ouest (R. Abgrall) [90, 3, 5, 96, 82, 10, 6, 84, 83, 85, 88] are still very active and have produced several collaborative results [116, 93, 91, 81].

Mature Work in September 2006: This work is the continuation of the work of Cédric Tavé [114], who was about to finish his PhD thesis when I started mine. Thus, [114] gives a fair

overview of what was already available at that time and what was not. If we look at one dimension variable problems (usually called *scalar* problems), the global progress was pretty much the same as today. It is in fact only on these simple cases that we have a real theoretical framework and this has been of course the task of the pioneering work. Scalar very high order methods were already developed with Lax-Friedrichs scheme in *Bordeaux* [8, 81, 115], and with the LDA scheme at *VKI*[2] [40, 52] but no results using more accurate approximations than quadratic polynomials had been presented. Scalar unsteady problems had also found second order solutions by different ways that are still in competition today. One can either consider the problem in a time-space domain [7, 95] or first discretize the time dependent terms and then solve the problem by $\mathcal{RDS}$ as a steady problem plus a time dependent source term [5]. For multidimensional problems (as Euler or Navier-Stokes equations), second order solutions on hybrid meshes were just produced [114], and some unsteady cases were treated [95, 7]. The treatment of the viscous terms was at the very beginning [63, 93].

1.3.2 New Developments

Higher Order Assessment: The first work of this PhD thesis was to develop a high order scalar code in order to validate the theory for very high order computations. This code is using polynomial representation of the solution up to 4^{th} order and the results are very good. We have been testing the code on several simple test cases and the general mesh convergence always get the expected slope. This proves that the theory on high order $\mathcal{RD}$ schemes is good and that the scheme we are using, based on the first order linear *Lax-Friedrichs* scheme, is able to reach this very high order convergence in seemingly all the possible scalar cases. Once this point had been verified, we could start implementing the scheme for multidimensional problems inside the Fortran platform for fluid simulations developed at INRIA Bordeaux Sud-Ouest, called "FluidBox".

Higher Order Quadrangle Treatment: At the beginning of this introduction, we were speaking about the general agreement of the community on the mandatory character of unstructured grids for their flexibility and adaptivity in the case of complex geometries. We call *Hybrid meshes*, the discretizations of the spatial domain that do not contain a unique type of element. In our case, they are built with both triangles and quadrangles. These hybrid meshes are even more interesting for complex geometries, because they are more flexible but above all, because for a given number of degrees of freedom they have up to twice as less elements.

The scalar code presented in the last paragraph has also been coded to handle with quadrangular elements. In Chapter 6, we are going to show that very high order can also be reached on hybrid meshes. Moreover, we notice that using hybrid meshes is often very interesting in term of CPU time for scalar problems: the computation of the residuals inside quadrangles is indeed more expensive, but as we already said, there are roughly twice as less elements in a hybrid mesh where a maximal number of quadrangles is used. Furthermore, the accuracy of the obtained solution is usually higher when using quadrangles, because of the higher polynomial degree of of their shape functions. Developing the high order formulation for quadrangles first on scalar problems gave us a global understanding of the difficulties of the formulation. We could then transpose the general hybrid scheme for multidimensional problems treatment into "FluidBox"

[2] Von Karman Institute, Brussels, Belgium

easily.

Code Parallelization: *compactness* is one of the major property of the Residual Distribution schemes, because it allows to parallelize the global algorithm with great efficiency. We had then to try to distribute the computation to several processors, in order to measure the real efficiency of the parallelization, but also simply to be able to run some big test cases that lasted forever when using a *sequential* method (1 processor only). The implementation of this task did not radically change our Fortran code, adding just some new routines and processor communications here and there, but its optimization is a hard challenge which is still ongoing at that moment. The parallel efficiency should be very near form 1.0 (n processors work n times faster than 1 single processor), it is not the case nowadays. Even if 2 or 4 processors are really working approximately 2 or 4 times faster than one, we cannot reach this efficiency for a growing number of processors. The mean parallel efficiency is today oscillating between 0.7 and 0.8, following the size of the treated problem.

3D Simulations: Three dimensional problems were the main argument for the code parallelization. Excluding a very small number of simple test cases, three dimensional problems require such an amount of calculations that they are almost impossible to run on a *sequential* machine. Just after the code has been parallelized, we developed a $\mathcal{RD}$ formulation for tetrahedra. We are today able to run inviscid second order simulations on any unstructured mesh composed uniquely with tetrahedra. This is illustrated in this thesis by figures representing continuous or discontinuous solutions around several types of aerodynamic objects, including a complete aircraft. Hybrid 3D mesh is indeed a next step in that branch, but the generalization of the actual code to hexahedra should not be very complex. On the contrary, taking into account the viscous phenomena seems to be a much harder challenge and it is an ongoing work inside INRIA project Bacchus.

Viscous Term Treatment: $\mathcal{RD}$ schemes are not very well suited at that moment to deal with viscous problems. The main reason is that $\mathcal{RD}$ formulation assumes the approximated quantities to be continuous, when viscous terms make use of the unknowns and their gradients. Because the unknowns are piecewise polynomial per elements, their gradients are discontinuous along the edges of the mesh. To bypass this constraint, we have been using a Finite Element Galerkin formulation for the viscous terms and coupled it with the $\mathcal{RD}$ formulation of the inviscid part of the fluid mechanics equations. We prove here that binding these two formulations together is consistent but unfortunately, it seems that high order convergence cannot be reached for fine meshes. However, the obtained solutions are satisfying, especially for coarse meshes which is a promising result for even higher order approximations.

Optimizations: here and there small improvements of the scheme are also an important part of the new developments brought by this thesis. These optimizations increase the execution speed of the code, as Jacobian matrices calculation by finite differences that requires a little bit more time than the solution we had before, but that tremendously helps the iterative convergence. We can also notice the effort of always finding the optimal number of points needed for each quadrature formula. We say optimal, because this does not always correspond to the minimal number of points. Some minimal quadrature formulas need to reconstruct the unknowns at the quadrature

points when a formula with one or to extra points makes use of already computed quantities and is therefore globally faster. This quadrature rules reduction is always done by studying the mandatory properties of the terms we are approximating. In that case, the optimization is thus not only a matter of execution speed but also a matter of memory size, as one needs less information to come to the same result. It is also important to think about next developments and to implement a code that is generic enough to integrate further steps easily, but not to much generic to keep a relative efficiency.

Finally, optimizations are indeed using a lot of development time but they are also greatly helping to find small errors in the program that are very common in our everyday work. These collateral improvements are at the end greatly helping the scheme to reach its optimal performances and sometimes also help to understand better the numerical properties of the scheme.

1.4 Structure of the Manuscript

The organization of the manuscript has been conceived keeping in mind the modeling steps which lead, starting from a physical problem, to a discrete solution verifying certain properties. In particular, the idea behind the structure of the thesis is to first present the continuous problem that needs to be solved, then to introduce the framework of a discrete space and discrete unknowns, to present theoretically and practically the discretization approach, and finally validate it on many test cases, showing at the same time some new developments. It is hoped this structure starting from the most theoretical aspects of the problem and ending by some very practical remarks is going to make clear the analytical tools that are going to be used and on what grounds some properties are claimed to be important. The text is structured as follows:

- The first part of this thesis is the most theoretical one. The goal is here to set down the whole framework in which is drown the numerical scheme we are describing in the next parts. Classical mathematical and physical concepts are recalled in those two chapters.
  - In **Chapter 2** are first presented in an as complete as possible way the mathematics of *Conservation Laws*. The goal is here to give an exhaustive overview of the ground results about the well-posedness of the problem and about the structure of the solution. Links with the physics are also given. In a second part of this chapter, we are going to recall the main ideas allowing to build the two main *Conservation Laws* that are used along this thesis: the Euler and the Navier-Stokes equations. Finally, some theoretical but also physical arguments about the boundary conditions are also discussed.
  - **Chapter 3** treats the problem of the discretization and the high order representation of the solution. It first starts by a very abstract explanation that shows the approximation of the problem is in fact just a reduction of the space of unknowns. The continuous problem living in a space of infinite dimension is recast into a discrete problem existing in finite dimensional functional space. A finite amount of degrees of freedom is needed and this introduces the concept of meshing for linear or higher order polynomial interpolation. Many useful relations and notations are introduced is this chapter. This part ends by a discussion on the advantages of the higher order formulation.

- The second part is dedicated to the Residual Distribution Schemes and their theory. We wish here to give a fair overview of what is known and what is not in the world of $\mathcal{RD}$ schemes and to detail as much as possible the practical implementation of the $\mathcal{RD}$ scheme based on the first order *Lax-Friedrichs* scheme.

  - **Chapter 4** recalls all the theoretical results needed to understand well the computation of a $\mathcal{RD}$ scheme. In order to stay clear, the problem is often reduced to a *scalar* problem or/and to a linear approximation of the data. It is unfortunately most of the time the only framework in which we are able to obtain any result. In a first section, we explain what a Residual Distribution Scheme is and where it does come from. In particular, links with other classical numerical formulations are given. In a second section are described and studied the main properties of the $\mathcal{RD}$ schemes. Consistency with the continuous solution, stability of the scheme and accuracy of the approximation are detailed and reformulated into simple properties. This chapter finally ends by a brief overview of the main Residual Distribution Schemes: N, LDA, Blended, PSI, SUPG and Lax-Friedrichs schemes.
  - In **Chapter 5**, we are much interested into the higher order formulation of the Lax-Friedrichs scheme. We here explain step by step what must be done in order to reach the steady state of a *Conservation Law* problem. First section details the high order residual computation and the limitation technique that turns any first order $\mathcal{RD}$ scheme into a high order one. Second section speaks about the problem resolution. An explicit method is described and several solutions for an implicit treatment are given. They are compared in term of efficiency. Third section deals with a convergence problem that is occurring when using the Limited *Lax-Friedrichs* scheme. We here give an explanation of the problem and propose a cure as well as a deep analysis of its practical computation. A global overview of the boundary conditions used in the following test cases is given in a fourth section. Finally, this part concludes by a summary of the effective implementation of the *Stabilized Limited Lax-Friedrichs Residual Distribution Scheme.*

- The third part of this thesis illustrates the above properties of the $\mathcal{RD}$ schemes by presenting a large panel of test cases. At the same time, it is the occasion to show the new developments that have been realized during the past three years. This being still ongoing work, the quality of the results is not always the one expected, and it is going to be honestly discussed.

  - **Chapter 6** deals with a generalization of the formulation to *hybrid* meshes. Whereas all the theoretical results of Part II are developed on triangles only, we present here a formulation adapted to quadrangles. The second section shows some numerical results. We first start by validating the hybrid meshes formulation on very simple scalar test cases. Convergence curves show a quasi perfect match with the expected results. We then go to the system case and show that most of the phenomena observed in the scalar cases are still noticed for multidimensional problems.
  - In **Chapter 7**, the matter is the extension of the scheme to three dimensional spaces. The problem is that 3D simulations are costly in terms of calculation. That is why we first begin this chapter with a detailed explanation of the parallelization of the code. An analysis of the computational speedup is also given. When this is done, we are able to run almost any kind of simulation, whatever it size can be, as soon

as we have enough processors. This allows us to present a large panel of inviscid results, starting from a very simple 3D Bump test case and finishing with a complete supersonic aircraft.

– **Chapter 8**, the last chapter of this thesis, presents a formulation and results for viscous problems. As explained earlier, there is at that moment no possible $\mathcal{RD}$ straightforward formulation for the viscous terms, because of the occurrence of the gradients of the unknowns. These viscous terms are then discretized by Finite Element Galerkin Formulation and we show in a second section that this treatment stays consistent but that the desired order of accuracy cannot be reached for finest meshes. This theory is validated on a very simple Blasius Layer test case and 2D viscous test cases are then shown.

- We finally conclude this manuscript by a summary of the content and by a global review of the new developments brought by this work. We also underline the current limitations of our approach and finally discuss some possible routes to improve and extend the presented work.

Part I

Theoretical Framework

In this part, we are about to explain theoretically the main context of this thesis: the mathematics of conservation laws, and more precisely some of the mathematics needed to solve well the problems associated to Fluid Dynamics. For clearness of our words, we will restrict our spatial domain to $\mathbb{R}^2$, or a part of $\mathbb{R}^2$. This will also greatly help the illustration of the presented ideas. When no further information is given, we are speaking about the whole $\mathbb{R}^2$. All the following ideas can be straightforwardly extended to a three-dimensional space though. Incidentally, this will be done in the appropriate part, see Chapter 7.

We first recall some useful mathematical results and techniques around Fluid Mechanics. It contains results on systems of conservation laws and mathematical description of the well known Navier-Stokes and Euler Equations. In a second Chapter, we present the techniques for the approximation of a problem applying a conservation law on a given domain. The polynomial order of the discretization is then defined. We finally explain why higher order formulation is today appealing in numerical simulation, above all in term of computation cost.

Chapter 2

Mathematics and Fluid Mechanics

The concepts described in this chapter are well known in CFD. They are recalled here for sake of completeness and to gain better understanding of the Residual Distribution Schemes ($\mathcal{RDS}$). Indeed, $\mathcal{RDS}$, the object of the thesis, as most of the schemes for hyperbolic problems, are built starting on one or several of the results presented in this chapter. Because there is always a realistic phenomenon behind a Partial Differential Equation (PDE), the link between the PDEs and the physics will also be underlined.

The following chapter is certainly not complete though, and we will try to show the results in the largest possible framework. Each of the following ideas have been demonstrated either in the scalar case or for a one-dimensional domain. In our case, we try to make these notions as clear as possible in a multidimensional system context, but this is not always possible. There are two potential reasons for that. First, no complete demonstration exists at this time in a general framework, and the concept is mathematically valid only in a one dimensional domain or for a scalar unknown. Extension to more complex situation is however often assumed. Second, a complete demonstration might exist, but the tools needed are too complex and their description would be much too long. In this case some reliable references are given. What the reader has to keep in mind is that the following ideal mathematical problems always come from a real context, and the tools developed to solve them mainly come from the physics. That means that even if no mathematical demonstration is today available, the extension of these notions on very simple cases is physically expected and then somewhere mathematically assumed.

In a first part, we set up the theoretical framework around the systems of conservation laws. We build the class of possible solutions and explain two tools needed to describe these solutions and find the only relevant one: hyperbolicity and entropy conditions. Boundary problems will also be discussed. In a second part, we present two main systems of conservation laws: the Euler equations and the Navier-Stokes equations. Because the complete formulation of these equations has always been unclear for me, I decided to start from the main conservation laws of mechanics (mass, momentum and energy conservation) and then build the expected Partial Differential System (PDS) using some physical hypothesis. This chapter is also the occasion to set down some useful notations.

2.1 Systems of Conservation Laws

2.1.1 Description

Let D be an open subset of $\mathbb{R}^m$, and $\mathbf{U}$ a vector of m variables $u_1, \ldots, u_m$. $\mathbf{U}$ is assumed to be a function from $\mathbb{R}^2 \times [0; +\infty[$ into D. We call *system of m equations of conservation laws*, the system

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} = 0, \quad \mathbf{X} = (x, y) \in \mathbb{R}^2, \quad t \geqslant 0 \tag{2.1}$$

where $\mathbf{F}$ and $\mathbf{G}$ are called the *flux-functions*. They are smooth functions from D into $\mathbb{R}^m$. We also introduce the *flux-vector* $\vec{\boldsymbol{\mathcal{F}}} = (\mathbf{F}, \mathbf{G})$, which enables us to rewrite equation (2.1) into an equivalent form

$$\frac{\partial \mathbf{U}}{\partial t} + \vec{\nabla}.\vec{\boldsymbol{\mathcal{F}}}(\mathbf{U}) = 0, \quad \mathbf{X} = (x, y) \in \mathbb{R}^2, \quad t \geqslant 0. \tag{2.1}$$

If we furthermore consider the *flux-functions* as differentiable, the system can be put into a so called *quasi linear form*

$$\frac{\partial \mathbf{U}}{\partial t} + \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla \mathbf{U}} = 0, \quad \mathbf{X} = (x, y) \in \mathbb{R}^2, \quad t \geqslant 0 \tag{2.2}$$

with $\vec{\boldsymbol{\lambda}} = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}, \frac{\partial \mathbf{G}}{\partial \mathbf{U}} \right)$, the *flux Jacobians*.

System (2.1) expresses the conservation of the quantities $u_1, \ldots, u_m$. In fact, if Ω is an arbitrary sub-domain of $\mathbb{R}^2$ and $\vec{\mathbf{n}}$ is the outward unit normal to $\partial\Omega$, the boundary of Ω, it follows from (2.1)

$$\frac{d}{dt}\left(\int_\Omega \mathbf{U} d\mathbf{X} \right) + \int_{\partial\Omega} \vec{\boldsymbol{\mathcal{F}}}(\mathbf{U}) . \vec{\mathbf{n}} \, ds = 0. \tag{2.3}$$

That means the time variation of $\int_\Omega \mathbf{U} d\mathbf{X}$ is equal to the mean flux $\vec{\boldsymbol{\mathcal{F}}}(\mathbf{U})$ entering Ω. And because the flux entering Ω is the flux going out of $\mathbb{R}^2 \backslash \Omega$, the quantities $u_1, \ldots, u_m$ are conserved inside the whole space.

2.1.2 1D Linear Riemann Problem

To understand well the resolution of such a non-linear system of conservation laws, we will first restrict our problem to a one dimensional linear equation, with Riemann initial conditions, the matrix A being constant.

$$\begin{cases} \dfrac{\partial \mathbf{U}}{\partial t} + \mathrm{A}.\dfrac{\partial \mathbf{U}}{\partial x} = 0, \quad x \in \mathbb{R}, \ t > 0 \\ \mathbf{U}(x, 0) = \mathbf{U}_l, \quad x < 0 \\ \mathbf{U}(x, 0) = \mathbf{U}_r, \quad x > 0 \end{cases} \tag{2.4}$$

If we consider A as diagonalizable, there exists $\mathcal{L}$ and $\mathcal{R}$, matrices of left and right eigenvectors respectively, such that $\mathrm{A} = \mathcal{R}\boldsymbol{\Lambda}\mathcal{L}$, with $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$. There is no restriction considering

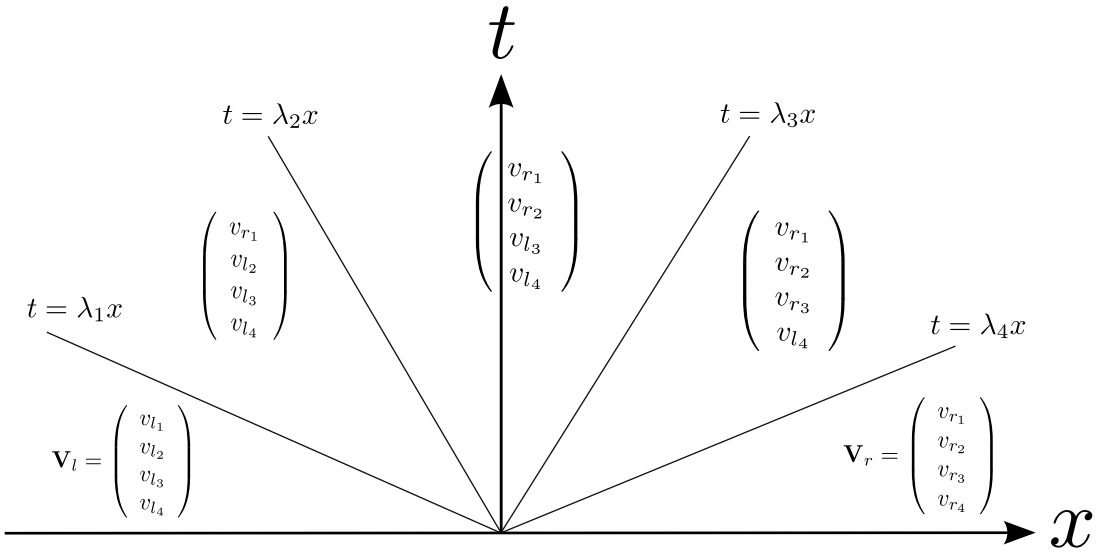


Figure 2.1: Solution of the 1D linear Riemann problem for a 4 dimensional unknown. The solution is represented in the eigenspace.

$\lambda_1, \ldots, \lambda_m$ sorted by increasing order. It is now straightforward that $\mathbf{V} = (v_1, \ldots, v_m) = \mathcal{L}\mathbf{U}$ verifies the decoupled system:

$$\begin{cases} \dfrac{\partial \mathbf{V}}{\partial t} + \mathbf{\Lambda}.\dfrac{\partial \mathbf{V}}{\partial x} = 0, \quad x \in \mathbb{R},\ t > 0 \\ \mathbf{V}(x,0) = \mathcal{L}\mathbf{U}_l = \mathbf{V}_l, \quad x < 0 \\ \mathbf{V}(x,0) = \mathcal{L}\mathbf{U}_r = \mathbf{V}_r, \quad x > 0 \end{cases} \tag{2.5}$$

One applies the theory of characteristic to each of the m independent one dimensional scalar problem and obtains:

$$v_i(x,t) = v_i(x - \lambda_i t, 0), \quad \forall (x,t) \in \mathbb{R} \times \mathbb{R}^+,\ \forall i = 1 \ldots m.$$

$\mathbf{U} = \sum_i v_i r_i$ gives then the expected solution of (2.4). An illustration of this result is represented on Figure 2.1.

By diagonalizing the system, we have decoupled the m equations and revealed m independent speeds of propagation of information, $\lambda_1, \ldots, \lambda_m$. This has allowed us to describe completely the solutions of such a problem. Generalizing this method to two dimensional problems, as in (2.2), is not as simple as in the one dimensional situation. The main drawback is that the matrices $\dfrac{\partial \mathbf{F}}{\partial \mathbf{U}}$ and $\dfrac{\partial \mathbf{G}}{\partial \mathbf{U}}$ are generally never diagonalizable in the same basis. The equations stay coupled and the system is still as hard to solve as before. But on the other hand, this gives us some very interesting properties, strongly bounded to the physics. This is described in the following. These results are fully studied in [109], [106], [4].

2.1.3 Linear Cauchy Problem with Constant Coefficients

Let us consider the following two dimensional system, the coefficient of matrices A and B being constant in space, but possibly functions of the time variable

$$\frac{\partial \mathbf{U}}{\partial t} + \mathrm{A}\frac{\partial \mathbf{U}}{\partial x} + \mathrm{B}\frac{\partial \mathbf{U}}{\partial y} = 0,\ x, y \in \mathbb{R}, t \in [0;T]\,. \tag{2.6}$$

We search the solutions of this problem for initial conditions taken in the set of tempered distributions, $\mathscr{S}'(\mathbb{R}^2)$. On this space, we can define the Fourier transform as the adjoint of the Fourier transform on the Schwartz class, $\mathscr{S}(\mathbb{R}^2)$. And the equation becomes, if $\boldsymbol{\xi}$ is the Fourier variable in space and $\hat{\mathbf{U}}$ the Fourier transform of $\mathbf{U}$:

$$\frac{\partial \hat{\mathbf{U}}}{\partial t} = -i\mathcal{A}(\boldsymbol{\xi})\hat{\mathbf{U}}, \quad \forall \boldsymbol{\xi} \in \mathbb{R}^2, t \in [0;T]$$

where we have used the notation $\mathcal{A}(\boldsymbol{\xi}) = \mathrm{A}\xi_1 + \mathrm{B}\xi_2$.

Because the Fourier transform is an isometry of $\mathcal{L}^2$, the problem (2.6) is well-posed in $\mathcal{L}^2(\mathbb{R}^2)$ if and only if

$$\sup_{\boldsymbol{\xi}\in\mathbb{R}^2} \|\exp(-i\mathcal{A}(\boldsymbol{\xi}))\| < +\infty. \tag{2.7}$$

A problem verifying (2.7) is called *weakly hyperbolic*[3]. This result can be generalized to any Sobolev space $H^s(\mathbb{R}^2)$, $s \in \mathbb{R}$ and also to $\mathscr{S}$ and $\mathscr{S}'$, see [106].

2.1.4 Hyperbolicity

It is not easy to verify the condition of weak hyperbolicity though, as it requires the calculation of the exponential of a complex matrix. That is why many sufficient conditions of $\mathcal{L}^2$ well-posedness have appeared, each one of them taking at some time the name of *hyperbolicity condition*. In the next paragraph, we give some definitions of these hyperbolicity conditions, valid even in the more complex cases (non constant coefficients) and link them to the *weak hyperbolicity condition* described above, in the case of problem (2.6), see [106].

Definition 2.1 (*Hyperbolicity*)

An operator

$$\mathcal{D} = \partial_t + \sum_{i=1}^{d} A_i(x,t)\partial_i$$

is called

- *hyperbolic, if the matrices $\mathcal{A}(\boldsymbol{\xi}) = \sum_i A_i\xi_i$ are diagonalizable with real eigenvalues for all $\boldsymbol{\xi}$ in $S^{d-1} = \left\{\boldsymbol{x} \in \mathbb{R}^d;\ \|\boldsymbol{x}\|_{\mathcal{L}^2} = 1\right\}$,*
- *constantly hyperbolic, if moreover the multiplicities of the eigenvalues remain constant as $\boldsymbol{\xi}$ covers the sphere S^{d-1},*
- *strictly hyperbolic, in the special case where all eigenvalues are real and simple for every $\boldsymbol{\xi}$.*

[3] some authors call it simply *hyperbolic* and use the term *strongly hyperbolic* for what we will call *hyperbolic*

Definition 2.2 (*Symmetrizability*)

Operator $\mathcal{D}$ is symmetrizable if there exists a symmetric positive-definite matrix S_0, such that every $S_0 A_i$ is symmetric.

Property 2.3

If an operator is symmetrizable or constantly hyperbolic, then it is weakly hyperbolic.

Proof: If we can write $\mathcal{A}(\boldsymbol{\xi}) = \mathrm{P}(\boldsymbol{\xi})^{-1}\mathrm{D}(\boldsymbol{\xi})\mathrm{P}(\boldsymbol{\xi})$ with $\mathrm{D}(\boldsymbol{\xi})$ a real diagonal matrix, we have

$$\|\exp(-i\mathcal{A}(\boldsymbol{\xi}))\| \leqslant \|\mathrm{P}(\boldsymbol{\xi})\|\,\|\mathrm{P}(\boldsymbol{\xi})^{-1}\|\,\|\exp(-i\mathrm{D}(\boldsymbol{\xi}))\|$$

And condition (2.7) is fulfilled when the *conditioning* $\|\mathrm{P}(\boldsymbol{\xi})\|\,\|\mathrm{P}(\boldsymbol{\xi})^{-1}\|$ of P is bounded independently of $\boldsymbol{\xi}$.

In the case of a symmetrizable system, S_0^{-1} admits a unique symmetric positive-definite square root R and one has:

$$\mathcal{A}(\boldsymbol{\xi}) = \mathrm{R}(\mathrm{RS}_0\mathcal{A}(\boldsymbol{\xi})\mathrm{R})\mathrm{R}^{-1}$$

The matrix $\mathrm{RS}_0\mathcal{A}(\boldsymbol{\xi})\mathrm{R}$ is symmetric and diagonalizable in an orthogonal basis and may be written as $\mathrm{Q}(\boldsymbol{\xi})^{\mathrm{T}}\mathrm{D}(\boldsymbol{\xi})\mathrm{Q}(\boldsymbol{\xi})$. We now have :

$$\|\mathrm{P}(\boldsymbol{\xi})\|\,\|\mathrm{P}(\boldsymbol{\xi})^{-1}\| = \|\mathrm{Q}(\boldsymbol{\xi})\mathrm{R}^{-1}\|\,\|\mathrm{RQ}(\boldsymbol{\xi})^{\mathrm{T}}\| = \|\mathrm{R}^{-1}\|\,\|\mathrm{R}\|,$$

a number independent of $\boldsymbol{\xi}$.

In the case of a constantly hyperbolic operator, the eigenspaces depend continuously on $\boldsymbol{\xi}$. Then for any $\boldsymbol{\xi}_0 \in \mathrm{S}^{d-1}$, there exists a neighborhood of $\boldsymbol{\xi}_0$ on which a choice of $\mathrm{P}(\boldsymbol{\xi})$ depends continuously on $\boldsymbol{\xi}$, and is thus bounded. And as the sphere S^{d-1} is compact, it is covered by a finite number of such neighborhoods. There now exists $C \in \mathbb{R}^+$ such that $\forall \boldsymbol{\xi} \in \mathrm{S}^{d-1}$, $\frac{1}{C} \leqslant \|\mathrm{P}(\boldsymbol{\xi})\| \leqslant C$. We have found a choice of the diagonalizing matrix, possibly not continuous, but which conditioning is bounded. ■

We finish this paragraph with the following theorem showing that in a constant coefficient symmetrizable hyperbolic system, the speed of propagation of the information is finite and bounded by the maximal spectral radius of the matrix $\mathcal{A}$. This result can be extended to any symmetrizable hyperbolic systems, as shown in [106].

Consider again equation (2.6) and use the notation, $\forall \boldsymbol{\xi} \in S^1$, $\mathcal{A}(\boldsymbol{\xi}) = \mathrm{A}\xi_1 + \mathrm{B}\xi_2$. If our system is symmetrizable, there exists a *s.p.d* constant matrix S_0 such that $S_0\mathrm{A}$ and $S_0\mathrm{B}$ are symmetric matrices. The system

$$S_0\frac{\partial \mathbf{U}}{\partial t} + S_0\mathrm{A}\frac{\partial \mathbf{U}}{\partial x} + S_0\mathrm{B}\frac{\partial \mathbf{U}}{\partial y} = 0 \tag{2.8}$$

can easily be transformed into a symmetric system using the variable $\mathbf{V} = S_0^{1/2}\mathbf{U}$. We therefore define the *characteristic polar envelope*

$$\mathit{Char} = \left\{(\boldsymbol{\xi}, \lambda) \in S^1 \times \mathbb{R}^+;\ \det\left(S_0\mathcal{A}(\boldsymbol{\xi}) + \lambda I_m\right) = 0\right\},$$

and for each point $(\mathbf{X}, T) \in \mathbb{R}^2 \times \mathbb{R}^+$, the *dependence cone*

$$K(\mathbf{X}, T) = \left\{(\mathbf{x}, t) \in \mathbb{R}^2 \times [0; T];\ \lambda(t - T) + (\mathbf{x} - \mathbf{X}).\boldsymbol{\xi} \leqslant 0, \forall (\boldsymbol{\xi}, \lambda) \in \mathit{Char}\right\}.$$

$K(\mathbf{X}, T)$ is the intersection of the half-spaces passing through $(\mathbf{X}, T)$ with *outward* normal $(\boldsymbol{\xi}, \lambda)$. It is then a convex cone with basis $(\mathbf{X}, T)$, and its boundary admits almost everywhere a tangent plane which equation is: $\lambda(t - T) + (\mathbf{x} - \mathbf{X}).\boldsymbol{\xi} = 0$ for some $(\boldsymbol{\xi}, \lambda) \in \mathit{Char}$, λ being necessarily maximal. The section of $K(\mathbf{X}, T)$ at time t is denoted by $\omega(t)$ and we have the following theorem:

Theorem 2.4 (*Finite Propagation Speed*)

If $\mathbf{V}_{|\omega(0)} \equiv 0$ *then* $\mathbf{V}(\mathbf{x},t) = 0, \quad \forall (\mathbf{x},T) \in K(\mathbf{X},T)$

Proof: If we take the scalar product of equation (2.8) by $\mathbf{V} = S_0^{1/2}\mathbf{U}$, we obtain the following additional conservation law (viewed as an energy identity)

$$\frac{\partial}{\partial t}\|\mathbf{V}\|_{2,m}^2 + \frac{\partial}{\partial x}\left\langle \mathrm{A}'\mathbf{V},\mathbf{V}\right\rangle_m + \frac{\partial}{\partial y}\left\langle \mathrm{B}'\mathbf{V},\mathbf{V}\right\rangle_m = 0 \tag{2.9}$$

where the notation $< .,. >_m$ is used for the canonical scalar product in $\mathbb{R}^m$ and matrices A' and B' are $\mathrm{A}' = S_0^{1/2}\mathrm{A}S_0^{-1/2}$, $\mathrm{B}' = S_0^{1/2}\mathrm{B}S_0^{-1/2}$.

For $0 < \varepsilon < T$, let us consider the truncated cone

$$\mathscr{K}(0,\varepsilon) = \{(\mathbf{x},t) \in K(\mathbf{X},t);\ 0 < t < T - \varepsilon\}$$

and integrate relation (2.9) over $\mathscr{K}(0,\varepsilon)$ (See Figure (2.2)). On the top (*resp.* bottom) of the truncated cone, the outward normal is the positive (*resp.* negative) axis of the time component. On the side, as we already showed it, there exists almost everywhere a normal which is $(\boldsymbol{\xi},\lambda) \in$ *Char*, λ being maximal in the direction $\boldsymbol{\xi}$. Thus we have:

$$\begin{aligned}\int_{\mathscr{K}(0,\varepsilon)} \vec{\nabla}.\begin{pmatrix}\langle I_m\mathbf{V},\mathbf{V}\rangle_m\\ \langle \mathrm{A}'\mathbf{V},\mathbf{V}\rangle_m\\ \langle B'\mathbf{V},\mathbf{V}\rangle_m\end{pmatrix} d\mathbf{x}dt &= \int_{\omega(t-\varepsilon)}\|\mathbf{V}\|_{2,m}^2 d\mathbf{x} - \int_{\omega(0)}\|\mathbf{V}\|_{2,m}^2 d\mathbf{x}\\ &\quad + \int_{\text{side}}\left\langle\left(\mathrm{A}'\xi_1 + \mathrm{B}'\xi_2 + \lambda I_m\right)\mathbf{V},\mathbf{V}\right\rangle d\mathbf{x}dt\\ &= 0\end{aligned}$$

But as for all $\boldsymbol{\xi}$, λ is maximal in the direction $\boldsymbol{\xi}$, matrix $\mathrm{A}'\xi_1 + \mathrm{B}'\xi_2 + \lambda I_m$ has only positive eigenvalues and the term integrated on the side of the cone is positive. That means no information enters the cone. And finally, if $\mathbf{V}$ is identically null on the bottom it is straightforward that it is null everywhere in $\mathscr{K}(0,\varepsilon)$. ε being arbitrarily small, $\mathbf{V}_{|K(\mathbf{X},T)} \equiv 0$. ■

This result shows that in the case of constant coefficients matrices, for any $(\mathbf{X},t)$ in the space-time domain, we can define a dependence cone, function of the eigenvalues of $\mathcal{A}(\boldsymbol{\xi})$ in all space direction $\boldsymbol{\xi}$. We then know that the value of the solution at point $(\mathbf{X},t)$ only depends on the value of the solution inside the cone because no information crosses the boundary of this cone. That demonstrates that in symmetrizable constant coefficients systems the speed of propagation of compactly supported initial condition is finite and bounded by the biggest eigenvalue of $\mathcal{A}(\boldsymbol{\xi})$, $\boldsymbol{\xi}$ covering S^{d-1}.

This result can actually be extended to constant hyperbolic problems and for systems with non constant coefficient matrices. The mathematical tools needed to reach this goal are rather complex though, and that is why we just refer to the book of Benzoni-Gavage [106].

2.1.5 Weak Solutions and the Rankine-Hugoniot Conditions

Another main feature of systems of conservation laws is they do not admit in general classical solutions (at least $\mathcal{C}^1$) over the whole space-time domain. This is true even for very regular initial conditions. In other words, for a given system and an - let say $\mathbb{C}^\infty$ - initial condition, there might exist a time T^* such that $\forall t \geqslant T^*$, the solution $\mathbf{U}$ of system (2.1) is not continuous in space. Let us illustrate this with the very simplest classical example: *the Burger equation.*

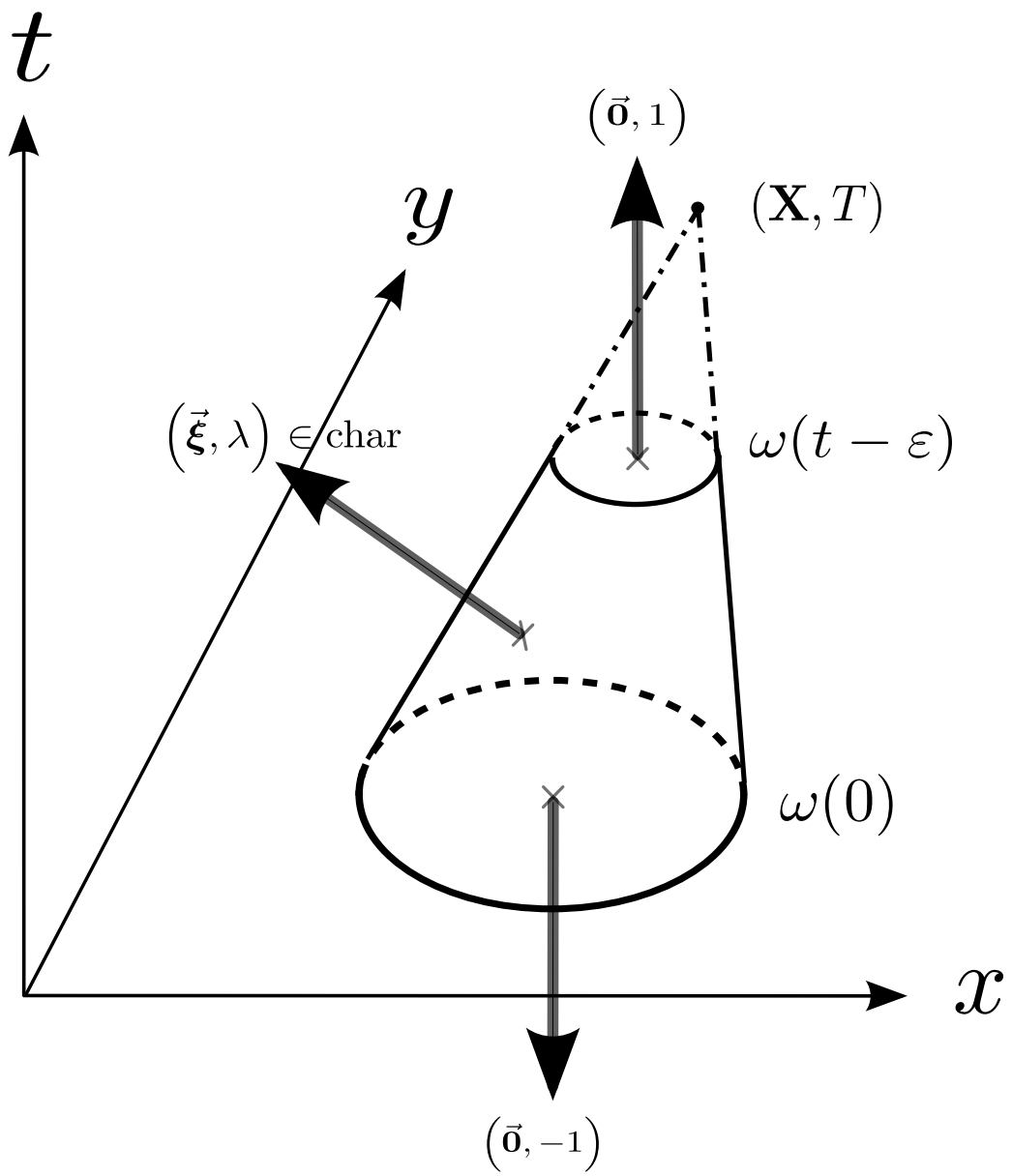


Figure 2.2: Dependence cone for point $(\mathbf{X}, T)$. The propagation is anisotropic. $\mathscr{K}(0, \varepsilon)$ is the part of the cone between the two surfaces $\omega(0)$ and $\omega(t-\varepsilon)$. $(\boldsymbol{\xi}, \lambda)$ is a normal to the side surface. It is an element of *Char*, with λ being maximal.

We consider the following scalar ($m = 1$) one-dimensional problem

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} = 0, & x \in \mathbb{R},\ t > 0 \\ u(x,0) = u_0(x), & x \in \mathbb{R} \end{cases} \tag{2.10}$$

It is a classical calculation to show that the solution is constant along the characteristic curves, and that these characteristic curves are straight lines which constant slopes depend on the initial data. The characteristic line passing through point $(x_0, 0)$ is defined by the equation:

$$x = x_0 + u_0(x_0).t$$

This is illustrated on Figure 2.3 for the initial condition

$$u_0(x) = \begin{cases} 1, & x \leqslant 0, \\ 1 - x, & 0 \leqslant x \leqslant 1, \\ 0, & x > 1. \end{cases} \tag{2.11}$$

This is of course not a very regular initial condition, but we took this one for sake of simplicity. The result would be exactly the same with any regular decreasing initial condition. As one can see on Figure 2.3, all characteristics curves generated in $[0;1]$ intersect at point $(1,1)$. That means that at this point of the space-time domain, the solution u can take any value between 0 and 1, and thus cannot be continuous here. In order to be able to solve problem (2.1), we must then consider a weaker definition of a solution. Instead of seeking our solution in the space of regular functions, we are going to define the solutions in the space of the distributions.

Definition 2.5 (*Weak Solution*)

Let $\mathbf{U}_0$ be a vector of m bounded function in $\mathbb{R}^2$. A function $\mathbf{U} \in \mathcal{L}^\infty(\mathbb{R}^2 \times [0;+\infty[)^m$ is called a weak solution of problem (2.1) with initial condition $\mathbf{U}_0$, if $\mathbf{U}(x,t) \in D$ a.e. and satisfies for any $\mathcal{C}^1$ function $\boldsymbol{\varphi}$ with compact support in $\mathbb{R}^2 \times [0;+\infty[$

$$\int_0^\infty \int_{\mathbb{R}^2} \left(\mathbf{U}.\frac{\partial \boldsymbol{\varphi}}{\partial t} + \vec{\boldsymbol{\mathcal{F}}}(\mathbf{U}).\overrightarrow{\nabla_{\mathbf{x}}\boldsymbol{\varphi}} \right) d\mathbf{x}\, dt + \int_{\mathbb{R}^2} \mathbf{U}_0(x).\boldsymbol{\varphi}(x,0) d\mathbf{x} = 0. \tag{2.12}$$

Remark 2.6

If $\mathbf{U}$ is a $\mathcal{C}^1$ solution of problem (2.1), it is of course a weak solution of this problem in the above sense.

A characterization of the *weak solutions* of a system of conservation is given by the following well known theorem. One can read [48] or [49] for a proof.

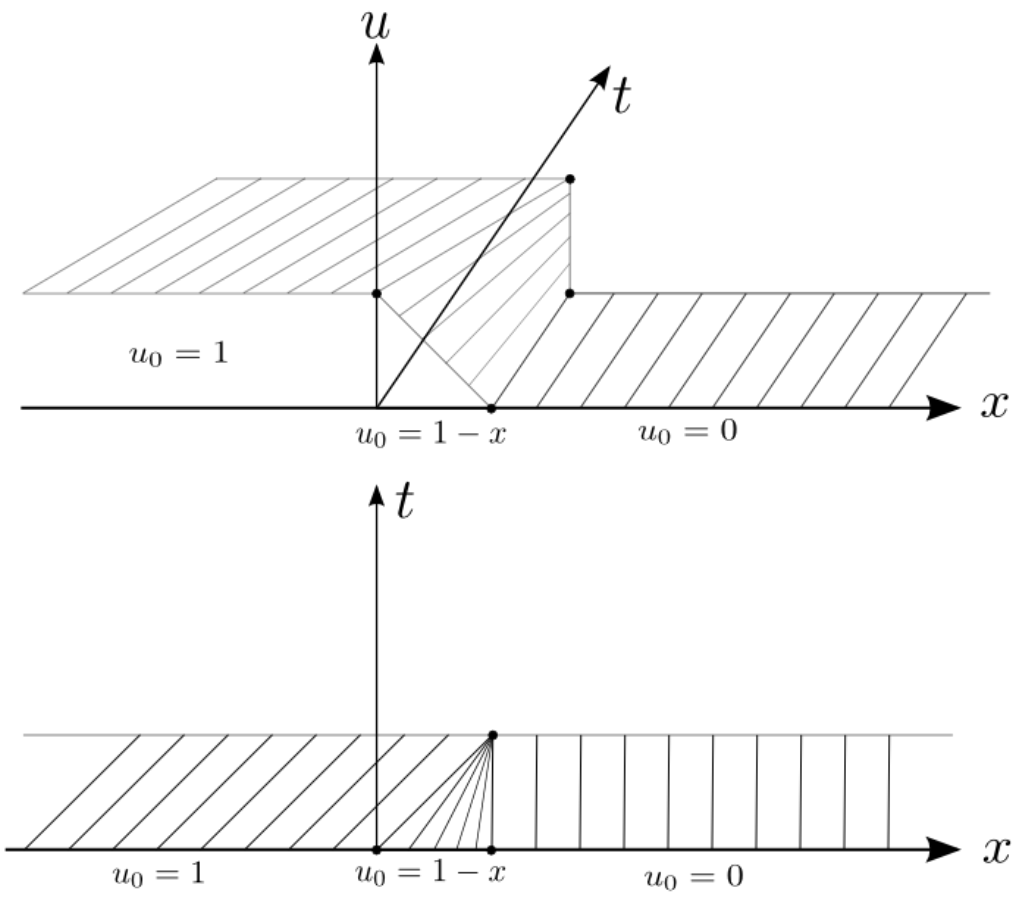


Figure 2.3: Solution of the 1D scalar Burger equation (2.10) with initial conditions (2.11). All the characteristics meet at point $(1, 1)$ and the solution cannot be continuous there.

Theorem 2.7 (*Rankine-Hugoniot*)

$\mathbf{U}$ is a piecewise $\mathcal{C}^1$ solution of problem (2.1) in the sense of distribution on $\mathbb{R}^2 \times [0; +\infty[$ if and only if:

(i) $\mathbf{U}$ is a classical solution of (2.1) in the domains where it is $\mathcal{C}^1$;

(ii) along the surfaces of discontinuity, $\mathbf{U}$ satisfies the vectorial jump condition:

$$(\mathbf{U}_+ - \mathbf{U}_-)n_t + \left(\overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U}_+) - \overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U}_-)\right).\overrightarrow{\mathbf{n_x}} = \overrightarrow{\mathbf{0}}_m \tag{2.13}$$

where $(n_t, \overrightarrow{\mathbf{n_x}})$ is a normal to the surface of discontinuity and $\mathbf{U}_+$ and $\mathbf{U}_-$ denotes the limit value of the solution at the discontinuity.

2.1.6 Non Uniqueness of the Weak Solution

This section deals with another problem of our systems of conservation laws: the non uniqueness of a weak solution. As before, we are going to illustrate it by means of a classical example: the scalar Riemann problem for the Burger's equation

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} = 0, \ (x,t) \in \mathbb{R}_x \times \mathbb{R}_t, \\ u_0(x) = \begin{cases} u_l, & x < 0, \\ u_r, & x > 0. \end{cases} \end{cases} \tag{2.14}$$

We suppose that $u_l \neq u_r$. The Rankine-Hugoniot condition (2.13) shows that we obtain a weak solution of (2.14) by propagating the discontinuity at speed $s = (u_l + u_r)/2$:

$$u(x,t) = \begin{cases} u_l, & x < st, \\ u_r, & x > st. \end{cases}$$

But for $u_l < u_r$, because the characteristic curves are never intersecting, one can also build a continuous solution

$$u(x,t) = \begin{cases} u_l, & x < u_l t, \\ x/t, & u_l t < x < u_r t, \\ u_r, & x > u_r t. \end{cases}$$

And worst, for any a between u_r and u_l, we have a family of admissible solutions:

$$u(x,t) = \begin{cases} u_l, & x < s_1 t, \\ -a, & s_1 t < x < 0, \\ a, & 0 < x < s_2 t, \\ u_r, & x > s_2 t. \end{cases}$$

with discontinuity propagating at speeds $s_1 = 0.5(u_l - a)$ and $s_2 = 0.5(u_r + a)$.

2.1.7 Entropy Solution

The mathematical problem of existence and uniqueness of the solution of problem (2.1) is at that point in a dead end. We have seen that some well chosen cases do not admit classical solutions. We have then extended the space of existence of the solutions to a larger class of functions and obtained an infinity of solutions. But realistic problems admit only one reproducible solution. We have now to find a criterion that will sort the *weak solutions* in order to pick the only physically relevant one. This criterion is based on the concept of *the entropy* that we introduce now.

In nature, there is always a dissipation phenomenon: no real problem coming from the physics is perfectly reversible. Let us consider the following one-dimensional scalar dissipative problem, $\varepsilon > 0$ being a small viscous parameter

$$\frac{\partial u_\varepsilon}{\partial t} + \operatorname{div}\left(f(u_\varepsilon)\right) = \varepsilon \Delta u_\varepsilon, \tag{2.15}$$

with initial condition $u_\varepsilon(x,0) = u_{0\varepsilon} \to u_0$ when $\varepsilon \to 0$. We still suppose that u_ε takes its value in D, a sub-domain of $\mathbb{R}$ ($m = 1$). If f is regular enough (Lipschitz), it has been shown that for any positive ε, for any initial condition $u_{0\varepsilon} \in \mathcal{L}^2$, equation (2.15) admits a unique solution. This result is partly demonstrated in [48]. One can also find a partial extension to systems (only existence in the space of distribution) in [51] and [47].

If we now consider a sequence of ε tending toward zero, and a sequence of solutions of (2.15) such that :

a) $\exists C \in \mathbb{R},\ \|u_\varepsilon\|_\infty \leqslant C$, independently of ε;

b) $u_\varepsilon \underset{\varepsilon \to 0}{\longrightarrow} u$ almost everywhere in $\mathbb{R}^2 \times [0; +\infty[$,

then u is a weak solution of (2.1) in its scalar form for initial condition u_0, and moreover verifies, in the sense of distributions, any inequality of the form:

$$\frac{\partial}{\partial t} S(u) + \operatorname{div}\left(\mathscr{G}(u)\right) \leqslant 0, \tag{2.16}$$

where

(i) $S : \mathrm{D} \to \mathbb{R}$ is a smooth convex function;

(ii) $\mathscr{G}$ is a vector of 2 scalar smooth functions such that

$$S'(u) f_j'(u) = \mathscr{G}_j'(u), \quad j = 1, 2. \tag{2.17}$$

$(S, \mathscr{G})$ is called *a pair of Entropy-Flux*, S an *entropy function* and $\mathscr{G}$ an *entropy flux*. This result may also be extended to systems, see [48] page 27. If we now take relation (2.2) and multiply it by $S'(\mathbf{U})$, quick calculation shows that $\mathbf{U}$ satisfies an additional conservation relation

$$\frac{\partial}{\partial t} S(\mathbf{U}) + \vec{\nabla}.\mathscr{G}\left(\mathbf{U}\right) = 0, \quad \mathbf{X} = (x, y) \in \mathbb{R}^2, \quad t \geqslant 0. \tag{2.18}$$

The next important result is available in the scalar case for *entropy solutions*. It is the main result of chapter 2 of [48] were one can find a complete and rigorous demonstration.

Theorem 2.8 (*Kruzhkov*)

A weak solution u of a scalar conservation law with a bounded initial condition $u_0 \in \mathcal{L}^\infty(\Omega)$, verifying relation (2.16) for any pair of Entropy-Flux $(S, \mathscr{G})$ is unique and called ***the entropy solution****. Moreover this solution is bounded*

$$\forall T > 0, \quad u \in \mathcal{L}^\infty(\Omega \times [0;T]).$$

We were looking for the solution of a sort of idealistic problem (without viscosity), and we found that the only relevant solution is the one coming from the physics. By "the one coming from the physics", we mean the solution being the limit of a sequence of solutions of an associated more realistic perturbed problem for a decreasing viscosity coefficient ε. But we do not have to construct such a sequence of realistic solutions in order to find our sought solution. We can simply sort the solution of the idealistic problem with an entropy criterion. Entropy is then a set of additional conservation relations the solution of problem (2.1) has to verify.

What one has to remember is that we started with a system verifying just the first principle of thermodynamics (conservation of the variables), and could find either no solutions (in the class of regular ones) or an infinity (in a weaker class of functions). But by looking at the physics intrinsic to the problem, we found the system of conservation laws is well-posed when it comes with an entropy condition. That is the second principle of thermodynamics and that binds strongly the mathematical problem to the one that comes from the physics.

In the following, we are not much going to speak about entropy. It is a very important notion though. In fact it is rather hard to define a criterion ensuring the solution of a numerical scheme will converge toward the entropy solution of the associated Partial Differential System (PDS). It is besides not always the case as one can build numerical schemes that converge toward a bad solution in the case of problem (2.14). For example, let us consider the case when $u_l = -1$ and $u_r = 1$. As we have seen, the characteristic straight lines never intersect and the solution is two constant plateau separated by a fan between the lines $t = -x$ and $t = x$. We now apply the finite difference second order consistent **Mac Cormack** method defined by:

$$\left\{ \begin{array}{lcl} u_i^* & = & u_i^n - \frac{\Delta t}{h}\left(f(u_{i+1}^n) - f(u_i^n)\right) \\ u_i^{n+1} & = & \frac{1}{2}\left(u_i^n + u_i^*\right) - \frac{\Delta t}{2h}\left(f(u_i^*) - f(u_{i-1}^*)\right) \end{array} \right. \tag{2.19}$$

with Δt and h being the time and spatial steps respectively and f being the equation flux, $f(u) = u^2/2$ in the case of the Burger equation. We see on Figure 2.4, that for any time or spatial step, the solution at time step n is identically reproduced in u^* and thus in u^{n+1}. At the end, we obtain a solution with a shock which equation is $x = 0$ and this is actually a weak solution of problem (2.14) as $s = (u_r + u_l)/2 = 0$. The scheme has converged toward a weak solution of the problem which is not the entropy solution. And making the problem more complex does not help: there exists multidimensional test cases for which unphysical shocks may appear. A general criterion ensuring a scheme is always converging toward the entropy solution is then still needed.

A last interesting result is the following theorem of Mock.

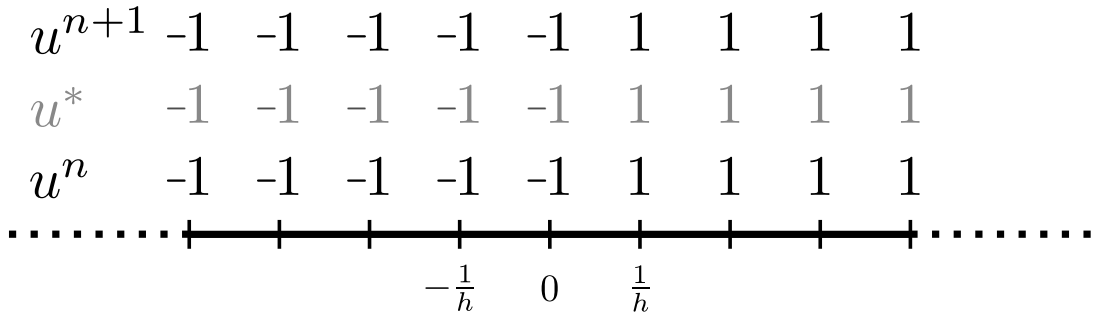


Figure 2.4: **Mac Cormack** second order consistent finite difference scheme applied to equation (2.14) with initial boundary conditions $u_l = -1$ and $u_r = 1$.

Theorem 2.9 (*Mock*)

Let $S : D \longrightarrow \mathbb{R}$ be a smooth convex function. A necessary and sufficient condition for S to be an entropy for system (2.1) is that the $m \times m$ matrices $S''(\mathbf{U})\boldsymbol{F}'(\mathbf{U})$ and $S''(\mathbf{U})\boldsymbol{G}'(\mathbf{U})$ are symmetric.

Proof: Let first assume S is a convex entropy for system (2.1). Then there exists a vector of smooth functions $\mathscr{G}$, such that $S'(\mathbf{U})\mathbf{F}'(\mathbf{U}) = \mathscr{G}_1'(\mathbf{U})$ and $S'(\mathbf{U})\mathbf{G}'(\mathbf{U}) = \mathscr{G}_2'(\mathbf{U})$. Let consider only the first relation and differentiate its k^{th}-line with respect to u_j. We obtain :

$$\frac{\partial}{\partial u_j}\left(\sum_{i=1}^{m}\frac{\partial \mathbf{F}_i}{\partial u_k}\frac{\partial S}{\partial u_i} - \frac{\partial \mathscr{G}_1}{\partial u_k}\right) = 0 \tag{2.20}$$

$$\Leftrightarrow \frac{\partial^2 \mathscr{G}_1}{\partial u_k u_j} - \sum_{i=1}^{m}\frac{\partial^2 \mathbf{F}_i}{\partial u_k u_j}\frac{\partial S}{\partial u_i} = \sum_{i=1}^{m}\frac{\partial \mathbf{F}_i}{\partial u_k}\frac{\partial^2 S}{\partial u_i u_j}. \tag{2.21}$$

Since the left-hand side is symmetric in the k and j variables, it holds for the right-hand side, and we have :

$$\sum_{i=1}^{m}\frac{\partial \mathbf{F}_i}{\partial u_k}\frac{\partial^2 S}{\partial u_i u_j} = \sum_{i=1}^{m}\frac{\partial \mathbf{F}_i}{\partial u_j}\frac{\partial^2 S}{\partial u_i u_k}. \tag{2.22}$$

This means exactly the matrix $S''(\mathbf{U})\mathbf{F}'(\mathbf{U})$ is symmetric. And same argument holds for the second coordinate $\mathbf{G}(\mathbf{U})$.

Conversely, assuming (2.22), we have

$$\frac{\partial}{\partial u_j}\left(\sum_{i=1}^{m}\frac{\partial \mathbf{F}_i}{\partial u_k}\frac{\partial S}{\partial u_i}\right) = \sum_{i=1}^{m}\left(\frac{\partial \mathbf{F}_i}{\partial u_k}\frac{\partial^2 S}{\partial u_i \partial u_j} + \frac{\partial^2 \mathbf{F}_i}{\partial u_k \partial u_j}\frac{\partial S}{\partial u_i}\right) \tag{2.23}$$

$$= \frac{\partial}{\partial u_k}\left(\sum_{i=1}^{m}\frac{\partial \mathbf{F}_i}{\partial u_j}\frac{\partial S}{\partial u_i}\right). \tag{2.24}$$

If our spatial domain is *contractible* (there is a homotopy that continuously deforms Ω to a point), it follows from Poincaré's lemma that there exists a function $\mathscr{G}_1$, such that

$$\frac{\partial \mathscr{G}_1}{\partial u_k} = \sum_{i=1}^{m}\frac{\partial \mathbf{F}_i}{\partial u_k}\frac{\partial S}{\partial u_i}, \quad \forall k \in [\![1, m]\!]$$

And because once more the same arguments hold for $\mathbf{G}$, S is an entropy function associated with the entropy fluxes $\mathscr{G}_1$ and $\mathscr{G}_2$. ■

This result adds an extra value to the concept of entropy. Not only the existence of an entropy can ensure the well-posedness of our problem in a certain class of non-continuous functions, but it also enforces the propagation of the information at a finite speed, because the existence of an entropy for a system is equivalent to the property of symmetrizability for this same system, and thus allows to build a dependence cone for each point of the space-time domain that cannot be crossed by any information, as demonstrated in theorem (2.4). Eventually, the symmetrizability of the system ensures that the initial value problem is well-posed in the $\mathcal{L}^2$ norm [24]. The solution depends continuously on the initial condition and it is thus possible to build a numerical scheme. In fact, if this was not the case, one could achieve a very ill-posed solution were the obtained numerical situation would depend on the round off of the machine. Furthermore, [24] also shows that the symmetrized problem (2.8) is well-posed in $\mathcal{L}^p$, $p \neq 2$, $1 \leqslant p \leqslant +\infty$, if and only if the symmetric Jacobian matrices $S_0 \mathrm{A} S_0^{-1}$ and $S_0 \mathrm{B} S_0^{-1}$ commute. This being generally not the case, looking for the entropy solution of a system of conservation laws is a well-posed problem only in $\mathcal{L}^2$.

2.1.8 Maximum Principle

We can go further in the analysis of the solution and show that the entropy solution of a *conservation law* respects a *maximum principle*. This prevents the sudden appearence of a new global extrema in the solution. This property is very important from a numerical point of view, because one would need it to be transpose to the solution of the numerical scheme used and hence ensure the $\mathcal{L}^\infty$ stability of the scheme and prevent the approximated solution to explode within a finite time. The next theorem comes from [48] and is there explained and demonstrated in details. It is true only in the scalar case but for any dimension of the spatial domain. It claims the *entropy solution* is bounded in $\mathcal{L}^\infty$ norm and monotonically depends on the initial condition.

Theorem 2.10

Let u_0 belong to $\mathcal{L}^\infty(\mathbb{R}^2)$. Then the unique entropy solution u of problem

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} + \frac{\partial g(u)}{\partial y} = 0, \qquad & \boldsymbol{x} = (x, y) \in \mathbb{R}^2, \quad t \geqslant 0 \\ u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), \qquad & \boldsymbol{x} \in \mathbb{R}^2 \end{aligned}$$

with smooth scalar fluxes f and g, belongs to $\mathcal{L}^\infty(\mathbb{R}^2 \times [0, T])$. This solution satisfies for almost all $t \geqslant 0$,

i)

$$\|u(., t)\|_{\mathcal{L}^\infty(\mathbb{R}^2)} \leqslant \|u_0\|_{\mathcal{L}^\infty(\mathbb{R}^2)}$$

ii) If v is also the entropy solution of (2.25) associated with initial condition v_0, we have

$$u_0 \geqslant v_0 \text{ a.e.} \quad \Longrightarrow \quad u(., t) \geqslant v(., t) \text{ a.e.}$$

2.1.9 Boundary Conditions

Let us come back to a one dimensional system, and moreover assume that the Jacobian matrices of the flux have still constant coefficients. Instead of solving our system on $\mathbb{R}_x$ entirely, we now would like to restrict our spatial domain to $\Omega \subset \mathbb{R}_x$, let say to $\mathbb{R}_x^+$ for example. This implies we have to give a boundary condition over the half-straight-line $\{(0,t); t > 0\}$. The study of the admissible boundary conditions is the aim of this paragraph.

Our new problem can be written as :

$$\begin{array}{l} \text{Find} \quad \mathbf{U} \in \mathrm{D} \subset \mathbb{R}^m, \text{ such that} \\ \left\{ \begin{array}{ll} \dfrac{\partial \mathbf{U}}{\partial t} + \mathrm{A}\dfrac{\partial \mathbf{U}}{\partial x} = 0, & (x,t) \in \mathbb{R}^+ \times \mathbb{R}^+ \\ \mathbf{U}(x,0) = \mathbf{U}_0(x), & x \in \mathbb{R}^+ \text{ (Initial Condition)} \\ \mathbf{U}(0,t) = \mathbf{V}_0(t), & t \in \mathbb{R}^+ \text{ (Boundary Condition)} \end{array} \right. \end{array} \tag{2.25}$$

We still denote by $\lambda_1, \ldots, \lambda_m$ the eigenvalues of A sorted by increasing modulus and by $r_1, \ldots, r_m$ the associated eigenvectors. For any object z of $\mathbb{R}^m$, let $z_1, \ldots, z_m$ be its components in the eigenbasis. Furthermore, we define $p \in [\![1, m]\!]$ as the index such that $\lambda_p < 0 \leqslant \lambda_{p+1}$, p being possibly 1 or m. We also use the notation : $\mathbf{U}(0+,t) = \lim_{x \to 0+} \mathbf{U}(x,t)$.

Using the theory of characteristics developed in section 2.1.2 dealing with the 1D Riemann Problem, we clearly see that for any $i \leqslant p$, for any $t \geqslant 0$, $\mathbf{U}_i(0+,t)$ is defined by : $\mathbf{U}_i(0+,t) = \mathbf{U}_{0_i}(-\lambda_i t)$. This means that enforcing a boundary condition in the direction of $r_i, (i = 1 \ldots p)$ is useless as the component of $\mathbf{U}$ in these directions are already defined by the initial condition. If we look at it from a purely mathematical point of view, there is no way of verifying continuously the condition $\mathbf{U}(0,t) = \mathbf{V}_0(t)$ for $t > 0$. On the other hand, if we take a point close enough to the space boundary, we see that its components of higher index depend only on the value of the function on the space boundary. Thus, if $\mathbf{V}_{0_i}$ is not defined for $i > p$, our problem is ill-posed. The above discussion is illustrated on Figure 2.5. It can be summarized as follows:

Property 2.11

In the case of a one dimensional system with constant coefficients Jacobians, the boundary conditions must be enforced on and only on the components of the solution which associated characteristics are entering the domain.

In fact in the numerical case, if we impose some information on the outgoing characteristics, it will be blown out of the computational domain at any time step and will not interfere with the computed solution. That is why, instead of property 2.11, numericians make often use of the following characterization, coming from [42]:

Property 2.12

If $\mathbf{U}$ *is a numerical solution of system (2.25) with boundary condition* $\mathbf{V}_0(t)$*, then* $\mathbf{U}$ *is also a numerical solution of all the assimilated problem with boundary condition :*

$$\mathbf{V}'(t) \in \left\{ \mathbf{V} \in D; \mathbf{V} = \mathbf{V}_0 + \sum_{i=1}^{p} \alpha_i r_i, \ \alpha_i \in \mathbb{R} \right\}.$$

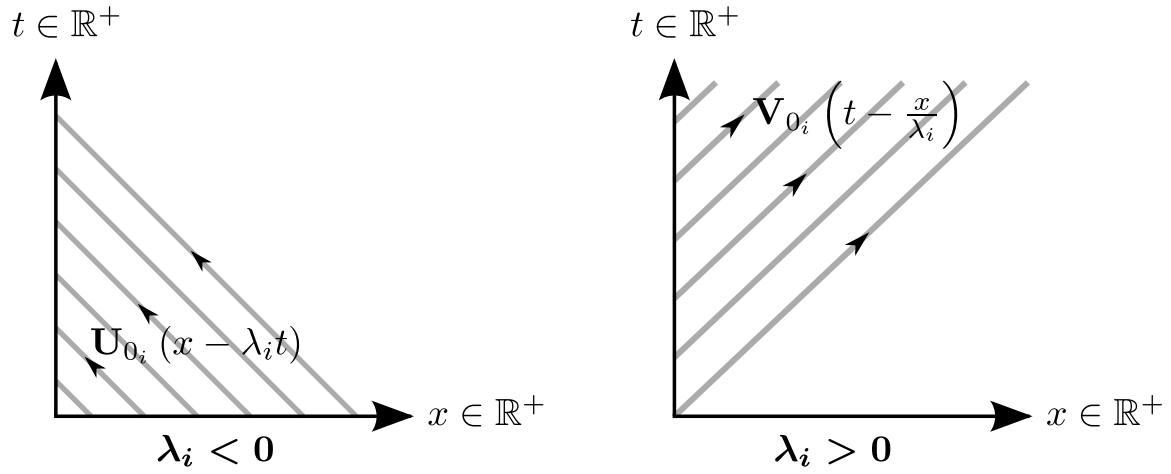


Figure 2.5: Effect of the boundary and initial conditions on the i^{th} component of the unknown in both cases when $\lambda_i < 0$ and $\lambda_i > 0$

We now come to a more complex problem, with space dimension n and non constant coefficient matrices. We are here dealing just with a formal generalization of the previous section. Some results are mathematically demonstrated, but we consider the physical explanation of the phenomenon as relevant enough. At almost any point of the boundary we have a tangent plane which is a hyperplane of $\mathbb{R}^n$. It is then well defined by its unit normal $\boldsymbol{\xi}$. We moreover suppose that $\boldsymbol{\xi}$ points inside the domain. If we further assume that our problem is symmetrizable, the Jacobian of the flux is diagonalizable in the direction of $\boldsymbol{\xi}$ and we once more call $\lambda_1, \ldots, \lambda_m$ its eigenvalues in the direction $\boldsymbol{\xi}$, sorted by increasing order and $r_1, \ldots, r_m$ the associated eigenvectors. If p is the integer index such that $\lambda_p < 0 \leqslant \lambda_{p+1}$, $r_1, \ldots, r_p$ are the direction of strictly outgoing information, $r_{p+1}, \ldots, r_m$ are the direction of entering information. We then see the boundary problem as a local one dimensional problem, and we assume that the problem is well-posed if the boundary condition enforces the solution on and only on the entering characteristic directions.

2.2 Euler and Navier-Stokes Equations

We will now describe physically the two systems of equations which solutions are going to be approximated during this thesis: the Euler and Navier-Stokes equations. We first start by the main mechanical conservation laws and apply some restrictions coming from fluid mechanics. Some finer hypothesis on the fluid behaviour will give the two systems of equations. Each term of these systems of partial derivatives will be described and analyzed. This will lead to some equivalent formulations that will be useful in the rest of the manuscript.

2.2.1 Lagrangian Coordinates

Let $\Omega_0 \subset \mathbb{R}^2$ be a set of particles of the plane at time $t = 0$, and $\Omega(t)$ its evolution at time t. For simplicity, we suppose that at any fixed time t, the function

$$f(t) : \left\{ \begin{array}{ccc} \Omega_0 & \longrightarrow & \Omega(t) \\ X & \longmapsto & x \end{array} \right.$$

is of class $\mathcal{C}^\infty$ and set up a diffeomorphism from Ω_0 into $\Omega(t)$. This allows us to define the Jacobian of the transformation, $J(X,t) = \det\left(\dfrac{\partial f(t)}{\partial X}\right)$, which is everywhere invertible. Because of its structure, a quick calculation [87] gives

$$\frac{\partial J}{\partial t}(X,t) = J(X,t)\ \mathrm{div}\left(\frac{\partial x}{\partial t}\right) = J(X,t)\ \vec{\nabla}.\vec{\mathbf{u}}.$$

The *Cartesian Coordinates* (x,t) are not very practical in the following construction, due to the fact that time derivatives must be calculated on the trajectories $x(t)$, depending on t. That is why it is really interesting to use the change of variable $f(t)$, leading to the *Lagrangian Coordinates* (X,t), where the spatial component $X = x(0)$ does not depend on time. Let now ω_0 be any subset of Ω_0 and $\omega(t)$ its image through $f(t)$. We are just going to apply basic physical conservation laws on the continuous medium $\omega(t)$, $t \in \mathbb{R}^+$.

2.2.2 Mass Conservation

Because by definition no particle enters or exits ω during the time, the global mass of ω is conserved:

$$\begin{aligned} \frac{\mathbf{D}m\left(\omega(t)\right)}{\mathbf{D}t} &= 0 \\ &= \frac{\mathbf{D}}{\mathbf{D}t}\left(\int_{\omega(t)} \rho(x,t)dx\right) \\ &= \int_{\omega_0} \frac{\mathbf{D}}{\mathbf{D}t}\left(\rho(f(X,t),t)J(X,t)\right)dX \\ &= \int_{\omega(t)} \left(\frac{\partial \rho}{\partial t}\right) + \mathrm{div}\left(\rho\vec{\mathbf{u}}\right)dx \end{aligned}$$

This being true for any ω, we obtain the *local mass conservation* equation

$$\frac{\partial \rho}{\partial t} + \mathrm{div}\left(\rho\vec{\mathbf{u}}\right) = 0, \quad \forall t > 0, \quad \forall x \in \Omega(t) \tag{2.26}$$

2.2.3 Momentum Conservation

Following the fundamental principle of dynamics, the variation of the total momentum in ω is given by

$$\frac{\mathbf{D}}{\mathbf{D}t}\left(\int_{\omega(t)} \rho\vec{\mathbf{u}}dx\right) = \int_{\omega(t)} \rho\vec{f}_v dx + \int_{\partial\omega(t)} \vec{F}_s(M,\vec{\mathbf{n}})ds \tag{2.27}$$

where $\vec{f_v}$ is the specific volumic force inside ω, and $\vec{F_s}(M, \vec{\mathbf{n}})$ is the surface force applied to the boundary of ω at point M and into the direction $\vec{\mathbf{n}}$, the outward normal to $\partial\omega$ at M.

A result of physics [23, 26, 53] shows that $\vec{F_s}$ must be a linear function of $\vec{\mathbf{n}}$. That means there exists a strain tensor $\sigma(M)$ such that

$$\forall M \in \Omega,\ \forall \vec{\mathbf{n}} \in \mathbb{R}^2,\ \vec{F_s}(M, \vec{\mathbf{n}}) = \sigma(M).\vec{\mathbf{n}}.$$

Therefore, using once more that the conservation relation above is verified for any subset ω_0 of Ω_0 and by applying the divergence theorem on the boundary term, we obtain the *local momentum conservation* equations, component by component ($i = 1, 2$)

$$\frac{\partial}{\partial t}(\rho u_i) + \operatorname{div}(\rho u_i \vec{\mathbf{u}}) = \rho(\vec{f_v})_i + \operatorname{div}(\sigma_i), \tag{2.28}$$

assuming σ_i is the i^{th} line of strain tensor σ.

2.2.4 Angular Momentum Conservation

Still following the fundamental principle of dynamics, the variation of the total angular momentum in ω is given by

$$\frac{\mathbf{D}}{\mathbf{D}t}\left(\int_{\omega(t)} \rho \vec{OM} \wedge \vec{\mathbf{u}}\, dM\right) = \int_{\omega(t)} \rho \vec{OM} \wedge \vec{f_v}\, dM + \int_{\partial\omega(t)} \vec{OM} \wedge (\sigma(M).\vec{\mathbf{n}})\, ds \tag{2.29}$$

In $\mathbb{R}^2$, this is a scalar equation on the direction Oz and using (2.26) and (2.28), we quickly find that $\sigma_{12} = \sigma_{21}$. In $\mathbb{R}^3$, we have 3 equations, each of them leading respectively to $\sigma_{32} = \sigma_{23}$, $\sigma_{13} = \sigma_{31}$ and $\sigma_{12} = \sigma_{21}$. In both two and three dimensional spaces, the angular momentum equation leads to the requirement that the strain tensor σ has to be symmetric.

2.2.5 Energy Conservation

The first principle of thermodynamics states that the variation of total energy with respect to time is equal to the power of all the forces applied to the system, plus the heat contributions. If we denote by $E = \frac{1}{2}\|u\|^2 + e$ the total energy per unit volume (e being the internal energy per unit volume), by w the specific heat creation by unit of time, and by $\vec{\mathbf{q}}$ the heat flux inside Ω, this is translated for any time t as

$$\begin{aligned}\frac{\mathbf{D}}{\mathbf{D}t}\left(\int_{\omega(t)} \rho E dx\right) &= \int_{\omega(t)} \rho \vec{f_v}.\vec{\mathbf{u}}\, dx + \int_{\partial\omega(t)} \vec{F_s}(M, \vec{\mathbf{n}}).\vec{\mathbf{u}}(M)\, ds \\ &\quad + \int_{\omega(t)} \rho w\, dx - \int_{\partial\omega(t)} \vec{\mathbf{q}}.\vec{\mathbf{n}}\, ds\end{aligned} \tag{2.30}$$

Once more using the divergence theorem if needed, and the fact ω is indifferently chosen, we obtain the local expression of the energy conservation equation

$$\frac{\partial \rho E}{\partial t} + \operatorname{div}(\rho E \vec{\mathbf{u}} - \sigma.\vec{\mathbf{u}} + \vec{\mathbf{q}}) = \rho \vec{f_v}.\vec{\mathbf{u}} + \rho w \tag{2.31}$$

2.2.6 Application to Fluids

Definition 2.13

A continuous medium is a Newtonian fluid when the strain tensor is a linear function of the stress tensor, defined by

$$(\mathbb{D})_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

We can then demonstrate [53, 26] there exists a variable p, called *pressure*, and two viscosity coefficients λ and μ called respectively *first and second Lamé coefficient of viscosity* such that

$$\sigma = (-p + \lambda \mathrm{div}\,(\vec{\mathbf{u}}))\mathbf{I} + 2\mu\mathbb{D} \tag{2.32}$$

Furthermore, these equations are just equations of conservation of the mass, the momentum, and the energy. They do not take into account the second principle of thermodynamics. We do have to find criteria in the system of equation and in the behaviour laws such that the compatibility with the second principle of thermodynamics is ensured. This second principle states there exists a scalar function s, called the *specific entropy*, such that for any ω

$$\frac{\mathbf{D}}{\mathbf{D}t}\left(\int_{\omega(t)} \rho s dx\right) \geqslant \int_{\omega(t)} \frac{\rho w}{T} dx - \int_{\partial\omega(t)} \frac{\vec{\mathbf{q}}.\vec{\mathbf{n}}}{T} ds. \tag{2.33}$$

We then obtain the *local entropy inequality*:

$$\frac{\partial \rho s}{\partial t} + \mathrm{div}\left(\rho s\vec{\mathbf{u}} + \frac{\vec{\mathbf{q}}}{T}\right) \geqslant \frac{\rho w}{T}. \tag{2.34}$$

Using the expression of the heat production coming from the *Energy conservation equation* (2.31)

$$\rho w = \rho\frac{\mathbf{D}e}{\mathbf{D}t} + \mathrm{div}\,(\vec{\mathbf{q}}) + \sigma : \mathbb{D},$$

where ':' denotes the operator $\sigma : \mathbb{D} = \sigma_{ij}\mathbb{D}_{ij}$, we obtain the well known *Clausius-Duhem Inequality* [53, 26]:

$$\rho\left(T\frac{\mathbf{D}s}{\mathbf{D}t} - \frac{\mathbf{D}e}{\mathbf{D}t}\right) - \frac{\vec{\mathbf{q}}.\overrightarrow{\nabla T}}{T} + \sigma : \mathbb{D} \geqslant 0. \tag{2.35}$$

This relation is essential in the study of the behaviour laws. For example, if we consider that the *internal energy* e only depends on the *specific entropy* s and on the *specific volume* $v = 1/\rho$, one has:

$$\begin{aligned}\rho\frac{\mathbf{D}e}{\mathbf{D}t} &= \rho\left(\frac{\partial e}{\partial s}\right)_\rho \frac{\mathbf{D}s}{\mathbf{D}t} + \left(\frac{\partial e}{\partial v}\right)_s \mathrm{div}\,(\vec{\mathbf{u}}) \\ &= \rho\left(\frac{\partial e}{\partial s}\right)_\rho \frac{\mathbf{D}s}{\mathbf{D}t} + \left(\frac{\partial e}{\partial v}\right)_s \mathbf{Tr}\,(\mathbb{D}),\end{aligned}$$

$\mathbf{Tr}\,()$ being the trace operator, and equation (2.35) is recast into

$$\rho\left(T - \left(\frac{\partial e}{\partial s}\right)_\rho\right)\frac{\mathbf{D}s}{\mathbf{D}t} - \left(p + \left(\frac{\partial e}{\partial v}\right)_s\right)\mathbf{Tr}\,(\mathbb{D}) + \lambda\left(\mathrm{div}(\vec{\mathbf{u}})\right)^2 + 2\mu\mathbb{D} : \mathbb{D} - \frac{\vec{\mathbf{q}}.\overrightarrow{\nabla T}}{T} \geqslant 0. \tag{2.36}$$

Let us consider the case of a constant velocity flow. The only possibility in order the *Clausius-Duhem Inequality* is always verified is ([53])

$$T = \left(\frac{\mathbf{D}e}{\mathbf{D}s}\right)_\rho \quad \text{and} \quad \frac{\vec{\mathbf{q}}.\overrightarrow{\nabla T}}{T} \leqslant 0.$$

If you consider the heat transfers follow the Fourier law $\vec{\mathbf{q}} = -k\overrightarrow{\nabla T}$, this implies in particular that the coefficient of heat conduction k has to be positive.

Moreover, if we consider now a flow at constant temperature, using the fact that $T = \left(\frac{\partial e}{\partial s}\right)_\rho$, *Clausius-Duhem Inequality* says

$$-\left(p + \left(\frac{\partial e}{\partial v}\right)_s\right)\mathbf{Tr}\left(\mathbb{D}\right) + \lambda\left(\mathrm{div}(\vec{\mathbf{u}})\right)^2 + 2\mu\mathbb{D}:\mathbb{D} - \frac{\vec{\mathbf{q}}.\overrightarrow{\nabla T}}{T} \geqslant 0,$$

which is always satisfied if and only if:

$$p = -\left(\frac{\partial e}{\partial v}\right)_s \quad \text{and} \quad \lambda\left(\mathrm{div}(\vec{\mathbf{u}})\right)^2 + 2\mu\mathbb{D}:\mathbb{D} - \frac{\vec{\mathbf{q}}.\overrightarrow{\nabla T}}{T} \geqslant 0$$

A quick calculation on the second term of the last equation [53] shows that this implies

$$3\lambda + 2\mu \geqslant 0. \tag{2.37}$$

Eventually, we can physically define entropy functions $(-S)$ which are concave [4, 60, 54, 69] and S is then also a convex mathematical entropy. That means, following the theorem of Mock, this system of equations is also symmetrizable and its symmetrizing matrix is the hessian $\nabla^2 S$. Then, all the properties of a symmetrizable system are valid here: propagation of the information at finite speed, *aso...*

2.2.7 Equation of State

We have built a system of PDE, with 4 equations and 5 unknowns (the conserved unknowns plus the pressure). In order to close the problem, we need an extra equation describing the nature of the fluid. This is an input that has to come from the physics. Indeed, the previous equations do not take into account the nature of the fluid we are dealing with (except for the viscosity coefficients). At this state of construction, we would apply the same set of equations to a balloon of helium as to a river of mercury, or to a cloud of vapor as to a large river. We need to find a relation between the physical variables describing the state of the fluid. These variables are usually the temperature, the pressure, the specific volume, the internal energy and the entropy. Starting from the equation of state of a physical system, it is possible to determine all the thermodynamic variables of the system and thus to express its properties.

Examples :

- **Ideal Gas:** the ideal gas law is known to be

$$pv = NRT \tag{2.38}$$

  where N is the number of mole of gas contained in the volume v and $R = 8.3144\,\mathrm{J.K^{-1}.mol^{-1}}$ is a universal constant.

- **Polytropic Gas:** a *polytropic gas* is merely an ideal gas for which the heat capacity at constant volume is constant. $c_v = \left.\frac{\partial e}{\partial T}\right|_v \Rightarrow e = c_v T$. Then relation (2.38) is reformulated into
$$p = (\gamma - 1)\rho e \tag{2.39}$$
where γ is the ratio of the heat capacities $\gamma = \frac{c_p}{c_v} (= 1.4$ for the air).

- **Other:** there exists many other equations of state, as Wan der Waals [119], hypersonic state [120], combustion [105, 37], mixed perfect gas [36], multiphase flow, dense gas [35], *etc.* But none of these have been used during this thesis. We just cite them here to show the numerous possibilities. When no further information is given, we are using the equation of state of polytropic gas.

2.2.8 Euler Equations

In this subsection, we consider the fluid as a *perfect fluid*. This is equivalent to the following three hypothesis:

1. The fluid is non-viscous : $\lambda = \mu = 0 \Rightarrow \varpi = -p\mathbf{I}$,
2. There is no body forces : $\vec{f_v} = \vec{0}$,
3. There is no heat transfer : $w = 0$, $\vec{\mathbf{q}} = \vec{0}$.

Gathering equations (2.26),(2.28) and (2.31), we obtain the very well known Euler system :

$$\left\{ \begin{array}{lcll} \dfrac{\partial \rho}{\partial t} & + & \operatorname{div}(\rho \vec{\mathbf{u}}) & = 0 \\ \dfrac{\partial \rho u_i}{\partial t} & + & \operatorname{div}(\rho u_i \vec{\mathbf{u}} + p\boldsymbol{\delta}_i) & = 0, \quad i = 1,2 \\ \dfrac{\partial \rho E}{\partial t} & + & \operatorname{div}((\rho E + p)\vec{\mathbf{u}}) & = 0 \end{array} \right. \tag{2.40}$$

where $\boldsymbol{\delta}_i$ is the i-th column of the 2×2 identity matrix.

Concerning the equation of state, we will always use *the incomplete equation of state* of polytropic gas (2.39). It is called incomplete because it is not a relation between all the state variables, but a simple pressure law. It is nevertheless a sufficient law for the closure of the Euler equations.

If we set

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho \vec{\mathbf{u}} \\ \rho E \end{pmatrix}, \quad \text{and} \quad \vec{\boldsymbol{\mathcal{F}}} = (\mathbf{F}_1, \mathbf{F}_2), \quad \text{with} \quad \mathbf{F}_i = \begin{pmatrix} \rho u_i \\ \rho u_i \vec{\mathbf{u}} + p\boldsymbol{\delta}_i \\ (\rho E + p) u_i \end{pmatrix} \tag{2.41}$$

system (2.40) is rewritten in the compact form

$$\frac{\partial \mathbf{U}}{\partial t} + \operatorname{div}\left(\vec{\boldsymbol{\mathcal{F}}}(\mathbf{U})\right) = 0$$

and if we denote by $\mathrm{A} = \dfrac{\partial \mathbf{F}_1}{\partial \mathbf{U}}$, $\mathrm{B} = \dfrac{\partial \mathbf{F}_2}{\partial \mathbf{U}}$ and $\vec{\boldsymbol{\lambda}} = (\mathrm{A}, \mathrm{B})$ the Jacobian of the fluxes, we obtain for a smooth enough solution the equivalent quasi-linear form

$$\frac{\partial \mathbf{U}}{\partial t} + \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla \mathbf{U}} = 0.$$

2.2.9 Properties of the Euler Equations

The matrices A and B are the following

$$\mathrm{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ (\gamma-1)\mathscr{E}_c - u^2 & (3-\gamma)u & (1-\gamma)v & (\gamma-1) \\ -uv & v & u & 0 \\ u\left((\gamma-1)\mathscr{E}_c - \mathcal{H}\right) & \mathcal{H} + (1-\gamma)u^2 & (1-\gamma)uv & \gamma u \end{pmatrix}$$

$$\mathrm{B} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ -uv & v & u & 0 \\ (\gamma-1)\mathscr{E}_c - v^2 & (1-\gamma)u & (3-\gamma)v & (\gamma-1) \\ v\left((\gamma-1)\mathscr{E}_c - \mathcal{H}\right) & (1-\gamma)uv & \mathcal{H} + (1-\gamma)v^2 & \gamma v \end{pmatrix}$$

where $\mathscr{E}_c = (u^2+v^2)/2$ and $\mathcal{H} = e + p/\rho$ denote the *kinetic energy* and the *enthalpy* per unit volume, respectively. Given a unit normal $\vec{\mathbf{n}} = (n_x, n_y) \in \mathrm{S}^1$, the matrix

$$\vec{\boldsymbol{\lambda}}.\vec{\mathbf{n}} = \begin{pmatrix} 0 & n_x & n_y & 0 \\ (\gamma-1)\mathscr{E}_c n_x - u\vec{\mathbf{u}}.\vec{\mathbf{n}} & \vec{\mathbf{u}}.\vec{\mathbf{n}} + (2-\gamma)un_x & un_y + (1-\gamma)vn_x & (\gamma-1)n_x \\ (\gamma-1)\mathscr{E}_c n_y - v\vec{\mathbf{u}}.\vec{\mathbf{n}} & vn_x + (1-\gamma)un_y & \vec{\mathbf{u}}.\vec{\mathbf{n}} + (2-\gamma)vn_y & (\gamma-1)n_y \\ \vec{\mathbf{u}}.\vec{\mathbf{n}}\left((\gamma-1)\mathscr{E}_c - \mathcal{H}\right) & \mathcal{H}n_x + (1-\gamma)u\vec{\mathbf{u}}.\vec{\mathbf{n}} & \mathcal{H}n_y + (1-\gamma)v\vec{\mathbf{u}}.\vec{\mathbf{n}} & \gamma\vec{\mathbf{u}}.\vec{\mathbf{n}} \end{pmatrix}$$

is diagonalizable and one has $\vec{\boldsymbol{\lambda}}.\vec{\mathbf{n}} = \mathcal{R}\boldsymbol{\Lambda}\mathcal{L}$ with:

$$\boldsymbol{\Lambda} = \begin{pmatrix} \vec{\mathbf{u}}.\vec{\mathbf{n}} - c & 0 & 0 & 0 \\ 0 & \vec{\mathbf{u}}.\vec{\mathbf{n}} & 0 & 0 \\ 0 & 0 & \vec{\mathbf{u}}.\vec{\mathbf{n}} & 0 \\ 0 & 0 & 0 & \vec{\mathbf{u}}.\vec{\mathbf{n}} + c \end{pmatrix},$$

$$\mathcal{R} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ u - cn_x & u & -n_y & u + cn_x \\ v - cn_y & v & n_x & v + cn_y \\ \mathcal{H} - \vec{\mathbf{u}}.\vec{\mathbf{n}}c & \mathscr{E}_c & \vec{\mathbf{u}}.\vec{\mathbf{n}}^{\perp} & \mathcal{H} + \vec{\mathbf{u}}.\vec{\mathbf{n}}c \end{pmatrix},$$

$$\mathcal{L} = \begin{pmatrix} \frac{1}{2c}\left(\frac{\gamma-1}{c}\mathscr{E}_c + \vec{\mathbf{u}}.\vec{\mathbf{n}}\right) & -\frac{1}{2c}\left(\frac{\gamma-1}{c}u + n_x\right) & -\frac{1}{2c}\left(\frac{\gamma-1}{c}v + n_y\right) & \frac{\gamma-1}{2c^2} \\ 1 - \frac{(\gamma-1)\mathscr{E}_c}{c^2} & \frac{(\gamma-1)u}{c^2} & \frac{(\gamma-1)v}{c^2} & \frac{(1-\gamma)}{c^2} \\ -\vec{\mathbf{u}}.\vec{\mathbf{n}}^{\perp} & -n_y & n_x & 0 \\ \frac{1}{2c}\left(\frac{\gamma-1}{c}\mathscr{E}_c - \vec{\mathbf{u}}.\vec{\mathbf{n}}\right) & -\frac{1}{2c}\left(\frac{\gamma-1}{c}u - n_x\right) & -\frac{1}{2c}\left(\frac{\gamma-1}{c}v - n_y\right) & \frac{\gamma-1}{2c^2} \end{pmatrix}.$$

We have introduced a new variable $c = \sqrt{\frac{\gamma p}{\rho}}$ which represents the speed of propagation of the acoustic phenomena. It is well known that for air $c \approx 330\mathrm{m.s}^{-1}$ at standard temperature. The last decomposition of the Jacobian matrices shows that the Euler equations are a system of conservation laws which is *constantly hyperbolic*.

2.2.10 Navier-Stokes Equations

We now come to the complete Navier-Stokes equations. We say "complete" because the Navier-Stokes equations are today considered as one of the physical system that best models some strange phenomena observed in the reality. Even if one would add new equations and new variables in order for instance to reproduce numerically some turbulence phenomena, they are in fact already described in the set of the Navier-Stokes equations. Turbulence equations and variables are just an artifact aiming to overcome the lack of accuracy of the nowadays numerical schemes, relatively to the space scale of the turbulent phenomena. Most of the instabilities, turbulence, etc... making fluid mechanics such an appealing subject are solution of this PDS.

As we did for the Euler equations, we first start by some hypothesis on the fluid:

1. The fluid is a *Newtonian fluid*:
$$\varpi = \left(-p + \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right)\mathbf{I} + \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right),$$
see Definition 2.13,
2. According to Fourier law, the heat diffusion is opposite to the gradient of temperature. The coefficient of proportionality $k > 0$ is the *coefficient of heat diffusion*: $\vec{\mathbf{q}} = -k\vec{\nabla}T$,
3. There is no body forces : $\vec{f_v} = \vec{0}$,
4. There is no heat production inside the domain : $w = 0$,
5. The fluid is a polytropic gas : $p = (\gamma - 1)\rho e$. This condition being just a *pressure law*, it can be easily replaced by another complete *Equation of State*. This one is used for its simplicity.
6. By *Clausius-Duhem Inequality*, we must have $3\lambda + 2\mu \geqslant 0$ and we respect this constraint by enforcing the viscous coefficient closure:
$$\lambda = -\frac{2}{3}\mu$$

If we gather these hypothesis with equations of conservation (2.26),(2.28) and (2.31), we obtain

$$\left\{\begin{array}{lcll} \dfrac{\partial \rho}{\partial t} & + & \operatorname{div}(\rho\vec{\mathbf{u}}) & = 0 \\ \dfrac{\partial \rho u_i}{\partial t} & + & \operatorname{div}(\rho u_i\vec{\mathbf{u}} + p\boldsymbol{\delta}_i) & = (\lambda+\mu)\dfrac{\partial}{\partial x_i}\Big(\operatorname{div}\vec{\mathbf{u}}\Big) + \mu\Delta u_i, \quad i = 1,2 \\ \dfrac{\partial \rho E}{\partial t} & + & \operatorname{div}((\rho E + p)\vec{\mathbf{u}}) & = \operatorname{div}\left(k\overrightarrow{\nabla T} + \mathbb{T}\vec{\mathbf{u}}\right). \end{array}\right. \tag{2.42}$$

This is the form in which Navier-Stokes equations are usually presented. In order to simplify, we have used the viscous tensor

$$\mathbb{T} = 2\mu\mathbb{D} + \lambda\operatorname{div}(\vec{\mathbf{u}})\,\mathbf{I} = \begin{pmatrix} \lambda\operatorname{div}(\vec{\mathbf{u}}) + 2\mu\dfrac{\partial u}{\partial x} & \mu(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}) \\ \mu(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}) & \lambda\operatorname{div}(\vec{\mathbf{u}}) + 2\mu\dfrac{\partial v}{\partial y} \end{pmatrix}$$

They are many different ways of writing these equations, above all depending on the application in mind.

One formulation will be however particularly useful in Chapter 8. It is a bit more complex than this one, but it has the advantage to present the system in a complete matricial form. It has been inspired by Chapter 2 of P.J. Capon's Thesis [27]. If we consider the advective flux defined in (2.41) and the following diffusive matrices

$$K_{11} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -\frac{4}{3}u & \frac{4}{3} & 0 & 0 \\ -v & 0 & 1 & 0 \\ -\left(2\mathscr{E}_c + \frac{u^2}{3} + \frac{\gamma}{\mathrm{Pr}}(e-\mathscr{E}_c)\right) & u\left(\frac{4}{3}-\frac{\gamma}{\mathrm{Pr}}\right) & v\left(1-\frac{\gamma}{\mathrm{Pr}}\right) & \frac{\gamma}{\mathrm{Pr}} \end{pmatrix},$$

$$K_{12} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{2}{3}v & 0 & -\frac{2}{3} & 0 \\ -u & 1 & 0 & 0 \\ -\frac{uv}{3} & v & -\frac{2}{3}u & 0 \end{pmatrix},\ K_{21} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -v & 0 & 1 & 0 \\ \frac{2}{3}u & -\frac{2}{3} & 0 & 0 \\ -\frac{uv}{3} & -\frac{2}{3}v & u & 0 \end{pmatrix},$$

and

$$K_{22} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -u & 1 & 0 & 0 \\ -\frac{4}{3}v & 0 & \frac{4}{3} & 0 \\ -\left(2\mathscr{E}_c + \frac{v^2}{3} + \frac{\gamma}{\mathrm{Pr}}(e-\mathscr{E}_c)\right) & u\left(1-\frac{\gamma}{\mathrm{Pr}}\right) & v\left(\frac{4}{3}-\frac{\gamma}{\mathrm{Pr}}\right) & \frac{\gamma}{\mathrm{Pr}} \end{pmatrix},$$

we can rewrite system (2.42) as

$$\mathbf{U}_{,t} + \mathrm{div}\left(\vec{\boldsymbol{\mathcal{F}}}(\mathbf{U})\right) = \left(K_{ij}\mathbf{U}_{,j}\right)_{,i} = \mathrm{div}\left(\mathbb{K}.\overrightarrow{\nabla\mathbf{U}}\right) \tag{2.43}$$

where we have used the Einstein notation and "$_{,j}$" refers to the derivative with respect to the j^{th} space variable.

2.2.11 Boundary Conditions

We finish this chapter with the boundary conditions that are going along with these two models: the Euler and Navier-Stokes equations. These conditions are needed to close the problem. It is rather hard to enumerate all the boundary conditions that have been developed for some specific purposes. We are here just going to list the boundary conditions we have been using during this thesis. We describe them here in their continuous versions. The way they are discretized is shown in Section 5.4.

- **Inflow and Outflow**: it is sometimes useful to impose a given state at an entrance or an output of a domain. This is for example the case when the domain is linking two tanks of pressure at two different states. We are also going to use this at the external boundaries of a domain containing an aircraft. The goal is to simulate the flow around the aircraft at a certain speed. The easiest way to do this is to consider the problem in the referential of

the aircraft: the domain is fixed and the air moves at the opposite velocity of the aircraft. The external boundary are considered as at infinity and we wish to impose there a *Far-field State*. In both cases, if $\mathbf{U}_\infty$ is the state we want to impose on boundary Γ_∞, one has:

$$\mathbf{U}(\mathbf{x}) = \mathbf{U}_\infty, \quad \forall \mathbf{x} \in \Gamma_\infty. \tag{2.44}$$

In practice, if $\vec{\mathbf{n}}$ is the inward normal to Γ_∞, some characteristics in the direction $\vec{\mathbf{n}}$ are often leaving the domain. Then, as noticed in Property 2.11, we do not have to impose anything on these characteristics, and the condition is usually recast into

$$\mathbf{U}(\mathbf{x}) = \mathcal{A}(\vec{\mathbf{n}})^+ \mathbf{U}_\infty, \quad \forall \mathbf{x} \in \Gamma_\infty.$$

where $\mathcal{A}(\vec{\mathbf{n}})^+$ denotes the positive part of the Jacobian operator in the direction $\vec{\mathbf{n}}$.

- **No-Slip Wall**: when the fluid is considered viscous, it sticks to the walls. By continuity, the velocity $\vec{\mathbf{u}}$ of the flow along the wall must be the same as the velocity of the wall $\vec{\mathbf{u}}_{\text{wall}}$

$$\vec{\mathbf{u}}(\mathbf{x}) = \vec{\mathbf{u}}_{\text{wall}}, \quad \forall \mathbf{x} \in \Gamma_{\text{wall}}. \tag{2.45}$$

In most of cases, the wall is still and $\vec{\mathbf{u}}_{\text{wall}} = \vec{0}$. Then, following the eigenvalues of the advection matrix given in Subsection 2.2.9, one has only one outgoing characteristic. The system having size $m = d + 2$, one needs an extra boundary condition. This is provided by the heat transfer between the wall and the fluid. This can be done in two ways. The temperature can either be considered continuous. In this case, we just impose the temperature of the wall T_{wall}

$$T(\mathbf{x}) = T_{\text{wall}}, \quad \forall \mathbf{x} \in \Gamma_{\text{wall}}. \tag{2.46}$$

Or, in the case of a steady simulation, one consider that the heat transfers are null at steady state. The heat flow between the wall and the fluid has to be zero and the boundary condition reads:

$$\frac{\partial T}{\partial \vec{\mathbf{n}}}(\mathbf{x}) = 0, \quad \forall \mathbf{x} \in \Gamma_{\text{wall}}. \tag{2.47}$$

- **Slip Wall**: finally, in the case of the Euler equations, the fluid is considered as non viscous, and it is completely possible that the fluid slips on the walls. But on the other hand, it is still impossible that the fluid enters the boundary (by definition of the wall). Then the *no-slip* condition of the viscous flows is formulated as

$$\vec{\mathbf{u}}(\mathbf{x}).\vec{\mathbf{n}} = 0, \quad \forall \mathbf{x} \in \Gamma_{\text{wall}}. \tag{2.48}$$

Chapter 3

High Order Schemes

This chapter is devoted to a brief introduction to high order numerical schemes. The main goal is to explain why high order schemes are today so attractive for CFD, but also what their main drawbacks are. It is the occasion to present roughly the concept of higher order schemes and to set down conventions and notations on mesh parameters and data representation. In a first part, we are going to introduce a general framework for numerical schemes and explain what a high order scheme is. We also introduce the main definitions on mesh and geometry. In a second part, we describe the polynomial representation of the data on triangles and quadrangles. A last section eventually treats the appealing features of high order schemes.

3.1 Numerical Schemes: a General Framework

In this section, we are about to present the numerical resolution of a PDS in a very abstract way. We see that the solution of a problem in a functional space with infinite dimension can be approximated by the solution of an associated problem, this time existing in a finite dimensional functional space. At the end, we have just projected the sought solution on a restricted finite dimensional space of unknowns, without even knowing this exact solution. All numerical schemes are included in this general framework.

In the following, we call $\mathscr{E}$ a functional space with infinite dimension and $\Diamond$ a differential operator on $\mathscr{E}$. We also denote by $\mathscr{L}$ a Hilbert functional space such that :

i) $\mathscr{E} \subset \mathscr{L}$;

ii) $\Diamond : \mathscr{E} \to \mathscr{L}$.

3.1.1 Finite Dimension Approximation

We want to solve the following problem:

$$\text{Find } u \in \mathscr{E}, \text{ such that } \begin{cases} \Diamond u = f, & \mathbf{x} \in \Omega \\ u = g, & \mathbf{x} \in \Gamma \subset \partial\Omega. \end{cases} \tag{3.1}$$

f and g are of course regular enough functions, in order our problem is well-defined. This is a very general problem, and most of the modelizations in physics lead to such a problem [87]. The difficulty is that we are today usually absolutely not able to find an exact solution of such a PDS, even in some apparently very simple cases. We have to approximate the solution and this is done numerically. We first remark that if u^* is a solution of problem (3.1), then $\forall v \in \mathscr{L}$, $\langle \Diamond u^*, v\rangle_{\mathscr{L}} = \langle f, v\rangle_{\mathscr{L}}$. We now denote by $\mathscr{W}_h$ a subspace of $\mathscr{E}$ with finite dimension n, and by $w_1^h, \ldots, w_n^h$ a basis of $\mathscr{W}_h$. The subscript "$_h$" is used in order to keep in mind that $\mathscr{W}_h$ depends on the geometry of Ω, and on a spatial discretization of Ω, $\mathcal{M}_h$, that will be called further the *mesh*. h represents a characteristic length associated to the mesh. The finite dimensional subset also depends on the order of representation of the data on the discretized space and on other geometrical parameters. We now define $\mathcal{P}_h$ as a projection from $\mathscr{E}$ to $\mathscr{W}_h$, for example

$$\mathcal{P}_h : \left\{ \begin{array}{ccc} \mathscr{E} & \longrightarrow & \mathscr{W}_h \\ u & \longmapsto & \displaystyle\sum_{i=1}^{n} \left\langle u, w_i^h \right\rangle_{\mathscr{L}} w_i^h \end{array} \right.$$

We will see next this is not the only way of defining a projection from $\mathscr{E}$ to $\mathscr{W}_h$, and we are for that matter usually not going to use this one. The reader has to consider this projection just as a theoretical example.

We can then associate (3.1) to a finite dimensional problem

$$\text{Find } u_h \in \mathscr{W}_h, \text{ such that } \left\{ \begin{array}{ll} \langle \Diamond u_h, v_h\rangle_{\mathscr{L}} = \langle f, v_h\rangle_{\mathscr{L}}, & \forall v_h \in \mathscr{W}_h \\ u_h = \mathcal{P}_h(g), & \forall \mathbf{x} \in \Gamma \end{array} \right. \tag{3.2}$$

If $\Diamond$ is a linear operator, this problem can be obviously put into the matricial form $A.U = B$ where $A_{ij} = \left\langle \Diamond w_i^h, w_j^h \right\rangle_{\mathscr{L}}$ and $B_i = \langle f, w_i^h\rangle_{\mathscr{L}} + \mathcal{F}_i^{bc}$. $\mathcal{F}_i^{bc}$ stands here for the contribution of some numerical fluxes on the boundary Γ, this ensuring the boundary condition $u_h = \mathcal{P}_h(g)$.

In this case, problem (3.2) is well-posed if matrix A is invertible and admits then a unique solution $u_h \in \mathscr{W}_h$. u_h is then called *the approximated solution*. We are going to see in the next section how the quality of the approximation of u^* by u_h is quantified: the order of accuracy of the scheme.

3.1.2 Error and Truncation Error

u^* and u_h are both functions of $\mathscr{L}$ and we can then write the global error of approximation as

$$\|u^* - u_h\|_{\mathscr{L}} \leqslant \underbrace{\|u^* - \mathcal{P}_h(u^*)\|_{\mathscr{L}}}_{I} + \underbrace{\|\mathcal{P}_h(u^*) - u_h\|_{\mathscr{L}}}_{II}.$$

The two terms of the right-hand side represent different things.

Term I: it is the *projection error*. It depends on the polynomial order of approximation of the data. Generally, if $\mathscr{W}_h$ is spanned by polynomials of order k and u^* is regular enough, the order of magnitude of term I is dominated by h^{k+1}, where h is a characteristic length of the discretization of Ω needed to define $\mathscr{W}_h$. That means in particular that $\mathcal{P}_h(u)$ converges toward u as h goes to 0 for any regular enough $u \in \mathscr{E}$, and that in a certain sense, $\mathscr{W}_h$ converges toward $\mathscr{E}$ as h gets smaller.

Term *II*: it is called the *truncation error* of the scheme. As one can see, if the truncation error is also of order $k+1$, then u_h is an approximation of order $k+1$ in $\mathscr{L}$-norm of the exact solution u^*. Thus, below we will speak of a $(k+1)^{\text{th}}$-order scheme when referring to a scheme using a k^{th} order representation of the data and which truncation error is of order $(k+1)$. As we have already seen in the introduction, there exists several different types of high order schemes. The main differences between these formulations come from the functional space approximation.

We have now presented the main concepts of the numerical resolution of a complex problem a very abstract way. The important thing here is to understand that a numerical resolution of a problem in an infinite functional space is done by defining a certain projection of the solution on a finite dimensional subspace. The projection of the exact solution is the unique solution of a finite dimensional problem which can be "easily" solved. The nature of the projection is defined by the type of the chosen numerical scheme. This will be explained later on. What one can expect is that the finer the approximation of $\mathscr{E}$ by $\mathscr{W}_h$ is, the closer to u^* u_h is. This is always the result of theorems we call *"Lax-Wendroff like"* and that are essential in the development of the numerical schemes.

Eventually, the finite dimensional subspace $\mathscr{W}_h$ is in fact completely defined by the discretization of the domain and the order of representation of the data inside the discrete meshing. This is the subject of the next sections.

3.1.3 Domain Discretization

In the last paragraphs, we have implicitly considered Ω as our spatial domain. To simplify the presentation, we suppose Ω is bi-dimensional. The illustrations will be much easier.

Let $\Omega \subset \mathbb{R}^2$ be the continuous spatial domain. A spatial approximation of Ω is a finite set $\mathcal{T}_h$ of non overlapping elements with strictly positive area such that $\bigcup_{\mathrm{T}\in\mathcal{T}_h} \mathrm{T} = \Omega$ or at least such that the area belonging to $\bigcup_{\mathrm{T}\in\mathcal{T}_h} \mathrm{T}$ or to Ω but not to both, tends toward zero when the refinement parameter h is getting smaller. Here, h represents a characteristic distance between two vertices of the mesh. In our case, it will be either the constant mesh spacing on the boundary of Ω or the maximal distance between two vertices or the square root of the area of the biggest element in $\mathcal{T}_h$. We also call $\mathcal{M}_h$ the set of the vertices of the elements of $\mathcal{T}_h$, but by abuse of notation, $\mathcal{M}_h$ also represents the set of any kind of entity of the mesh. It contains the vertices of the mesh as well as the edges, the faces or the elements, etc...

There are many types of meshes and there is a wide vocabulary on this subject. We give hereafter the main nomenclature used here. Even if the elements of $\mathcal{T}_h$ are denoted by T they must not always be triangles. They can be triangles or quadrangles or any type of polyhedral or even isoparametric elements as shown on figure 3.1, and this will be true for the rest of this manuscript. We are not going to speak here about isoparametric elements as a whole section is devoted to them, see page 137. The construction of such an element is detailed in this section. When the mesh is composed only by triangles, it is called a *triangulation*. In order to eliminate too "flat" triangles, we assume that the mesh is regular enough and that there exist two constants C_1 and C_2 such that the ratio of two heights of any triangle of the mesh stands between C_1 and

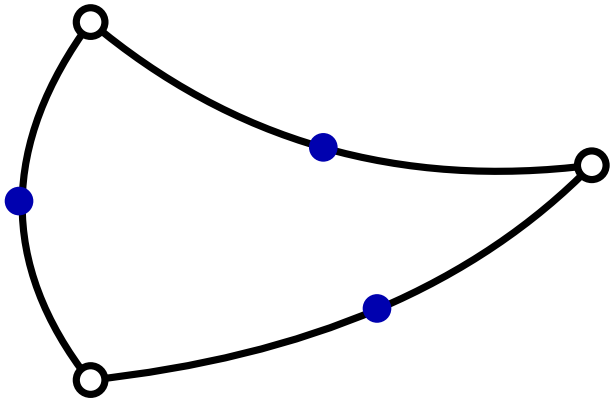


Figure 3.1: **(Isoparametric Elements)** The edges of these elements are represented by the same polynomial order k as the one used inside the element to approximate the solution. In that case, $k = 2$ and the edges are quadratic, uniquely defined by the vertices and the middles of the edges. These elements are very useful to represent the boundaries with a much better accuracy.

C_2.

$$\exists C_1, C_2 \in \mathbb{R}^+, \text{ such that } \forall \mathrm{T} \in \mathcal{M}_h, \forall h_1, h_2 \text{ heights of T},$$
$$C_1 \leqslant \frac{h_1}{h_2} \leqslant C_2 \tag{3.3}$$

The same argument is suitable for quadrangles with the ratio of the diagonal lengths. There exists two main types of triangulations: the *structured* and the *unstructured* ones, see Figure 3.2. The main difference is the number of direct neighbours of each vertex (the vertices of $\mathcal{M}_h$ sharing an edge with it). In the case of a structured triangulation, the mesh is really regular, all the elements are identical or quasi-identical, and the number of direct neighbours stays constant. Whereas in the unstructured case, this number of direct neighbours is not necessarily constant and it is generally not. When a mesh mixes different types of elements it is called a *hybrid mesh*. Hybrid meshes are very interesting from a geometrical point of view. As we have seen a meshing does not have to match the domain perfectly but must approach it with the area of the difference depending on h. As one can guess it is now much easier to match some complex geometries as an obtuse angle or round nose with a hybrid unstructured mesh than with a structured triangulation.

In this thesis, we are also dealing only with *conformal meshes*. A mesh is conformal, when no vertex of an element lies inside an edge of another element. This is represented on Figure 3.3. Residual Distribution Scheme on non conformal meshes is actually a rather complex development even if it is not declared as impossible. The main problems are how to define the direct neighbours of the non conformal vertices as well as its dual cells (see next paragraph for definition). It is then quite complex to associate a basis function to those vertices. This is not the aim of this manuscript and that is why all the meshes are thereafter conformal.

For any type of meshing, the following notations are useful. For any element T of $\mathcal{T}_h$, we denote by $|\mathrm{T}|$ its area. For any vertex $i \in \mathcal{M}_h$, $\mathcal{D}_i$ is the subset of elements containing i. $|\mathcal{D}_i|$ is the sum of the areas of the elements of $\mathcal{D}_i$. By abuse of notation, $\mathcal{D}_i$ also denotes the direct neighbours of i, *ie.* the nodes of the elements members of $\mathcal{D}_i$. To any node i of the mesh, we

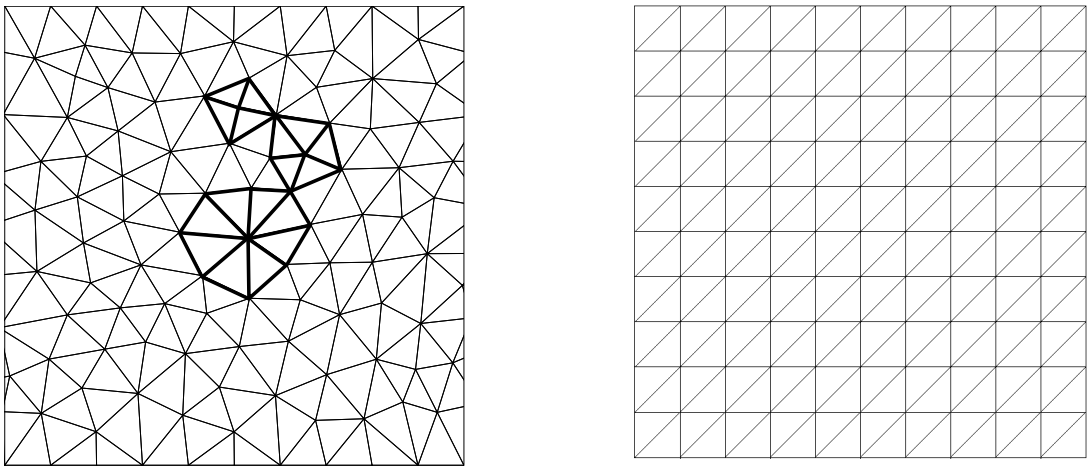


Figure 3.2: Unstructured (left) and structured (right) triangulation

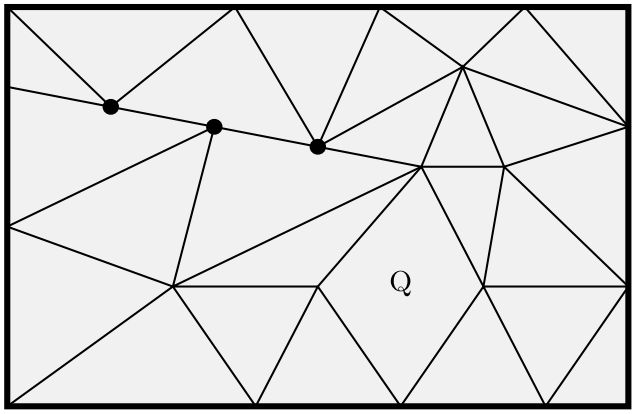


Figure 3.3: **(Non Conformal Mesh)** The 3 black points denote non conformal points, because they lie inside the edge of another element. Q denotes the only quadrangle of this mesh.

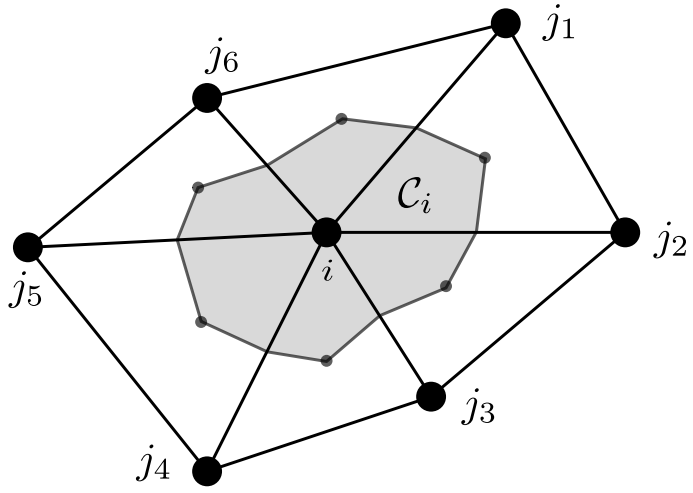


Figure 3.4: **(Dual Cell)** On this figure is represented node i, $\mathcal{D}_i$ the subset of elements sharing i, its direct neighbours $j_1, \ldots, j_6$ and the associated dual cell $\mathcal{C}_i$. $\mathcal{C}_i$ is defined by joining the midpoints of the edges sharing i and the centroids of the triangles of $\mathcal{D}_i$. This can be generalized to any polyhedral.

associate its *dual cell*, $\mathcal{C}_i$, represented on figure 3.4. It represents the domain of influence of the scheme for node i. It is obtained by joining the gravity centers of the elements of $\mathcal{D}_i$ with the midpoints of the edges meeting at i. This notion is very important in the case of *Finite Volume Schemes* ($\mathcal{FV}$), see Subsection 4.1.3 page 62. In the case of $\mathcal{RD}$ schemes, we are mainly interested by the dual cell area

$$|\mathcal{C}_i| = \frac{|\mathcal{D}_i|}{3},$$

especially for linear representation of the data.

Euler Formula We are here giving a formula linking the number of elements, faces, edges and vertices in a 2D mesh. It is called the *Euler Formula* and it has been conjectured in 1752. This formula has actually a much wider generalization though and can be applied on any kind of really weird topology [71, 112]. This is not the object of this work and we restrict our demonstration to two dimensional unstructured hybrid meshes. The main argument of this demonstration can be applied as it is to the three dimensional case.

Property 3.1

Let $\mathcal{M}_h$ be a unstructured hybrid meshing of a two dimensional simply connected domain Ω and F, E, V being respectively, the number of elements, edges and vertices in $\mathcal{M}_h$. Then

$$F - E + V = 1 \tag{3.4}$$

Remark 3.2 (*Euler Characteristic*)

The quantity $\chi = F - E + V$ is called the Euler Characteristic. It is defined in any polyhedral meshing, in any dimension, as the alternate sum $\chi = k_0 - k_1 + k_2 - k_3 + \ldots$, where k_n denotes

the number of cells of dimension n in the mesh. It is a constant, depending on the topology in which Ω is drawn (the number of connected components, the number of holes in Ω, etc...). In two or three dimensions, when Ω is simply connected, we always have $\chi = 1$.

For example in a tetrahedron, we have 1 tetrahedron, 4 faces, 6 edges and 4 vertices : $V - E + F - T = 1$. In a cube, we have 1 cube, 6 faces, 12 edges and 8 vertices: $\chi = 1$.

We wrote property 3.1 that way because it is the way it will be used later. But the demonstration below is in fact valid in a much more general framework, that is why we put here this lemma.

Lemma 3.3

Let $\mathscr{N}$ be a cloud of points of $\mathbb{R}^2$, $\mathscr{E}$ a set of edges that links some vertices two by two and F the number of polygons formed by these edges. We denote by C the number of connected components in $\mathscr{N}$ (two vertices are part of a same connected component when there exists a path of edges linking them both). Then

$$\#\mathscr{N} - \#\mathscr{E} + F = C \tag{3.5}$$

Proof: This proof is illustrated by Figure 3.5. Let us now remove one edge. There are two possibilities:

1. The removal increases the number of connected components C by one (e.g. edge marked with '$(*)$' in Figure 3.5). The edge is then "single", which means it is not part of an element, and the number of elements stays constant. As the number of vertices stays also constant, we have added one both to the right and left hand side. Formula (3.5) is conserved through this transformation.
2. C does not change as a result of the edge removal (e.g. edge marked with '$(\%)$' in Figure 3.5). Then there exists a path different of the removed edge that links the both end points of this edge. The removed edge was then part of an element, and the edge removal has destroyed this element. The number of both edges and faces decreases by one while the number of vertices still stays fixed. Thus, both sides of Formula (3.5) do not change.

This means that Euler formula holds for a meshing if and only if it holds for a meshing with one edge removed. By induction, it holds for a meshing if and only if it holds for the cloud of points with all edges removed. But after all edges are removed, what we are left with is only $\#\mathscr{N} = n$ *separated vertices.* Thus, $C = n$, $\#\mathscr{E} = 0$, $F = 0$ and the formula (3.5) is obviously satisfied. ■

3.2 Polynomial Representation of the Data

Now that we have defined what a mesh is, we can go further and associate a basis function to each vertex of the mesh. In the beginning of this chapter, $\mathscr{W}_h$ has been defined abstractly as a finite dimensional subset of the whole functional space $\mathscr{E}$. In fact, $\mathscr{W}_h$ is spanned by the basis functions associated to the degrees of freedom of the mesh $\mathcal{M}_h$. As these basis functions form a linearly independent subset of $\mathscr{W}_h$, it is a basis of $\mathscr{W}_h$. One can note that we have used the word *degree of freedom* (DoF) instead of *vertex*. These notions are the same when using a linear representation of the data. But we aim to develop a polynomial representation of the data with

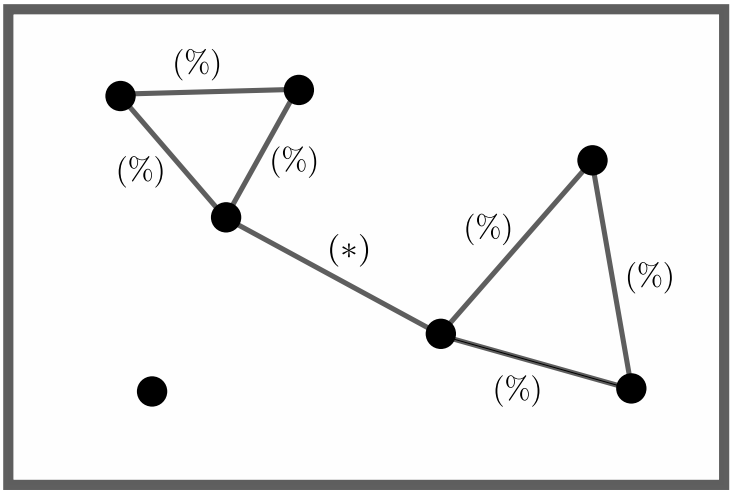


Figure 3.5: 7 points, 7 edges, 2 triangles and 2 connected components: $7 - 7 + 2 = 2$.

any choosen order k. To do so, we have to add new degrees of freedom inside each element, in order to defined what we are going to call $\mathbb{P}^k$ basis functions on these elements.

3.2.1 Lagrangian Data Representation on Triangles

We suppose the mesh is a triangulation.

Linear Mapping: Through three non-colinear points of a three dimensional space passes a unique plane. That allows for a given triangle of a mesh, to define the unique plane that takes value 1 at some vertex and 0 at the two others. If we denote by i this vertex and T the triangle, we call this function φ_i^{T}, and we can do the same for all the triangles of $\mathcal{D}_i$. Because these functions defined on each triangles are linear, they are also linear along the edges of $\mathcal{D}_i$ and we can join these planes by continuity. Furthermore, these functions vanish on the vertices of the boundaries of $\mathcal{D}_i$. This means we can continuously connect these functions defined on $\mathcal{D}_i$ with the null function outside of $\mathcal{D}_i$. And if we use the convention: $\forall \mathrm{T} \notin \mathcal{D}_i, \quad \varphi_i^{\mathrm{T}} \equiv 0$, we define the basis function associated to node i by

$$\varphi_i^1(\mathbf{x}) = \varphi_i^{\mathrm{T}}(\mathbf{x}), \quad \text{when } \mathbf{x} \in \mathrm{T}. \tag{3.6}$$

This well known continuous linear basis function is represented on Figure 3.6. Superscript 1 stands for the basis function is piecewise of degree one.

We now define the finite subset $\xi^1 = \left\{\varphi_i^1, i \in \mathcal{M}_h\right\}$. Its elements are obviously linearly independent because a linear combination of these function is the null function if and only if all the coefficients of the combination are null. Then ξ^1 is a basis of $\mathscr{W}_h^1 = \mathrm{Span}\left\{\xi^1\right\}$, and $\mathscr{W}_h^1$ is the space of continuous functions that are piecewise linear over each triangle of $\mathcal{M}_h$. In the following, this space will be called $\mathbb{P}^1\left(\mathcal{M}_h\right)$ or simply $\mathbb{P}^1$ when no confusion is possible. $\mathscr{W}_h^1$ is isomorphic to $\mathbb{R}^n$, where n is the number of vertices in $\mathcal{M}_h$ and if $(v_i)_{i\in[\![1,n]\!]}$ is a vector of $\mathbb{R}^n$, it is the coordinates of the function of $\mathscr{W}_h^1$ taking value v_i at node i, in the basis $\left(\varphi_i^1\right)_{i\in[\![1,n]\!]}$.

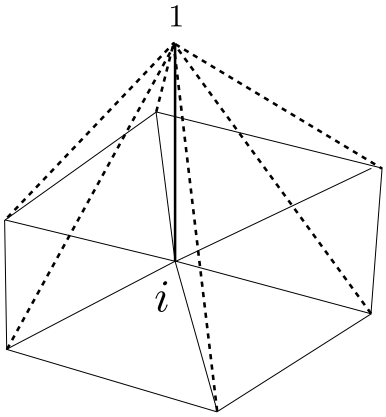


Figure 3.6: ($\mathbb{P}^1$ **Basis Function)** φ_i^1 takes value 1 at i and 0 everywhere else.

Higher Order Mapping: As we have already seen in section 3.1.2, in order to reach a higher order of approximation of the exact solution, we need at least the *projection error* of the desired order. This is possible with a higher order representation of the data, see [80]. We construct the space $\mathscr{W}_h^k$ of continuous functions that are polynomials of order k over each triangle of $\mathcal{M}_h$, and prove that the projection π_h^k of any regular enough function u on $\mathscr{W}_h^k$ is converging toward u proportionally to h^{k+1}, [80]

$$|u - \pi_h^k u| \leqslant Ch^{k+1}.$$

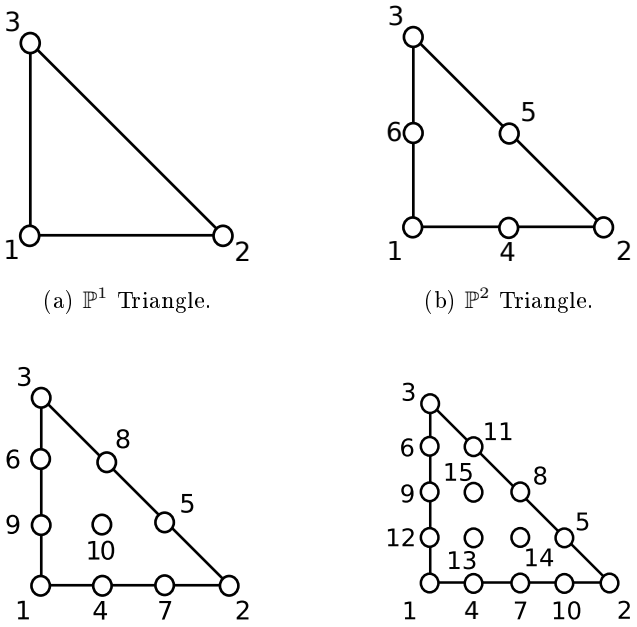
We moreover see further that this higher order of representation of the data allows to build a scheme with *truncation error* of order $k+1$.

Repeating what has been already done for linear basis functions, we are now looking for a continuous function that takes value 1 at node $i \in \mathcal{M}_h$, 0 at any other node of $\mathcal{D}_i$, that is a polynomial of order k inside each triangle of $\mathcal{D}_i$ and that can be continuously joined to the null function outside of $\mathcal{D}_i$. In order to obtain continuous junction along the edges of $\mathcal{D}_i$, we need the polynomial function to be identical on either side of the edge, and because the restriction of our basis function to the edge is also a polynomial of order k, we need $k+1$ degrees of freedom on every edge. That means $k-1$ extra DoFs inside the edge plus the two tips. For sake of simplicity, we place these extra points regularly on the edges. A polynomial function of order k in $\mathbb{R}^2$ can be written as

$$f(x,y) = \sum_{\substack{i,j \in [\![0,k]\!] \\ i+j \leqslant k}} a_{ij} x^i y^j$$

and is then defined by $\frac{(k+1)(k+2)}{2}$ values at different points. We have already $n_{edge} = 3+3(k-1) = 3k$ DoFs along the edges and need then $n_{inside} = \frac{(k-1)(k-2)}{2}$ DoFs inside the triangle. As you can see this last number is zero for $k = 1$ or 2. As we did for the DoFs on the edges, we *equi-distribute* these new points inside the triangle; even though we don't have to... The repartition of these extra DoFs as well as the convention of numbering used along this thesis are shown on Figure 3.7. We also extend the notations $\mathcal{D}_i$ and $\mathcal{C}_i$ for the new DoFs:

- $\mathcal{D}_i$ is still the set of triangles sharing DoF i;



(a) $\mathbb{P}^1$ Triangle. (b) $\mathbb{P}^2$ Triangle.

(c) $\mathbb{P}^3$ Triangle. (d) $\mathbb{P}^4$ Triangle.

Figure 3.7: High order triangles up to 4^{th} polynomial order. The most interesting thing here is our numbering convention.

- $\mathcal{C}_i$ is now the dual cell of i in the associated mesh where each triangle of order k as been cut into k^2 sub-triangles (see Figure 3.8 for the $\mathbb{P}^2$ case).

$\mathcal{M}_h$ now also represents the k^{th}-order mesh and contains the elements, edges and vertices as well as the extra DoFs and the sub-triangles.

The i^{th} high order basis function is well defined as the continuous junction of the unique polynomials of order k defined on each triangle T of $\mathcal{D}_i$, taking value 1 at i and 0 at any other DoF of T. It is extended by continuity by the null function outside $\mathcal{D}_i$. These functions are rather complex to obtain, but they are in fact products of the first order basis functions $\varphi_i^{\mathrm{T},1}$ inside each triangle T of $\mathcal{D}_i$. Here are these expressions, the numbering following the one given on Figure 3.7.

k=2:
- $i = 1..3$
$$\varphi_i^{\mathrm{T},2} = \varphi_i^{\mathrm{T},1}(2\varphi_i^{\mathrm{T},1} - 1)$$
- $i = 4..6$, j, k are the tips of the edge i is part of
$$\varphi_i^{\mathrm{T},2} = 4\varphi_j^{\mathrm{T},1}\varphi_k^{\mathrm{T},1}$$

k=3:
- $i = 1..3$
$$\varphi_i^{\mathrm{T},3} = \frac{1}{2}\varphi_i^{\mathrm{T},1}(3\varphi_i^{\mathrm{T},1} - 1)(3\varphi_i^{\mathrm{T},1} - 2)$$
- $i = 4..9$, j is the vertex of T the nearest to i, k is the other tip of the edge
$$\varphi_i^{\mathrm{T},3} = \frac{9}{2}\varphi_j^{\mathrm{T},1}\varphi_k^{\mathrm{T},1}(3\varphi_j^{\mathrm{T},1} - 1)$$

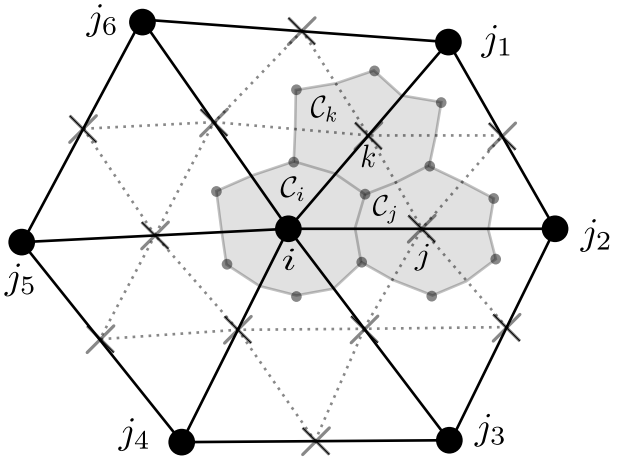


Figure 3.8: Control cells $\mathcal{C}_i$, $\mathcal{C}_j$ and $\mathcal{C}_k$ and sub-triangulation in $\mathbb{P}^2$ formulation.

- $i = 10$

$$\varphi_{10}^{\mathrm{T},3} = 27\varphi_1^{\mathrm{T},1}\varphi_2^{\mathrm{T},1}\varphi_3^{\mathrm{T},1}$$

k=4:

- $i = 1..3$

$$\varphi_i^{\mathrm{T},4} = \frac{1}{3}\varphi_i^{\mathrm{T},1}(4\varphi_i^{\mathrm{T},1} - 3)(2\varphi_i^{\mathrm{T},1} - 1)(4\varphi_i^{\mathrm{T},1} - 1)$$

- $i = 4..9$, j is the vertex of T the nearest to i, k is the other tip of the edge

$$\varphi_i^{\mathrm{T},4} = \frac{16}{3}\varphi_j^{\mathrm{T},1}\varphi_k^{\mathrm{T},1}(4\varphi_j^{\mathrm{T},1} - 1)(2\varphi_j^{\mathrm{T},1} - 1)$$

- $i = 10..12$, j, k are the tips of the edge i is part of

$$\varphi_i^{\mathrm{T},4} = 4\varphi_j^{\mathrm{T},1}\varphi_k^{\mathrm{T},1}(4\varphi_j^{\mathrm{T},1} - 1)(4\varphi_k^{\mathrm{T},1} - 1)$$

- $i = 13..15$, j is the vertex of T the nearest to i

$$\varphi_i^{\mathrm{T},4} = 32\varphi_1^{\mathrm{T},1}\varphi_2^{\mathrm{T},1}\varphi_3^{\mathrm{T},1}(4\varphi_j^{\mathrm{T},1} - 1)$$

We still use the convention $\forall \mathrm{T} \notin \mathcal{D}_i, \quad \varphi_i^{\mathrm{T},k} \equiv 0$ and thus define the k^{th}-order basis function associated to node i by :

$$\varphi_i^k(\mathbf{x}) = \varphi_i^{\mathrm{T},k}(\mathbf{x}), \quad \text{when } \mathbf{x} \in \mathrm{T}. \tag{3.7}$$

Once more the finite subset $\left(\varphi_i^k\right)_{i\in[\![1,\#\mathrm{DoF}]\!]}$ has linearly independant elements and is then a basis of $\mathscr{W}_h^k$. And if $(v_i)_{i\in[\![1,n]\!]}$ is a vector of $\mathbb{R}^n$, n being the number of degrees of freedom in the k^{th}-order mesh, it is the coordinates in this basis of the function of $\mathscr{W}_h^k$ taking value v_i at node i.

For any function $u : \mathbb{R}^2 \to \mathbb{R}$, we can therefore define its projection u_h^k on $\mathscr{W}_h^k$, also denoted by $\pi_h^k u$, by

$$\pi_h^k u = u_h^k = \sum_{i\in\mathcal{M}_h} u\left(\mathbf{x}_i\right)\varphi_i^k. \tag{3.8}$$

This will be often denoted by u_h when the order of approximation is obvious.

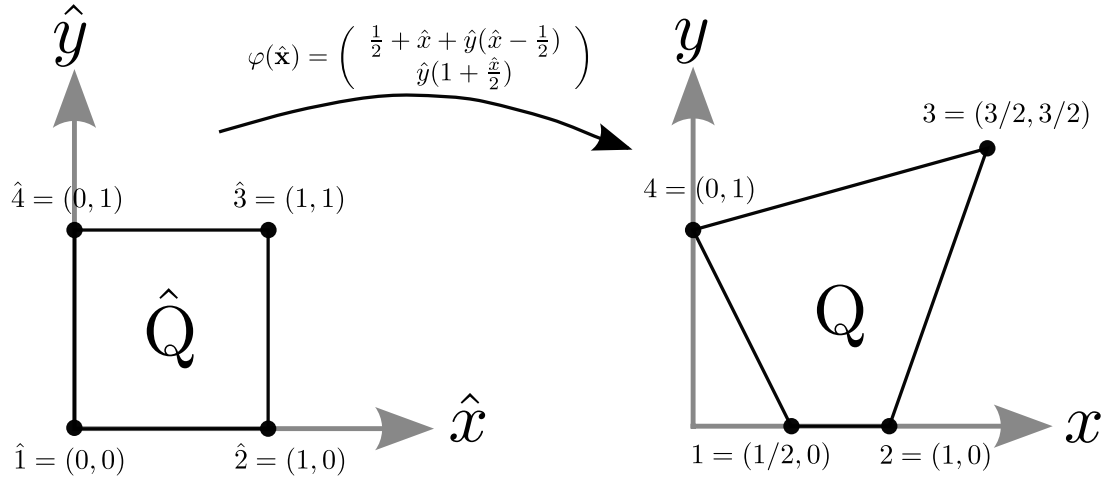


Figure 3.9: If Q is convex, there exists a unique diffeomorphism φ transforming $\widehat{\mathrm{Q}}$ into Q.

3.2.2 Quadrangles Case

In order to define numerical schemes that could be applied on hybrid meshes, we now have to build the same type of basis functions such that for any regular enough function u, its projection on the space spanned by these basis functions is converging toward u at speed h^{k+1}. This is done through what is called $\mathbb{Q}^k$ functions.

$\mathbb{Q}^1$ Representation: Let us consider a general convex quadrangle Q (which means not inevitably regular). We have 4 vertices and would like to build functions inside Q having the same property as $\mathbb{P}^1$ functions inside a triangle:

a) $\varphi_i^{\mathrm{Q}}(\mathbf{x}_j) = \delta_{ij}$;

b) φ_i^{Q} is linear along the edges of Q.

If we consider the familly of function that can be written as

$$f(x,y) = axy + bx + cy + d, \tag{3.9}$$

we obtain easily condition a) because coefficients a, b, c and d are uniquely defined. On the contrary, condition b) is never satisfied but in the "regular" quadrangles (the ones where the edges are two by two orthogonal). That is why the basis functions inside Q are first defined on the reference quadrangular element $\widehat{\mathrm{Q}} = [0;1]^2$ and then mapped on Q through the unique diffeomorphism mapping $\widehat{\mathrm{Q}}$ into Q, see Figure 3.9. It is explained in the following.

If we consider the numbering of the reference element $\widehat{\mathrm{Q}}$ given on Figure 3.9, we have the following reference $\mathbb{Q}^1$ functions:

$$\widehat{\mathscr{Q}}_1^1 = (1-x)(1-y),\ \widehat{\mathscr{Q}}_2^1 = x(1-y),\ \widehat{\mathscr{Q}}_3^1 = xy,\ \widehat{\mathscr{Q}}_4^1 = (1-x)y. \tag{3.10}$$

Superscript '1' here recalls these are the first order basis function on the reference quadrangle. These functions verify obviously above conditions a) and b) in $\widehat{\mathrm{Q}}$. We consider next the following

transformation of the plane :

$$\varphi : \left\{ \begin{array}{ccc} \widehat{\mathrm{Q}} & \longrightarrow & \mathrm{Q} \\ \widehat{\mathbf{x}} & \longmapsto & \displaystyle\sum_{i=1}^{4} \widehat{\mathscr{Q}}_i\left(\widehat{\mathbf{x}}\right) \mathbf{x}_i \end{array} \right. \tag{3.11}$$

It is a $\mathcal{C}^1$-diffeomorphism if and only if Q is convex. We then always assume that our quadrangular elements are convex. If not, they are cut into two triangles! That allows us to define $J[\mathrm{Q}]$ as the determinant of the Jacobian of φ. φ is not a linear transformation as the $\widehat{\mathscr{Q}}_i$ are not linear either, but it is a linear transformation along the edges of the quadrangles because the $\widehat{\mathscr{Q}}_i$ are linear along the edges of $\widehat{\mathrm{Q}}$. Thus, the functions defined by :

$$\mathscr{Q}_i^1 = \widehat{\mathscr{Q}}_i^1 \circ \varphi^{-1} \tag{3.12}$$

verify the conditions a) and b) inside Q. They are called the $\mathbb{Q}^1$ basis functions in Q. In a hybrid mesh, $\mathbb{P}^1$ and $\mathbb{Q}^1$ functions can be joined continuously and a function of the approximated space $\mathscr{W}_h$ is well defined by its value at the degrees of freedom.

$\mathbb{Q}^k$ Representation: In the case of a higher order representation in Ω, we also wish to define higher order basis functions in the quadrangle. A generalization of formula (3.9) at order k would be the familly of function of the form

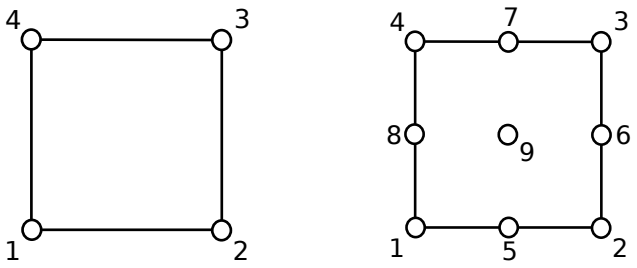
$$f(x,y) = \sum_{i,j \in [\![0,k]\!]} a_{ij} x^i y^j,$$

which means $(k+1)^2$ DoFs inside each quadrangle in order to be well-defined. As remarked in the case $k = 1$, such a function taking value 1 at a vertex of Q and 0 at the other DoFs is not polynomial of order k along the edges of Q, as soon as the edges of Q are not perpendicular. It can thus not be joined continuously with $\mathbb{P}^k$ functions, if Q is surrounded by triangles. We then use the same trick, defining first the k^{th}-order basis functions on the reference quadrangle $\widehat{\mathrm{Q}}$ and then transport them to Q via formula (3.12). About the degrees of freedom: in order to be consistant with the $\mathbb{P}^k$ formulation we need $k-1$ DoFs inside the edges (regularly distributed), which means $(k+1)^2 - 4k = (k-1)^2$ DoFs inside each quadrangle that are also equi-distributed. The common distribution of the degrees of freedom for the reference quadrangle are given in Figure 3.10.

3.2.3 Time-Dependent Problem Treatment

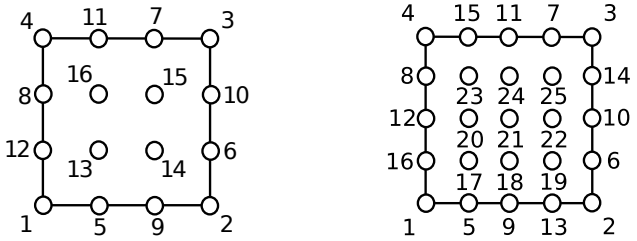
To consider unsteady problems, we have actually two choices of schemes. The first one is to discretize the time derivative terms by finite differences and then obtain a time marching scheme that would solve a space problem at each time step. On the other hand, we could approximate the unsteady solution in the space-time domain. Unfortunately, we are not going to present any unsteady results at the end. But in the theoritical part on $\mathcal{RDS}$ (Chapter 4), we will always extend the presented concepts to unsteady cases when possible.

We are here interested in the space-time formulation and we then need elements and basis functions in space-time [7, 68]. We introduce what we call *prismatic elements*, which can be considered as the translation into the time direction of the space meshing. Prismatic elements



(a) $\mathbb{Q}^1$ Quadrangle.

(b) $\mathbb{Q}^2$ Quadrangle.



(c) $\mathbb{Q}^3$ Quadrangle.

(d) $\mathbb{Q}^4$ Quadrangle.

Figure 3.10: High order quadrangles up to 4^{th} polynomial order.

associated to triangles and quadrangles are represented on Figure 3.11. For the form functions, we decouple the influence of space and time and define the basis function at node i as the product of the k^{th}-order basis function in space at node i by the one dimensional ℓ^{th}-order time basis function

$$\varphi_i^k(\mathbf{x}).\lambda_i^\ell(t). \tag{3.13}$$

3.3 Appeals of Higher Order Schemes

We begin this section by a quick summary of the ideas already presented in this chapter. We first gave an abstract definition of a numerical scheme and explained what a k^{th}-order scheme is. In particular, we have seen that for a $(k+1)^{\text{th}}$-order scheme we generally need a polynomial representation of the solution of order at least k. In the last paragraph, we have eventually presented domain discretization and k^{th}-order representation of the data on this discretization. But what is the goal of higher order schemes ? What do we win with this much more complex representation of the solution ?

To be as clear as possible, we are going to treat the problem at a constant approximation error ε. If the scheme is of order k, there exists a proportionality coefficient C_k such that the behaviour of the error can be modeled by

$$\varepsilon \leqslant C_k h^k.$$

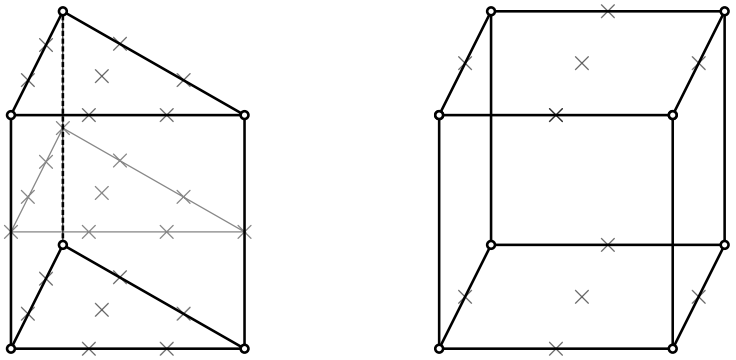


Figure 3.11: (**Prismatic Elements**) Left: $\mathbb{P}^3$ in space, $\mathbb{P}^2$ in time. Right: $\mathbb{Q}^2$ in space, linear in time.

k \ ε	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
1	1.00E+08	1.00E+10	1.00E+12	1.00E+14	1.00E+16
2	4.00E+04	4.00E+05	4.00E+06	4.00E+07	4.00E+08
3	4.18E+03	1.94E+04	9.00E+04	4.18E+05	1.94E+06
4	1.60E+03	5.06E+03	1.60E+04	5.06E+04	1.60E+05
5	9.95E+02	2.50E+03	6.28E+03	1.58E+04	3.96E+04
6	7.76E+02	1.67E+03	3.60E+03	7.76E+03	1.67E+04
7	6.81E+02	1.31E+03	2.54E+03	4.90E+03	9.46E+03
8	6.40E+02	1.14E+03	2.02E+03	3.60E+03	6.40E+03
9	6.27E+02	1.05E+03	1.75E+03	2.91E+03	4.86E+03
10	6.31E+02	1.00E+03	1.58E+03	2.51E+03	3.98E+03

Figure 3.12: Maximal number of DoFs needed to obtain precision ε at order k. Coefficients S and C_k have been normalized. The problem is supposed to be two dimensional.

Let consider a structured grid composed only of triangles, as the one presented on Figure 3.2. If we set $n_{\mathrm{V}}, n_{\mathrm{E}}, n_{\mathrm{DoFs}}$ and n_{T} respectively the number of vertices, edges, degrees of freedom and triangles inside the mesh and S the surface of the domain, we have the relations

$$n_{\mathrm{T}} = \frac{2S}{h^2} = 2n_{\mathrm{V}} \quad \text{and} \quad n_{\mathrm{DoFs}} = n_{\mathrm{V}} + (k-1)n_{\mathrm{E}} + \frac{(k-1)(k-2)}{2} n_{\mathrm{T}},$$

see section 3.2 for last formula. And if we apply Euler Formula (3.4), we obtain

$$n_{\mathrm{DoFs}} \leqslant k^2 S \left(\frac{C_k}{\varepsilon}\right)^{d/k} \tag{3.14}$$

where d stands for the dimension of the domain (2 in our case). We represent the maximal number of DoFs needed to obtain precision ε at order k in tabular 3.12. It is a simple exercice to see that n_{DoFs} is equivalent to k^2 when k goes to infinity, when the coefficient C_k is supposed independent of k. Then for a given sought precision ε, there always exists an optimal order with

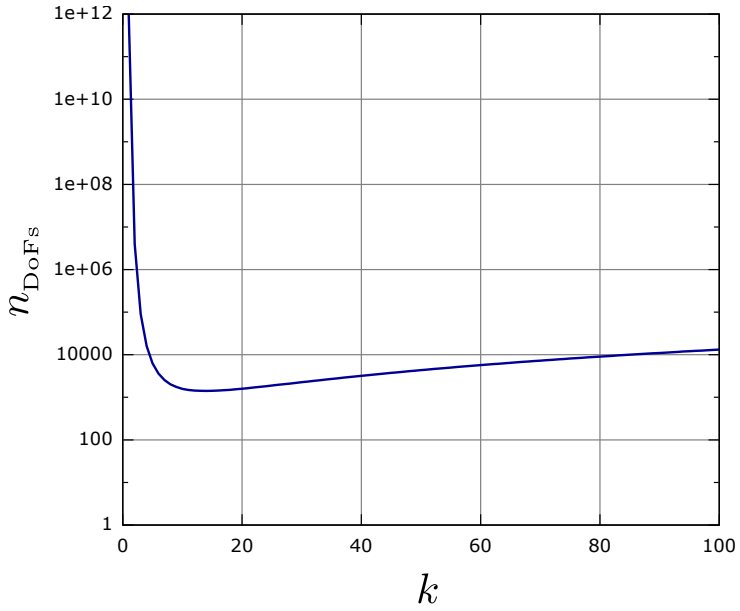


Figure 3.13: Maximal number of DoFs needed to reach accuracy 10^{-6} for order of approximation $k = 1 \dots 100$.

respect to the number of DoFsneeded. And this is really important as n_{DoFs} represents the size of the finite dimensional problem to solve at the end. As one can see on figure 3.13, even if, as for tabular 3.12, a value of 1 has been taken for S and C_k in order to simplify the calculation, there is a huge factor between the number of DoFs needed at first and optimum order to reach 6^{th} order of accuracy.

Furthermore, as we will next see, in the case of the Residual Distribution Schemes the solving algorithm treats the problem element per element. The less elements we get in the mesh, the less computations we have to do. We have already seen that in a k^{th}-order triangulation, the number of elements is proportional to $\varepsilon^{-d/k} \underset{k \to +\infty}{\longrightarrow} 1$. Starting from this point of view, we would like to have the largest possible order. What is hidden is that increasing order of approximation provides less elements but on the other hand more work to do per element. And as the number of triangles is exponentially decreasing toward 1, there once more must exist an optimum order.

Part II

Residual Distribution Schemes

Chapter 4

Introduction to Residual Distribution Schemes

Until now, we have been simplifying the general framework of the problem along the pages. We started by the very general case (2.1) and restricted it for sake of simplicity. From now on, the trend is being inverted, and the problem is going to be complicated along the chapters. For this introduction, we are going to consider the simplest framework for the conservation laws. But, even if we start here by the well described $\mathbb{P}^1$ steady scalar non viscous case, we still aim at explaining the end of this manuscript the treatment of a 3D, $\mathbb{P}^k$, Navier-Stokes problem.

We are looking for the value of a scalar unknown u verifying, on a two dimensional domain Ω, a simple conservation equation

$$\begin{aligned} & \operatorname{div}\left(\vec{\boldsymbol{\mathcal{F}}}(u)\right) = 0 \\ + \quad & \text{Boundary Conditions (Dirichlet, Neumann, strong or weak...)} \end{aligned} \tag{4.1}$$

As we did before, the flux vector $\vec{\boldsymbol{\mathcal{F}}}$ can be split into its two one dimensional components, F and G. For a real problem, we would have of course to add some boundary conditions, but in order to simplify the explanation, we are going to ignore them. In fact, one could use the homogeneous boundary condition $u_{|\partial\Omega} = 0$ and obtain exactly the same results. For those interested in our weak or strong formulation of some Dirichlet, Neumann, *aso...* boundary conditions, more details are given in Section 5.4.

4.1 Principle

The formulation of the Residual Distribution Schemes ($\mathcal{RDS}$) applied to equation (4.1) is rather simple to understand. However, a sound mathematical framework is still not available at the present. Often, geometrical and more or less qualitative arguments have been used to study the properties of the schemes. Moreover, as soon as we treat vectorial problems or want to use any kind of high order method, the formal constructions developed in the simple scalar $\mathbb{P}^1$ case do not apply any longer. Most properties are nevertheless assumed to be still valid and anyway verified numerically. For these reasons, we first present how the scheme is built, without giving any formal justification, next show its computational properties (consistence, stability,...) and

only at the end give evidences that the solution of such a scheme approximates the exact solution of (4.1) with the desired order.

As the construction of such a scheme is rather simple, and mathematicians liking simple things, it would be very interesting to find a complete "Residual" formulation of equation (4.1), defined on the continuous domain. It could really help to understand the properties of $\mathcal{RDS}$, obviously, but also all the numerical formulations on conservative systems. In particular, it is very hard to show that a $\mathcal{RDS}$ has an unique solution in a given functional space and we need to see the problem an other way to be able to answer to this question.

4.1.1 Residual and Residual Distribution

For each element, we define the **Global Residual** or **Element Residual** as

$$\Phi^{\mathrm{T}} = \int_{\mathrm{T}} \mathrm{div}\left(\vec{\boldsymbol{\mathcal{F}}}(u)\right) d\mathbf{x} = \int_{\partial \mathrm{T}} \vec{\boldsymbol{\mathcal{F}}}(u).\vec{\mathbf{n}}\, ds, \tag{4.2}$$

where T does not have to be a triangle and $\vec{\mathbf{n}}$ is the outward unit normal. This quantity represents the global flux $\vec{\boldsymbol{\mathcal{F}}}$ leaving the triangle. If we look at the exact solution of the equation on the continuous domain (4.1), the residual should be zero on every triangle. This could be one way to write the scheme: nullify the global amount of flux entering or leaving each triangle. However, we want to define the scheme point-wise. To be able to write an equation for each degree of freedom, we nullify the global flux entering some control cell around each DoF.

This is obtained in practice by distributing Φ^{T} to each DoF of the element with a certain **distribution coefficient** β_i^{T}

$$\Phi_i^{\mathrm{T}} = \beta_i^{\mathrm{T}} \Phi^{\mathrm{T}}, \tag{4.3}$$

and for each degree of freedom of the mesh, gather the received information:

$$\sum_{\mathrm{T} \in \mathcal{D}_i} \Phi_i^{\mathrm{T}}.$$

Φ_i^{T} is usually called the **Nodal Residual**. Here is the core of the method. They are many possibilities of distributing the global residual, each one of them having a different combination of properties: monotonicity, linearity preservation, higher order accuracy, upwinding, etc... We are going to detail those words in the next section.

If we want the scheme to be conservative, no information must suddenly appear or disappear. In other words, we need the global residual to be exactly distributed in each element

$$\sum_{i \in \mathrm{T}} \Phi_i^{\mathrm{T}} = \Phi^{\mathrm{T}}. \tag{4.4}$$

This can be straightforwardly rewritten in term of distribution coefficients:

$$\sum_{i \in \mathrm{T}} \beta_i^{\mathrm{T}} = 1.$$

As we see in the next subsection, gathering all the nodal residuals sent to a node corresponds in some simple cases to estimate the balance of flux entering some control cell around i. We wish then to nullify this global flux, and the scheme writes

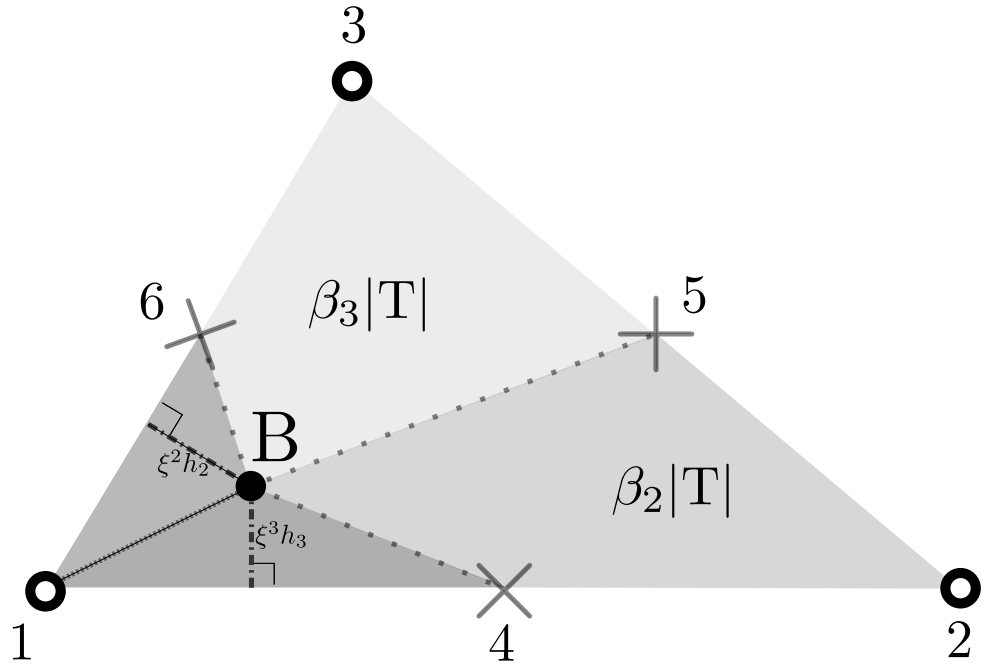


Figure 4.1: Find barycentric coordinates (ξ^1, ξ^2, ξ^3) of B such that quadrilaterals 14B6, 24B5 and 36B5 have areas $\beta_1|\mathrm{T}|$, $\beta_2|\mathrm{T}|$ and $\beta_3|\mathrm{T}|$ respectively. $|14\mathrm{B}6| = (\xi^2 + \xi^3)\frac{|\mathrm{T}|}{2} = \beta_1|\mathrm{T}|$ and the same reasoning being true for the two other vertices, one gets $\xi^i = 1 - 2\beta_i,\ i = 1, \ldots, 3$

$$\sum_{\mathrm{T} \in \mathcal{D}_i} \Phi_i^{\mathrm{T}} = 0,\ \forall i \in \mathcal{M}_h. \tag{4.5}$$

4.1.2 Geometrical Interpretation in the $\mathbb{P}^1$ Case

Let consider a $\mathbb{P}^1$ mesh. Each triangle has three degrees of freedom. Because $\sum_{i \in \mathrm{T}} \beta_i^{\mathrm{T}} = 1$, it is possible to define an inner point B of T, such that for each vertex i, the quadrilateral generated by node i, the two mid-edges next to i and B has area $\beta_i^{\mathrm{T}}|\mathrm{T}|$. This point has barycentric coordinates $(1 - 2\beta_1, 1 - 2\beta_2, 1 - 2\beta_3)$, see figure 4.1. If we define the new control cell associated to node i with these quadrilaterals, and denote it by $\mathcal{C}_i^\beta$, we obtain that the integral of equation (4.1) on each control cell gives the expression of the scheme (4.5)

$$\forall i \in \mathcal{M}_h,\ \sum_{\mathrm{T} \in \mathcal{D}_i} \Phi_i^{\mathrm{T}} = \int_{\mathcal{C}_i^\beta} \mathrm{div}\left(\vec{\boldsymbol{\mathcal{F}}}(u)\right) d\mathbf{x} = 0.$$

Then, the control cell defines a discrete closed ways in the domain through which the global entering flux is null. Linking the different control cells together, we obtain a new meshing, dual of the original one $(\mathcal{M}_h)$. It is obvious that the balance of flux entering any sub-domain of this dual mesh is null. If we now consider the dual control cells as the indivisible two dimensional entities of the domain, or as the infinitesimal surfaces of Ω, equation (4.1) has been discretized on the dual mesh. But β_i^{T} depends on the value of the solution u_h. Then the problem writes:

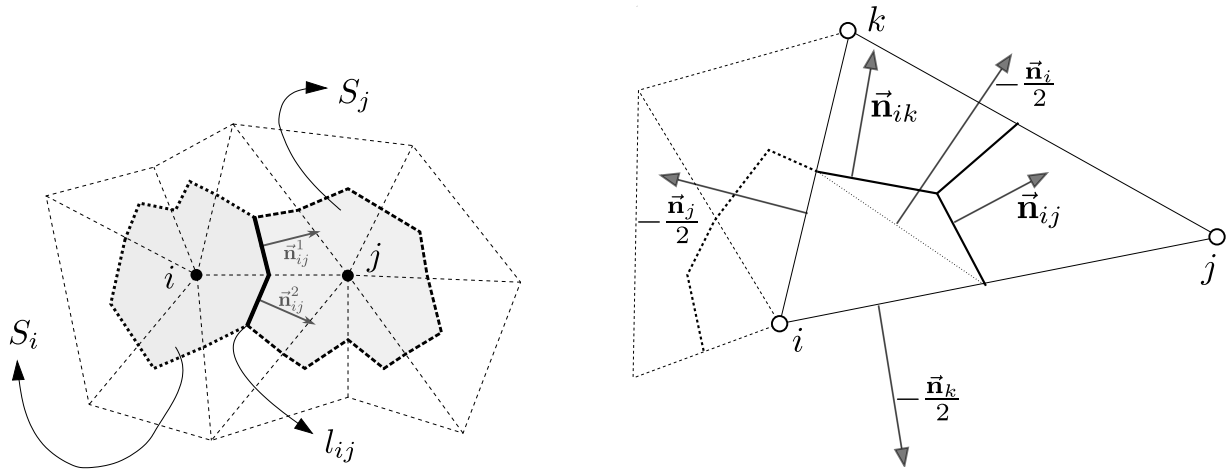


Figure 4.2: $\mathcal{FV}$ scheme. Neighboring cells S_i and S_j (left) and cell normals (right)

Find u_h and $\beta_i^T(u_h)$ such that $\vec{\nabla}.u_h$ is zero over the dual mesh associated to the distribution coefficients β_i^T.

The control cells define a discrete closed ways in the domain through which the global entering flux is null : equation (4.1) is solved on the dual mesh.

4.1.3 Links with Other Classical Formulations

We here present some relations between the $\mathcal{RD}$ framework and other classes of classical numerical schemes. The goal is just to show the proposed formulation can be seen as another point of view for the treatment of the conservative equations. The comparison in the following examples however usually stops as soon as we leave the simple $\mathbb{P}^1$ scalar case. If possible, more details will be given.

Finite Volume Schemes: The following explanation essentially comes from [2] and Mario Ricchuito's thesis [89]. Symbol $\mathcal{FV}$ denotes the finite volume schemes. All geometrical entities are illustrated on Figure 4.2.

We consider a meshing of a domain, and for any DoF i its associated median dual cell $\mathcal{C}_i$, generated by the midpoint of the edges and the barycentric centers of the elements i is part of, see Figure 4.2. The new meshing constituted by the DoFs and their median dual cells is called the *median dual mesh*. We consider a piecewise constant numerical approximation over the dual cells:

$$u_h \in \left\{ f : \Omega \longrightarrow \mathbb{R};\ \forall i \in \mathcal{M}_h, \quad f_{|\mathcal{C}_i} \text{ is constant } \right\}.$$

$\mathcal{FV}$ formulation of continuous scalar equation (4.1) reads

$$\sum_{l_{ij} \in \mathcal{C}_i} \int_{l_{ij}} \vec{\boldsymbol{\mathcal{H}}}\left(u_i, u_j, \bar{\mathbf{n}}\right) \, dl = 0. \tag{4.6}$$

where $\vec{\boldsymbol{\mathcal{H}}}(u, v, \bar{\mathbf{n}})$ stands for the $\mathcal{FV}$ numerical flux, l_{ij} is the portion of $\partial\mathcal{C}_i$ separating $\mathcal{C}_i$ from $\mathcal{C}_j$ (see Figure 4.2) and $\bar{\mathbf{n}}$ is the outward unit normal.

A large set of $\mathcal{FV}$ scheme is included in the Q-scheme framework. This type of schemes is based on the family of flux functions defined as

$$\begin{aligned}\overrightarrow{\boldsymbol{\mathcal{H}}}(u_i,u_j,\vec{\mathbf{n}}_{ij}) &= \frac{1}{2}\left(\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_i)+\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_j)\right).\vec{\mathbf{n}}_{ij}^1 - Q(u_i,u_j,\vec{\mathbf{n}}_{ij}^1)(u_i-u_j)\\ &+ \frac{1}{2}\left(\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_i)+\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_j)\right).\vec{\mathbf{n}}_{ij}^2 - Q(u_i,u_j,\vec{\mathbf{n}}_{ij}^2)(u_i-u_j)\end{aligned} \tag{4.7}$$

with $Q(u,v)$ being a *dissipation matrix* (*e.g.* Roe's absolute value matrix, see [98] or [89] page 61), $\overrightarrow{\boldsymbol{\mathcal{F}}_h}$ being the linear interpolant of the flux function $\overrightarrow{\boldsymbol{\mathcal{F}}}$ and $\vec{\mathbf{n}}_{ij}^1$ and $\vec{\mathbf{n}}_{ij}^2$ being defined on the left side of Figure 4.2. They all verify the consistence property of the $\mathcal{FV}$ schemes:

$$\overrightarrow{\boldsymbol{\mathcal{H}}}(u,u)=\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u) \Longrightarrow \forall i\in\mathcal{M}_h, \int_{\partial\mathcal{C}_i}\overrightarrow{\boldsymbol{\mathcal{H}}}(u_i,u_i,\bar{\mathbf{n}})\,dl=0. \tag{4.8}$$

Then scheme writes:

$$\begin{aligned}\sum_{l_{ij}\in\mathcal{C}_i}\int_{l_{ij}}\overrightarrow{\boldsymbol{\mathcal{H}}}\left(u_i,u_j,\bar{\mathbf{n}}\right)\,dl &= 0\\ &= \frac{1}{2}\sum_{\mathrm{T}\in\mathcal{D}_i}\sum_{\substack{j\in\mathrm{T}\\ j\neq i}}\Big\{\left(\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_j)-\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_i)\right).\vec{\mathbf{n}}_{ij}\\ &\quad -Q(u_i,u_j,\vec{\mathbf{n}}_{ij})(u_i-u_j)\Big\}\\ &= \sum_{\mathrm{T}\in\mathcal{D}_i}\Phi_i^{\mathrm{T},\mathcal{FV}}.\end{aligned}$$

$\vec{\mathbf{n}}_{ij}$ is defined on figure 4.2 and since the boundary of S_i is closed, one has

$$\sum_{\mathrm{T}\in\mathcal{D}_i}\sum_{j\in\mathrm{T}}\overrightarrow{\boldsymbol{\mathcal{F}}_h}(U_i).\vec{\mathbf{n}}_{ij}=0,$$

what has been subtracted in the above equation.

The only thing left to check in order to prove this class of $\mathcal{FV}$ schemes is included in the $\mathcal{RD}$ framework is the conservative property in each element: the quantity of information sent to the nodes must be equal to the global residual of the element.

$$\begin{aligned}\sum_{i\in\mathrm{T}}\Phi_i^{\mathrm{T},\mathcal{FV}} &= \sum_{i\in\mathrm{T}}\frac{1}{2}\sum_{j\in\mathrm{T},j\neq i}\Big\{\left(\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_j)-\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_i)\right).\vec{\mathbf{n}}_{ij}-Q(u_i,u_j,\vec{\mathbf{n}}_{ij})(u_i-u_j)\Big\}\\ &= \sum_{i\in\mathrm{T}}\frac{1}{2}\sum_{j\in\mathrm{T},j\neq i}\left(\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_j)-\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_i)\right).\vec{\mathbf{n}}_{ij}\\ &= \sum_{i\in\mathrm{T}}\frac{\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_i).\vec{\mathbf{n}}_i}{2}\\ &= \int_{\partial\mathrm{T}}\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u).\bar{\mathbf{n}}\,ds\\ &= \Phi^{\mathrm{T}}\end{aligned}$$

This shows that any finite volume scheme operating on the median dual cells with a Q-form numerical flux function defined in (4.7) is equivalent to the RD scheme with the local nodal residuals

$$\Phi_i^{\mathrm{T}} = \sum_{j\in \mathrm{T}, j\neq i} \left\{ \left(\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_j) - \overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_i)\right).\vec{\mathbf{n}}_{ij} - Q(u_i, u_j, \vec{\mathbf{n}}_{ij})(u_j - u_i)\right\},$$

obtained with a continuous piecewise linear approximation of the flux. Note that the analysis is general and can be extended to nonlinear problems and systems. Moreover, as shown in [89] page 62, it applies to general $\mathcal{FV}$ numerical fluxes and not only to (4.7). Surprisingly, starting from the piecewise constant $\mathcal{FV}$ approximation, we arrived to a scheme based on a continuous flux approximation which, moreover, respects all the assumptions of the Lax-Wendroff theorem presented in next section.

Galerkin Finite Element Method: It is well known the Finite Element Method ($\mathcal{FE}$) enjoys a complete mathematical formulation which transforms formally the strong continuous problem (4.1) into its weak form, and the two formulations are consistent. We consider here its $\mathbb{P}^1$ numerical resolution. We have in that case to solve the finite dimensional problem:

$$\int_\Omega \overrightarrow{\nabla\psi_i}.\overrightarrow{\boldsymbol{\mathcal{F}}}(u_h)\, d\mathbf{x} = 0, \quad \forall i \in \mathcal{M}_h. \tag{4.9}$$

ψ_i denotes the $\mathbb{P}^1$ basis function associated to node i. As explained in the introduction of this chapter, the boundary conditions have been neglected or supposed to be homogeneous Dirichlet condition. Then, if the flux $\overrightarrow{\boldsymbol{\mathcal{F}}}$ is continuously approximated by its $\mathbb{P}^1$ projection $\overrightarrow{\boldsymbol{\mathcal{F}}}_h$, $\overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}}_h(u_h)$ is constant over every element and we obtain

$$\begin{aligned}
\forall i \in \mathcal{M}_h, \quad \sum_{\mathrm{T}\in\mathcal{D}_i} \int_{\mathrm{T}} \psi_i \overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}}_h(u_h)\, d\mathbf{x} &= 0 \\
&= \sum_{\mathrm{T}\in\mathcal{D}_i} \frac{1}{3} \int_{\mathrm{T}} \overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}}_h(u_h)\, d\mathbf{x} \\
&= \sum_{\mathrm{T}\in\mathcal{D}_i} \frac{1}{3}\Phi^{\mathrm{T}}.
\end{aligned}$$

This shows the $\mathbb{P}^1$ Galerkin Finite Element Method is a $\mathbb{P}^1$ centered Residual Distribution Scheme with uniform constant distribution coefficients:

$$\beta_i^{\mathcal{FE}} = \frac{1}{3}.$$

Petrov-Galerkin Formulation: The Galerkin Finite Element Method is known to be unstable. This can be easily shown in the case of a constant advection problem (see [1, 73, 64]):

$$\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla u} = 0. \tag{4.10}$$

A new class of schemes has been developed [73, 25, 72] in order to stabilize the $\mathcal{FE}$ in the case of conservation laws; they are called the *Petrov-Galerkin* scheme and just add to the Galerkin formulation a stabilization term. They are all included into the formulation:

$$\int_\Omega \overrightarrow{\nabla\psi_i}.\overrightarrow{\boldsymbol{\mathcal{F}}}(u_h)\, d\mathbf{x} + \sum_{\mathrm{T}\in\mathcal{D}_i} \int_{\mathrm{T}} \left(\frac{\partial\overrightarrow{\boldsymbol{\mathcal{F}}}}{\partial u}.\overrightarrow{\nabla\psi_i}\right) \bar{\boldsymbol{\tau}}\, \overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}}(u_h)\, d\mathbf{x} = 0, \quad \forall i \in \mathcal{M}_h. \tag{4.11}$$

$\bar{\boldsymbol{\tau}}$ is a matrix of local nondimensionalization which characteristic size must be proportional to $\frac{h}{(\|\vec{\mathbf{u}}\|+c)}$. And if we use the notation

$$k_i^{\mathrm{T}} = \int_{\mathrm{T}} \frac{\partial \overrightarrow{\boldsymbol{\mathcal{F}}}}{\partial u}.\overrightarrow{\nabla \psi_i}\, d\mathbf{x}, \tag{4.12}$$

and suppose the advection wind $\dfrac{\partial \overrightarrow{\boldsymbol{\mathcal{F}}}}{\partial u}$ to be constant inside T, we obtain that $\mathbb{P}^1$ Petrov-Galerkin schemes can be rewritten into the form

$$\forall i \in \mathcal{M}_h, \quad \sum_{\mathrm{T}\in\mathcal{D}_i} \left(\frac{1}{3} + \frac{k_i^{\mathrm{T}}\bar{\boldsymbol{\tau}}}{|\mathrm{T}|}\right) \Phi^{\mathrm{T}},$$

which means they fit the $\mathcal{RDS}$ formalism with distribution coefficients

$$\beta_i^{\mathrm{T}} = \frac{1}{3} + \frac{k_i^{\mathrm{T}}\bar{\boldsymbol{\tau}}}{|\mathrm{T}|}.$$

This is unfortunately not true in the general case, as the extra dissipative term in (4.11) cannot be expressed in terms of k_i^{T}.

Another thing to observe is that this dissipative term brings to the scheme some kind of upwind bias in the distribution, which is one way to explain the stabilizing character of this term. In particular, because $\overrightarrow{\nabla\psi_i}$ is perpendicular to the edge opposite to i and points toward node i, k_i^{T} is positive when i is downstream and negative when i is upstream. Then the constant distribution coefficient $\beta_i = 1/3$ of the pure Galerkin $\mathcal{FE}$ formulation is modulated by a coefficient that measures the power and the direction of the advection inside the element. One can look at [89] or [3] for an energy stability study. It gives a better understanding of the stabilization mechanism but also of the $\mathcal{RD}$ stability. One has to remember that the schemes with an upwind character are always more stable, as they push the information in the direction of the advection and therefore always dissipate the possible numerical errors.

$\mathcal{RDS}$ is a particular Galerkin Scheme The following idea has first been expressed in 1993 during the first von Karman Institute for Fluid Dynamics Lecture Series or in [28]. It consists in claiming $\mathcal{RDS}$ is a particular *finite element* weak formulation with modified basis functions. That for, we define what we call the *Bubble Functions* γ^{T}. It is defined over each element of the mesh as the unique piecewise linear continuous form function taking value 1 at the barycentric center of T and 0 over the edges, see Figure 4.3. We can then define

$$\mathscr{N}_i^{\mathrm{T}} = \varphi_i^{\mathrm{T},1} + \alpha_i^{\mathrm{T}}\gamma^{\mathrm{T}} \tag{4.13}$$

as a new linear form function over the element, with α_i^{T} a fitting parameter. The extra nodal form function $\alpha_i^{\mathrm{T}}\gamma^{\mathrm{T}}$ will also be denoted by γ_i^{T}. In order the scheme stays conservative, we need to ensure the following condition:

$$\sum_{i\in\mathrm{T}} \mathscr{N}_i^{\mathrm{T}} = 1 \Longrightarrow \sum_{i\in\mathrm{T}} \gamma_i^{\mathrm{T}} = 0. \tag{4.14}$$

Let us apply the *finite element* theory to equation (4.1) with the approximated functional space being spanned by the $\mathscr{N}_i^{\mathrm{T}}$. We furthermore assume that

$$\forall \mathrm{T} \in \mathcal{M}_h, \forall i \in \mathcal{M}_h, \quad \alpha_i^{\mathrm{T}} = 3\beta_i^{\mathrm{T}} - 1. \tag{4.15}$$

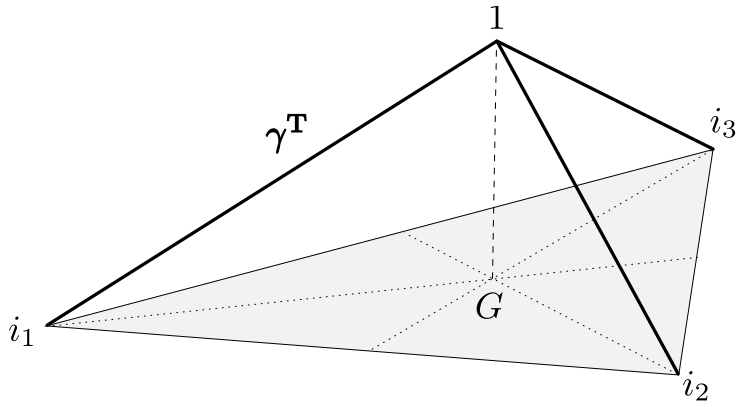


Figure 4.3: (**Bubble Function**) This shape function allows to modify the space of approximation while maintaining the continuous representation of the variable because $\gamma^{\mathrm{T}}{}_{|\partial \mathrm{T}} = 0$.

Then in $\mathbb{P}^1$, the scheme writes

$$\begin{aligned} \forall i \in \mathcal{M}_h, \quad \sum_{\mathrm{T}\in\mathcal{D}_i} \int_{\mathrm{T}} \mathscr{N}_i^{\mathrm{T}} \mathrm{div}\left(\vec{\boldsymbol{\mathcal{F}}}_h(u_h)\right) d\mathbf{x} &= 0 \\ &= \sum_{\mathrm{T}\in\mathcal{D}_i} \int_{\mathrm{T}} \mathscr{N}_i^{\mathrm{T}} d\mathbf{x} \frac{\boldsymbol{\Phi}^{\mathrm{T}}}{|\mathrm{T}|} \\ &= \sum_{\mathrm{T}\in\mathcal{D}_i} \beta_i^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}}, \end{aligned}$$

which is exactly the $\mathbb{P}^1$ $\mathcal{RD}$ scheme. This formulation can be straightforwardly extended to 3D. Unfortunately, we have trouble to extend this idea to higher order formulation. It would be possible if

$$\alpha_i^{\mathrm{T}} = \frac{\Phi_i^{\mathrm{T},k} - \Psi_i^{\mathrm{T},k}}{\int_{\mathrm{T}} \gamma^{\mathrm{T}} \mathrm{div}\left(\vec{\boldsymbol{\mathcal{F}}}_h(u_h)\right) d\mathbf{x}} \tag{4.16}$$

were always defined. But it is not always the case, as $\mathrm{div}\left(\vec{\boldsymbol{\mathcal{F}}}_h(u_h)\right)$ is no more constant in $\mathbb{P}^k, k > 1$, and can take positive as negative values inside T.

4.2 Properties of $\mathcal{RDS}$

This section is devoted to the definition of the numerical properties of $\mathcal{RDS}$. This will help to understand the construction of the high order residual schemes that are going to be presented in the next chapter.

4.2.1 Consistency

We start by verifying under which conditions the computed solution is really an approximation of the weak solution of problem (4.1). The following Lax-Wendroff-Like Theorem has been

demonstrated in 2002 by Abgrall and Roe and a complete proof can be found in the associated article [9]. Although [9] treats the complete unsteady case, we are first going to consider only the steady problem. The demonstration is almost completed in that case and we will then just give a remark on how to deal with the time derivative terms.

To begin with, we need to define the framework we are going to work in.

Assumption 4.1

The mesh $\mathcal{M}_h$ is conformal and regular. The word conformal has already been defined in section 3.1.3 and on Figure 3.3. By regular we mean that the triangles are roughly the same size, more precisely that there exist two constants C_1 and C_2 such that the ratio of two heights of any triangle of the mesh stands between C_1 and C_2, as already expressed in (3.3).

If $\mathcal{M}_h$ is a mesh verifying assumption 4.1 and $\mathcal{D}_h^k$ is the set of dual volumes associated with the degrees of freedom $i \in \mathcal{M}_h$ ($\mathbb{P}^k$), we define the following vectorial subspaces

$$\mathscr{W}_h^k = \mathbb{P}^k(\mathcal{M}_h),$$

$$\mathscr{X}_h^k = \left\{ v_h; \forall \mathcal{D} \in \mathcal{D}_h^k, \quad v_{h|\mathcal{D}} = \text{ constant } \right\}.$$

As defined in (3.8), π_h^k defines the piecewise k^{th} order interpolation of any function defined at any degree of freedom $i \in \mathcal{M}_h$. Then the *mass lumping* operator :

$$L_h^k : \left\{ \begin{array}{ccc} \mathscr{W}_h^k & \longrightarrow & \mathscr{X}_h^k \\ v & \longmapsto & \displaystyle\sum_{i \in \mathcal{M}_h} v(\mathbf{x}_i) \chi_{\mathcal{D}_i} \end{array} \right.$$

where $\chi_{\mathcal{D}_i}$ is the characteristic function of cell $\mathcal{D}_i$, defines an isomorphism between $\mathscr{W}_h^k$ and $\mathscr{X}_h^k$ which reciprocal function is the function π_h^k restricted to $\mathscr{X}_h^k$.

The next assumption must be seen as asserting continuity of the local nodal residual Φ_i^{T} with respect to the nodal values of u_h inside T. In particular, when u_h is constant over T, $\Phi_i^{\mathrm{T}}(u_h)$ must be zero.

Assumption 4.2

Let $\mathcal{M}_h$ be a meshing verifying assumption 4.1. Then for any $C \in \mathbb{R}^+$ and any $u_h \in \mathscr{X}_h^k$ ensuring $\|u_h\|_{\mathcal{L}^\infty} \leqslant C$, there exists $C' \in \mathbb{R}^+$ which depends only on C and the geometry of $\mathcal{M}_h$, such that:

$$\forall T \in \mathcal{M}_h, \forall i \in T, \quad |\Phi_i^T| \leqslant C' h \sum_{j \in T} |u_i - u_j|.$$

In the case of a $\mathbb{P}^1$ interpolation, if $\overrightarrow{\boldsymbol{\mathcal{F}}_h}$ is also the $\mathbb{P}^1$ projection of the continuous flux $\overrightarrow{\boldsymbol{\mathcal{F}}}$, we can write:

$$\begin{aligned} \Phi_i^{\mathrm{T}} &= \beta_i^{\mathrm{T}} \Phi^{\mathrm{T}} = \beta_i^{\mathrm{T}} \int_{\mathrm{T}} \overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_h) d\mathbf{x} \\ &= \beta_i^{\mathrm{T}} \int_{\partial \mathrm{T}} \overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_h).\vec{\mathbf{n}}\, ds = \beta_i^{\mathrm{T}} \sum_{i \in \mathrm{T}} \frac{1}{2} \overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_i).\vec{\mathbf{n}}_i \\ &= \beta_i^{\mathrm{T}} \sum_{i \neq j} \frac{\left(\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_i) - \overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_j) \right).\vec{\mathbf{n}}_i}{2} \end{aligned}$$

We have here used a convention that is going to be really useful in the rest of the manuscript. In the last equation, $\vec{\mathbf{n}}$ represents the generic outward unit normal to the edges of the triangle, while $\vec{\mathbf{n}}_i$ represents the inward normal to the edge opposite to node i, scaled by the length of this edge. If the distribution coefficients β_i^{T} are uniformly bounded and the approximation of flux $\overrightarrow{\boldsymbol{\mathcal{F}}}$ is regular enough, assumption 4.2 is fulfilled. Unfortunately, this is not as simple for higher order schemes, and we have to verify this hypothesis case by case. In the following, we just assume that assumption 4.2 is always verified.

As an additional hypothesis, we need to define how regular the approximation $\overrightarrow{\boldsymbol{\mathcal{F}}_h}$ of $\overrightarrow{\boldsymbol{\mathcal{F}}}$ must be.

Assumption 4.3

The approximation $\overrightarrow{\boldsymbol{\mathcal{F}}_h}$ of the flux $\overrightarrow{\boldsymbol{\mathcal{F}}}$ verifies:

i) $\overrightarrow{\boldsymbol{\mathcal{F}}}_h$ is a continuous function from $\mathscr{X}_h^k$ into $\mathscr{X}_h^k$,

ii) For any sequence $(u_h)_h$ bounded in $\mathcal{L}^\infty(\mathbb{R}^2)$ independently of h and converging in $\mathcal{L}^2_{loc}(\mathbb{R}^2)$ to u, we have

$$\lim_{h\to 0} \|\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_h) - \overrightarrow{\boldsymbol{\mathcal{F}}}(u)\|_{\mathcal{L}^1_{loc}(\mathbb{R}^2)} = 0.$$

As we have seen above, the $\mathbb{P}^k$ projection of continuous flux $\overrightarrow{\boldsymbol{\mathcal{F}}}$ is usually going to be used for the flux approximation:

$$\overrightarrow{\boldsymbol{\mathcal{F}}_h}(v) = \sum_{i\in\mathcal{M}_h} \overrightarrow{\boldsymbol{\mathcal{F}}}(v_i)\varphi_i^k. \tag{4.17}$$

In this case, the two items of assumption 4.3 are always verified.

In the following theorem we ignore the boundary conditions or just assume they are homogeneous Dirichlet boundary conditions.

Theorem 4.4 (*Lax-Wendroff Like*)

Let $(u_h)_h$ be a sequence of numerical solutions of (4.5) for some given meshes $\mathcal{M}_h$. We assume that the meshes always verify assumption 4.1, and that the scheme satisfies assumptions 4.2 and 4.3. We also assume there exist a constant C depending only on C_1 and C_2 and a function $u \in \mathcal{L}^2(\mathbb{R}^2)$ such that

$$\sup_h \sup_{\mathbf{x}\in\Omega} |u_h(\mathbf{x})| \leqslant C$$

$$\lim_{h\to 0} \|u - u_h\|_{\mathcal{L}^2_{loc}(\mathbb{R}^2)} = 0$$

Then u is a weak solution of (4.1).

Proof: Let Υ be any $\mathcal{C}^1$ function of $\mathbb{R}^2$ with compact support in Ω and Υ_i its value at node i. We also define the Galerkin residual

$$\Psi_i^{\mathrm{T}}(u_h) = \int_{\mathrm{T}} \varphi_i^k \overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_h) d\mathbf{x}, \tag{4.18}$$

where φ_i^k stands for the k^{th} order Lagrangian basis function at node i. Let us take scheme system (4.5), multiply by Υ_i and sum over the degrees of freedom. We obtain:

$$\sum_{i\in\mathcal{M}_h} \Upsilon_i \sum_{\mathrm{T}\in\mathcal{D}_i} \Phi_i^{\mathrm{T}}(u_h) = 0.$$

If we swap the two summation indices, add and remove $\left(\Psi_i^{\mathrm{T}}(u_h)\Upsilon_i\right)$ and use the conservation property

$$\sum_{i\in\mathrm{T}}\left(\Phi_i^{\mathrm{T}}(u_h)-\Psi_i^{\mathrm{T}}(u_h)\right)=\Phi^{\mathrm{T}}-\Phi^{\mathrm{T}}=0,$$

we get, with q being the number of DoFs in each element

$$\underbrace{\frac{1}{q}\sum_{\mathrm{T}\in\mathcal{M}_h}\sum_{i,j\in\mathrm{T}}\left(\Phi_i^{\mathrm{T}}(u_h)-\Psi_i^{\mathrm{T}}(u_h)\right)\left(\Upsilon_i-\Upsilon_j\right)}_{\mathbf{I}}+\underbrace{\sum_{\mathrm{T}\in\mathcal{M}_h}\sum_{i\in\mathrm{T}}\Psi_i^{\mathrm{T}}(u_h)\Upsilon_i}_{\mathbf{II}}\;=\;0. \tag{4.19}$$

We first begin with term **II**:

$$\begin{aligned}
\mathbf{II} &= \sum_{\mathrm{T}\in\mathcal{M}_h}\sum_{i\in\mathrm{T}}\int_{\mathrm{T}}\varphi_i^k(\mathbf{x})\overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_h)\Upsilon_i d\mathbf{x} && \text{(4.20a)}\\
&= \int_\Omega\left(\pi_h^k\Upsilon\right)(\mathbf{x})\overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_h)d\mathbf{x} && \text{(4.20b)}\\
&= -\int_\Omega\overrightarrow{\nabla(\pi_h^k\Upsilon)}.\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_h)d\mathbf{x}+\int_\Omega\overrightarrow{\nabla\Upsilon}.\overrightarrow{\boldsymbol{\mathcal{F}}}(u)d\mathbf{x}\\
&\quad -\int_\Omega\overrightarrow{\nabla\Upsilon}.\overrightarrow{\boldsymbol{\mathcal{F}}}(u)d\mathbf{x} && \text{(4.20c)}\\
&= -\int_\Omega\overrightarrow{\nabla\Upsilon}.\overrightarrow{\boldsymbol{\mathcal{F}}}(u)d\mathbf{x}+\int_\Omega\left(\overrightarrow{\nabla\Upsilon}-\overrightarrow{\nabla(\pi_h^k\Upsilon)}\right).\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_h)d\mathbf{x}\\
&\quad +\int_\Omega\overrightarrow{\nabla\Upsilon}.\left(\overrightarrow{\boldsymbol{\mathcal{F}}}(u)-\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_h)\right)d\mathbf{x} && \text{(4.20d)}\\
&= -\int_\Omega\overrightarrow{\nabla\Upsilon}.\overrightarrow{\boldsymbol{\mathcal{F}}}(u)d\mathbf{x}+o_h(1) && \text{(4.20e)}
\end{aligned}$$

In equation (4.20b), we just use the fact that $\sum_{i\in\mathrm{T}}\Upsilon_i\varphi_i^k$ is the $\mathbb{P}^k$ projection of $\mathcal{C}^1$ test function Υ. In equation (4.20c), we apply the Green formula, enjoying the compact support of Υ and add and remove the second integral. Equation (4.20d) is just a crafty redistribution of the terms, in order to come to the last sought line.

The second integral in (4.20d) is bounded by the $\mathcal{L}^1$ norm of $(\overrightarrow{\nabla\Upsilon}-\overrightarrow{\nabla(\pi_h^k\Upsilon)})$ because the sequence of u_h is bounded in $\mathcal{L}^\infty$ norm and $\overrightarrow{\boldsymbol{\mathcal{F}}_h}$ is a continuous function on $\mathscr{X}_h^k$. And since Υ is a $\mathcal{C}_0^1$ function in Ω,

$$\|\overrightarrow{\nabla\Upsilon}-\overrightarrow{\nabla(\pi_h^k\Upsilon)}\|_{\mathcal{L}^1(\mathbb{R}^2)}=o_h(1).$$

Because Υ is $\mathcal{C}^1$ with compact support in Ω, its gradient is uniformly bounded by a constant independent of h. The third integral in (4.20d)) is then dominated by $\|\overrightarrow{\boldsymbol{\mathcal{F}}}(u)-\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_h)\|_{\mathcal{L}^1(\mathbb{R}^2)}$ which tends to 0 by assumption 4.3(ii), as $\|u_h\|_\infty$ is bounded independently of h, and $u_h\underset{h\to0}{\longrightarrow}u$ in $\mathcal{L}^2_{loc}$.

Let give a look to term **I**. We first obviously have

$$\mathbf{I}\leqslant\frac{1}{q}\sum_{\mathrm{T}\in\mathcal{M}_h}\sum_{i,j\in\mathrm{T}}\left|\Phi_i^{\mathrm{T}}(u_h)-\Psi_i^{\mathrm{T}}(u_h)\right|\left|\Upsilon_i-\Upsilon_j\right| \tag{4.21}$$

and since Υ is $\mathcal{C}_0^1$ in Ω, $|\Upsilon_i-\Upsilon_j|$ is dominated by $h.\sup_\Omega\|\overrightarrow{\nabla\Upsilon}\|=Ch$. Then

$$\mathbf{I}\leqslant\frac{Ch}{q}\sum_{\mathrm{T}\in\mathcal{M}_h}\sum_{i,j\in\mathrm{T}}\left|\Phi_i^{\mathrm{T}}(u_h)-\Psi_i^{\mathrm{T}}(u_h)\right| \tag{4.22}$$

and by assumption 4.2, we obtain

$$\mathbf{I} \leqslant \frac{Ch^2}{q} \sum_{\mathrm{T}\in\mathcal{M}_h} \sum_{i,j\in\mathrm{T}} |u_i - u_j| \tag{4.23}$$

It is now quite a hard work to show this last estimation tends to zero with h. It would be very easy if the u_h were $\mathcal{C}^1$, but it is not the case here. The following lemma proves the last needed limit. Its demonstration can be found in the appendix of [9].

Lemma 4.5

We consider $\Omega \subset \mathbb{R}^2$, a bounded domain, and $(u_h)_h$ a sequence such that $u_h \in \mathscr{X}_h^k$, $\forall h$. We assume there exist a constant C independent of h and $u \in \mathcal{L}^2_{loc}(\Omega)$ such that

$$\sup_h \sup_{\mathbf{x}\in\Omega} |u_h(\mathbf{x})| \leqslant C \quad \text{and} \quad \lim_{h\to 0} \|u - u_h\|_{\mathcal{L}^2(\mathbb{R}^2)} = 0$$

Then

$$\lim_{h\to 0} \left(\sum_{T\in\mathcal{M}_h} |T| \sum_{i,j\in T} |u_i - u_j| \right) = 0$$

The hypothesis of the Lemma are exactly those of Theorem 4.4 which ends to demonstrate that:

$$\sum_{\mathrm{T}\in\mathcal{D}_i} \Phi_i^{\mathrm{T}}(u_h) = 0, \ \forall i \in \mathcal{M}_h, \forall h$$

$$\Rightarrow \quad \int_\Omega \overrightarrow{\nabla\Upsilon}.\overrightarrow{\boldsymbol{\mathcal{F}}}(u) d\mathbf{x} = o_h(1)$$

and u is thus a weak solution of continuous equation (4.1). ■

We have here presented the problem in the steady two dimensional scalar high order case. As we have seen in the beginning of this section, the assumption 4.2 and 4.3 are usually automatically verified by the $\mathcal{RDS}$. The only thing we have to do is to ensure assumption 4.1 which depends only on the meshing.

Vectorial Case: It is in fact possible to prove the same result for unsteady vectorial problems in any space dimension, and that is what is done in the appendix of [9]. We have chosen not to treat the complete demonstration mainly to avoid some really extensive notations and reduce the length of the proof. For the vectorial problems, the only thing to do is to consider the vectorial norm instead of the absolute value. The proof is otherwise similar. This proof can also be very straightforwardly extended to more than two dimensions of space.

Unsteady Case: For the unsteady case, there is a bit more work to do depending on the treatment of the time derivatives. As we observed in Section 3.2.3, there are two ways of treating the unsteady problems. The first one is to consider the unsteady conservation law in space as a steady conservation law in space-time. Then a two dimensional unsteady problem becomes a steady three dimensional one, and this entirely fits the framework used in the theorem demonstration. Equation (4.5) is just expressed into *prismatic elements*, see Figure 3.11. On the other hand, one would like to discretize the time derivative terms by finite differences and then obtain

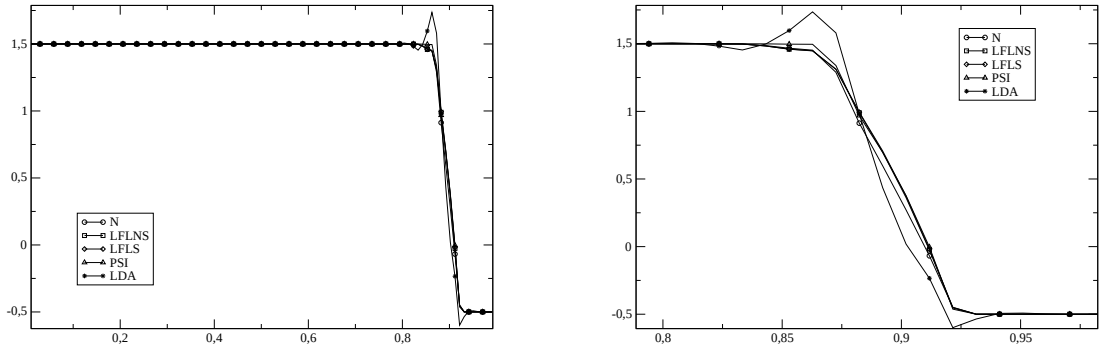


Figure 4.4: Cut through the shock of a Burger solution for different $\mathcal{RD}$ schemes. All the schemes are going to be presented in Section 4.4. The LDA scheme is known to be non positive and we can see that in the two over/under-shoots on both sides of the shock. The exact solution is of course monotone. The right figure is just a zoom of the left one.

a time marching scheme that would solve a two dimensional space problem at each time step. Equation

$$\frac{\partial u}{\partial t} + \vec{\nabla}.\vec{\boldsymbol{\mathcal{F}}}(u) = 0, \ \forall \mathbf{x}, t \in \Omega \times [0;T] \tag{4.24}$$

is approximated by

$$\forall i \in \mathcal{M}_h, \quad \frac{u_i^{n+1} - u_i^n}{\Delta t} + \sum_{\mathrm{T} \in \mathcal{D}_i} \Phi_i^{\mathrm{T}}(u_h^n) = 0. \tag{4.25}$$

The proof of the Lax Wendroff-Like theorem now needs a test function Υ, $\mathcal{C}_0^1$ both in space and time and prove that the term $\frac{u_i^{n+1}-u_i^n}{\Delta t}$ implies

$$-\int_{\Omega \times [0;T]} u \frac{\partial \Upsilon}{\partial t} d\mathbf{x}dt - \int_{\Omega} u_0 \Upsilon(.,0) d\mathbf{x} + o_{(h.dt)}(1). \tag{4.26}$$

For the space dependent term, one has just to handle with integrals in space and time instead of just space sums. More details are given in the appendix of [9].

4.2.2 Maximum Principle and Monotonicity Preserving Condition

As we have already seen in Chapter 2, solutions of conservation laws may lack regularity and even be discontinuous. These discontinuities have always been a source of numerical instabilities since the beginning of numerical computations, partly because the data are mostly represented continuously. If we consider for example a strong shock and allow the solution to *overshoot* or *undershoot* the shock (see Figure 4.4), we are in fact introducing exciting frequencies inside the scheme. And if it is not stable enough, the solution will blow up quickly starting from the region of the shock. One may also control only the stability in a certain norm (let say $\mathcal{L}^2$) but not in another (for example $\mathcal{L}^\infty$). Then during a certain amount of time steps, the distance of the computed solution to the real one could decrease in $\mathcal{L}^2$ norm, but exponentially grow in the $\mathcal{L}^\infty$ one. Such a situation always leads to a numerical blow up.

Even if not speaking about numerical blow up, just allowing theoretically some oscillations in the solution always leads them to appear, because the round off introduces numerical errors that are often amplified. This is observed routinely in the simulations, when using non-monotone schemes. Overshoots and undershoots spoil the solution and sometimes destroy accuracy. That is why we need to define a criterion that will ensure the solution to be smooth, and such that the scheme conserves this property. A way to do this is to enforce the solution to verify a discrete maximum principle. Moreover, as seen in Section 2.1.8, this criterion is intrinsically bound with the entropy condition and has a certain physical meaning. To do so, we first admit that any residual distribution scheme can be recast into the form:

$$\Phi_i^{\mathrm{T}} = \sum_{j \in \mathrm{T}} c_{ij}^{\mathrm{T}} \left(u_i - u_j \right), \tag{4.27}$$

where once more T is not inevitably a triangle. We see in the following that this hypothesis is true for all the $\mathcal{RD}$ schemes developed at this time. As one can see, the value of c_{ii}^{T} can be arbitrary. It is further useful to consider that

$$\forall i \in \mathcal{M}_h, \forall \mathrm{T} \in \mathcal{D}_i, \quad c_{ii}^{\mathrm{T}} = 0$$

As numerical problem (4.5) is non linear, we find in fact its solution u_h as the steady state at infinite time of the pseudo-unsteady problem

$$\frac{\partial u_i}{\partial \tau} + \sum_{\mathrm{T} \in \mathcal{D}_i} \Phi_i^{\mathrm{T}} = 0, \quad \forall i \in \mathcal{M}_h. \tag{4.28}$$

We use the word "pseudo" because the iterative time τ is non real: it is a numerical artifact.

The differential equation is now solved in u_h and τ using a numerical explicit scheme. It is not the only way to get to the steady state but it is the formulation for which the explanations and the definitions are the simplest. The numerical scheme reads

$$\forall i \in \mathcal{M}_h, \quad \frac{u_i^{n+1} - u_i^n}{\Delta \tau} + \omega_i^n \sum_{\mathrm{T} \in \mathcal{D}_i} \Phi_i^{\mathrm{T}}(u_h^n) = 0 \tag{4.29}$$

$$\Leftrightarrow \forall i \in \mathcal{M}_h, \quad \frac{u_i^{n+1} - u_i^n}{\Delta \tau} = -\omega_i^n \sum_{\mathrm{T} \in \mathcal{D}_i} \sum_{j \in \mathrm{T}} c_{ij}^{\mathrm{T}} \left(u_i^n - u_j^n \right) \tag{4.30}$$

$$= -\omega_i^n \sum_{j \in \mathcal{D}_i} \left(\sum_{\mathrm{T} \in \mathcal{D}_i \cap \mathcal{D}_j} c_{ij}^{\mathrm{T}} \right) \left(u_i^n - u_j^n \right),$$

ω_i^n standing here for a local pseud-time stepping parameter that ensures the sought stability through maximum principle, as we will see in the following, see Susection 5.2.1 page 96.

If we denote by $\tilde{c}_{ij}$ the quantity

$$\forall i, j \in \mathcal{M}_h, \quad \tilde{c}_{ij} = \begin{cases} \displaystyle\sum_{\mathrm{T} \in \mathcal{D}_i \cap \mathcal{D}_j} c_{ij}^{\mathrm{T}}, & \text{if } j \in \mathcal{D}_i, j \neq i \\ 0, & \text{else} \end{cases} \tag{4.31}$$

we have the following property:

Property 4.6 (*Local Extremum Decreasing*)

The numerical scheme defined in the previous equation is called Local Extremum Decreasing (LED) if and only if

$$\tilde{c}_{ij} \geqslant 0, \quad \forall i, j \in \mathcal{M}_h. \tag{4.32}$$

Proof: Let us suppose, u_i^n is a local maximum. Then, $u_i^n - u_j^n$ is positive $\forall j \in \mathcal{D}_i$ and the quantity $-\sum_{j\in\mathcal{D}_i} \tilde{c}_{ij}\left(u_i^n - u_j^n\right)$ is negative. At next time step, we will have : $u_i^{n+1} \leqslant u_i^n$.

Exactly like in the *maximum* case, if u_i^n is a local minimum, u_i^{n+1} is obviously going to be greater than u_i^n.

Eventually, if equation (4.32) is not true, it is always possible to build a vector of u_i^n's which local extrema will be increased through this explicit scheme. ■

In fact, the most important sentence in this proof is the last one. Because the *Local Extremum Decreasing* property does not ensure the explicit scheme to be stable, it just describes what is not going to happen. It says that if the solution blows up, it won't come from an increasing of the extrema. The problem of stability is not solved however because this condition does not prohibit another node to become an extremum, or a maximum to become suddenly a senseless minimum. The maximum principle or the $\mathcal{L}^\infty$ stability is still not obtained.

Ensuring condition (4.32) is not easy. That is why we usually ensure a stronger but non necessary condition, much easier to verify : the *Sub-element LED*, also called the *Monotonicity Preserving* condition.

Definition 4.7 (*Monotonicity Preserving Property*)

The above explicit scheme is called Monotonicity Preserving if

$$\forall i \in \mathcal{M}_h, \forall T \in \mathcal{D}_i, \forall j \in T \quad c_{ij}^T \geqslant 0. \tag{4.33}$$

There are two remarks to add to this definition. First, a *Monotonicity Preserving* scheme is obviously *Local Extremum Decreasing*. Second, we are going to see in Section 5.2 that under this new condition, the explicit scheme verifies a discrete maximum principle under a CFL condition. The scheme is then stable in $\mathcal{L}^\infty$ norm. Furthermore, we are also going to describe an implicit method to solve differential system (4.28), and prove condition (4.33) is sufficient to ensure a discrete maximum principle and then stability in $\mathcal{L}^\infty$ norm for the solution obtained by this method. The solution of an implicit monotonicity preserving $\mathcal{RDS}$ is unconditionally stable!

Vectorial Case : Finally, one would like to generalize these results in the case of vectorial problems. In that case, the c_{ij} coefficients become matrices, and one would like to find a criterion similar to (4.32), that would ensure the solution respects some maximum principle. But this is a very hard task as it is complex to define what a local maximum is. A node can absolutely be a local maximum for a variable and at the same time a local minimum for another variable. This still stays as an open question, and we therefore define that for multidimensional problems, the scheme is said to be *monotonicity preserving* when all the c_{ij} are *positive* in the sense

$$\forall M \in \mathcal{M}_n(\mathbb{R}), \quad M \geqslant 0 \Leftrightarrow \left(x^{\mathrm{T}} M x \geqslant 0, \ \forall x \in \mathbb{R}^n\right). \tag{4.34}$$

In fact, this definition has a meaning as it ensures in some way a discrete energy stability, see [2].

4.2.3 Accuracy

As already discussed in section 3.1.2, an important property of a numerical scheme is its accuracy. It is crucial to know how far the computed approximated function u_h is from the weak solution u^* of the continuous problem. In this subsection, we are going to analyze the two dimensional steady scalar problem discretized by means of an approximation at fixed polynomial degree k. The extension to 3D or vectorial problem is straightforward. The following arguments also work for the time dependent case, when using space-time prismatic elements. They just have to be adapted to the situation. If the time derivative terms are treated by finite differences, one could use the following demonstration to analyze the accuracy in space, and then add the study of accuracy in time of the chosen time stepping scheme to get the complete space-time accuracy analysis.

It is impossible to determine $\|u^* - u_h\|$, as u^* is completely unknown. However, the injection of the exact solution into the scheme gives a good estimation of the distance between u_h and u^*. As problem (4.1) is solved through scheme (4.5), one can define the *truncation error vector* $(\xi_i)_{i\in\mathcal{M}_h}$ by

$$\forall i \in \mathcal{M}_h, \quad \xi_i = \sum_{\mathrm{T}\in\mathcal{D}_i} \Phi_i^{\mathrm{T}}(\pi_h^k u^*), \tag{4.35}$$

$\pi_h^k u^*$ being still the $\mathbb{P}^k$ projection of u^*. One could study the norm of this vector. We rather prefer to study the quantity $\Theta(\pi_h^k u^*)$, called the *truncation error*, and defined for any test function $\Upsilon \in \mathcal{C}_0^1(\Omega)$ by:

$$\Theta(\pi_h^k u^*) = \sum_{i\in\mathcal{M}_h} \Upsilon_i \xi_i = \sum_{i\in\mathcal{M}_h} \Upsilon_i \sum_{\mathrm{T}\in\mathcal{D}_i} \Phi_i^{\mathrm{T}}(\pi_h^k u^*). \tag{4.36}$$

Υ_i is of course the value taken by the test function Υ at node i. We give then the following definition:

Definition 4.8 (k^{th} *order accuracy for steady problems*)

A Residual Distribution Scheme is said to be k^{th} order accurate at steady state, if it verifies

$$\Theta(\pi_h^k u^*) = \mathcal{O}(h^k)$$

for any smooth exact solution u^*, with $\Theta(\pi_h^k u^*)$ given by (4.36).

As we did in Section 4.2.1, we need to define the Galerkin residual

$$\Psi_i^{\mathrm{T}}(u_h) = \int_{\mathrm{T}} \varphi_i^k \overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}_h}(u_h) d\mathbf{x},$$

where φ_i^k still stands for the k^{th} order Lagrangian basis function at node i. If we swap the two sums in (4.36), add and remove the Galerkin residual and use the fact that

$$\sum_{i\in\mathrm{T}} \Phi_i^{\mathrm{T}}(u_h) - \Psi_i^{\mathrm{T}}(u_h) = \Phi^{\mathrm{T}} - \Phi^{\mathrm{T}} = 0,$$

we obtain

$$\begin{aligned}\Theta(\pi_h^k u^*) &= \underbrace{\frac{1}{q}\sum_{\mathrm{T}\in\mathcal{M}_h}\sum_{i\in\mathrm{T}}\left(\Phi_i^{\mathrm{T}}(\pi_h^k u^*)-\Psi_i^{\mathrm{T}}(\pi_h^k u^*)\right)(\Upsilon_i-\Upsilon_j)}_{\mathbf{I}} \\ &+ \underbrace{\sum_{\mathrm{T}\in\mathcal{M}_h}\sum_{i\in\mathrm{T}}\Psi_i^{\mathrm{T}}(\pi_h^k u^*)\Upsilon_i}_{\mathbf{II}} \end{aligned} \tag{4.37}$$

We first start with term **II**. Because u^* is the weak solution of (4.1),

$$\int_\Omega \Upsilon_h^k \overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}}(u^*)d\mathbf{x}=0,$$

and

$$\begin{aligned}\mathbf{II} &= \int_\Omega \Upsilon_h^k\left(\overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\pi_h^k u^*)-\overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}}(u^*)\right)d\mathbf{x} \\ &= -\int_\Omega \overrightarrow{\nabla\Upsilon_h^k}.\left(\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\pi_h^k u^*)-\overrightarrow{\boldsymbol{\mathcal{F}}}(u^*)\right)d\mathbf{x}\end{aligned}$$

Now, $(\pi_h^k u^*)$ is a $\mathbb{P}^k$ approximation of u^*, $\overrightarrow{\boldsymbol{\mathcal{F}}}$ is supposed to be continuous and $\overrightarrow{\nabla\Upsilon_h^k}$ is bounded, because $\Upsilon\in\mathcal{C}_0^1(\Omega)$. Then if $\overrightarrow{\boldsymbol{\mathcal{F}}_h}$ is an approximation of flux $\overrightarrow{\boldsymbol{\mathcal{F}}}$ of order $k+1$, we have:

$$\mathbf{II}=\mathcal{O}(h^{k+1}). \tag{4.38}$$

Let us now come to term **I**. The number of degrees of freedom per element is bounded, as k is fixed. The number of triangles in $\mathcal{M}_h$ is of order $\mathcal{O}(h^{-2})$ and because the gradient of Υ is bounded in Ω, $\Upsilon_i-\Upsilon_j=\mathcal{O}(h)$. What gives:

$$\mathbf{I}=\mathcal{O}(h^{-2})\times\mathcal{O}(h)\times\left(\mathcal{O}(\Phi_i^{\mathrm{T}}(\pi_h^k u^*))+\mathcal{O}(\Psi_i^{\mathrm{T}}(\pi_h^k u^*))\right) \tag{4.39}$$

But

$$\begin{aligned}\Psi_i^{\mathrm{T}}(\pi_h^k u^*) &= \int_{\mathrm{T}}\varphi_i^k\overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\pi_h^k u^*)d\mathbf{x} \\ &= \int_{\mathrm{T}}\varphi_i^k\left(\overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\pi_h^k u^*)-\overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}}(u)\right)d\mathbf{x} \\ &= \int_{\partial\mathrm{T}}\varphi_i^k\left(\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\pi_h^k u^*)-\overrightarrow{\boldsymbol{\mathcal{F}}}(u)\right).\vec{\mathbf{n}}\,d\mathbf{x}-\int_{\mathrm{T}}\overrightarrow{\nabla\varphi_i^k}.\left(\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\pi_h^k u^*)-\overrightarrow{\boldsymbol{\mathcal{F}}}(u)\right)d\mathbf{x} \\ &= \mathcal{O}(h^{k+2}).\end{aligned}$$

Then the *truncation error* $\Theta(\pi_h^k u^*)$ is of desired order $k+1$, if $\Phi_i^{\mathrm{T}}(\pi_h^k u^*)$ is of order $k+2$.

We conclude by the following proposition, extended to d dimensions for sake of completeness:

Proposition 4.9 (*High Order Accuracy*)

A Residual Distribution Scheme using $\mathbb{P}^k$ Lagrangian interpolation polynomial is of order

$(k+1)$ if, when u^* is the weak solution of (4.5), the following two conditions are fulfilled:

a) $\overrightarrow{\boldsymbol{\mathcal{F}}_h}$, the flux approximation, is of order $(k+1)$;

b) For a problem in d spatial dimensions, the local nodal residuals verify:

$$\Phi_i^T(\pi_h^k u^*) = \mathcal{O}(h^{k+d}). \tag{4.40}$$

Condition (4.40) guarantees that the scheme has formally a $\mathcal{O}(h^{k+1})$ error. In practice, it is absolutely not sure this convergence rate will be observed, unless some stability constraints are also met. For example, we have proved the Galerkin scheme (that can be easily put into a $\mathcal{RD}$ form) is always of the desired formal order. But it is also well known that this type of scheme is unstable and diverges when the mesh is refined. In this sense, the conditions of Proposition 4.9 are only necessary.

4.2.4 Linearity Preserving Condition

As we have just seen in the previous subsection, reaching $(k+1)^{\text{th}}$ accuracy needs in particular that $\Phi_i^{\mathrm{T}}(\pi_h^k u^*) = \mathcal{O}(h^{k+2})$. What we are going to see here is that this condition is in particular achieved as soon as the distribution coefficients β_i^{T} are bounded independently of h. That is what we call the *Linearity Preserving Condition*.

Let us give a look at the injection of the $\mathbb{P}^k$ projection of an exact smooth solution u^* into the element residual.

$$\begin{aligned}
\Phi^{\mathrm{T}}(\pi_h^k u^*) &= \int_{\mathrm{T}} \overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\pi_h^k u^*) d\mathbf{x} \\
&= \int_{\partial \mathrm{T}} \left(\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\pi_h^k u^*) - \overrightarrow{\boldsymbol{\mathcal{F}}}(u^*) \right) .\vec{\mathbf{n}}\, d\mathbf{x} \\
&= \mathcal{O}(h^{k+2}).
\end{aligned}$$

Then, if the distribution coefficients are bounded independently of h, the $\mathcal{RD}$ scheme reaches the desired order. In that case

$$\Phi_i^{\mathrm{T}}(\pi_h^k u^*) = \beta_i^{\mathrm{T}} \Phi^{\mathrm{T}}(\pi_h^k u^*) = \mathcal{O}(h^{k+2})$$

and

$$\Theta(\pi_h^k u^*) = \mathcal{O}(h^{k+1}).$$

Furthermore, we have seen in Assumption 4.2 that if the distribution coefficients of an $\mathcal{RDS}$ are bounded, the local nodal residuals Φ_i^{T} depend continuously on the values of u_h at nodes $j \in \mathrm{T}$, which is a required condition for Theorem 4.4.

Definition 4.10

A $\mathcal{RD}$ scheme is called Linearity Preserving ($\mathcal{LP}$) if its distribution coefficients β_i^T defined in (4.3) are uniformly bounded independently of h with respect to the solution and the data of the problem:

$$\max_{T \in \mathcal{M}_h} \max_{i \in T} |\beta_i^T| < C < \infty, \quad \forall \Phi^T, u_h, u_h^0, \tau, \ldots \tag{4.41}$$

$\mathcal{LP}$ schemes satisfy by construction the necessary condition for $(k+1)^{th}$ order of accuracy of Proposition 4.9.

We will see further a method recasting automatically a non-$\mathcal{LP}$ scheme into a $\mathcal{LP}$ one. This method will be used to transform any known $\mathcal{RDS}$ of any order of accuracy into a scheme having the maximal order of accuracy.

4.3 Godunov Theorem

Before presenting some classical $\mathcal{RD}$ schemes, and analyze their properties, we wish to present the following theorem that is restricting the panel of possible $\mathcal{RD}$ schemes for high order generalization. This theorem is going to be formulated in the scalar framework. Generalization to vectorial valued problem is assumed. We first begin by the following definition:

Definition 4.11 (*Linear Scheme*)

*A Residual Distribution Scheme of the form (4.30) is said to be **linear** if all the c_{ij} are independent of the numerical solution.*

We recall from the introduction that the goal is here to build a numerical scheme that is *stable* and of the maximal order of *accuracy*. If we consider a $\mathbb{P}^k$ formulation, one wishes then to obtain a scheme that is both $(k+1)^{\text{th}}$ *order accurate* and *monotonicity preserving*. The following theorem claims [50, 76]:

Theorem 4.12 (*Godunov*)

A $\mathbb{P}^k$ Residual Distribution Scheme that is both $(k+1)^{th}$ order accurate (which means $\mathcal{LP}$) and monotonicity preserving cannot be linear.

Proof: This proof is given here because it is valuable for an $\mathcal{RD}$ scheme of any polynomial order of approximation, applied on any type of element with q DoFs. It has been inspired by [114].

Let us consider an $\mathcal{LP}$ linear scheme on an element T having q DoFs. Then the distribution coefficients β_i^{T}, $i \in \mathrm{T}$ as well as the c_{ij} are independent of the solution u. We recall:

$$\Phi_i^{\mathrm{T}} = \beta_i^{\mathrm{T}} \Phi^{\mathrm{T}} = \sum_{j \neq i} c_{ij} \left(u_i - u_j\right). \tag{4.42}$$

Then by summing over $i \in \mathrm{T}$, one obtains:

$$\begin{aligned} \sum_{i \in \mathrm{T}} \Phi_i^{\mathrm{T}} &= \Phi^{\mathrm{T}} \\ &= \sum_{i \in \mathrm{T}} \sum_{j \in \mathrm{T}} \left(c_{ij} - c_{ji}\right) u_i \\ &= \sum_{i \in \mathrm{T}} k_i u_i \end{aligned}$$

where k_i coefficients are also independent of u and moreover verifying

$$\sum_{i \in \mathrm{T}} k_i = \sum_{i,j \in \mathrm{T}} \left(c_{ij} - c_{ji}\right) = 0, \tag{4.43}$$

what allows us to write

$$\Phi^{\mathrm{T}} = \sum_{j \neq i} k_j \left(u_j - u_i\right). \tag{4.44}$$

Then by (4.42), one gets

$$\sum_{j \neq i} c_{ij} \left(u_i - u_j\right) = \sum_{j \neq i} -\beta_i^{\mathrm{T}} k_j \left(u_i - u_j\right), \tag{4.45}$$

and by identification, because all the coefficients of the sums are independent of u,

$$c_{ij} = -\beta_i^{\mathrm{T}} k_j. \tag{4.46}$$

Finally, that means that $\sum_{j \in \mathrm{T}} c_{ij} = 0$ and at least one c_{ij} is negative. This contradict the fact that the scheme is *monotonicity preserving*, see equation (4.33). ■

4.4 Some $\mathcal{RD}$ schemes

We finish this chapter by a review of the different known Residual Distribution Schemes. There exists three different types of them in the literature. They are classified as follows: the four first schemes (N, LDA, Blended and PSI) are called *multidimensional upwind*, the fifth (SUPG) is called *upwind* and could have been presented along with the Finite Volume schemes ($\mathcal{FV}$) and the Lax-Wendroff scheme ($\mathcal{LW}$). Finally, the last presented Lax-Friedrichs (LxF) scheme is known as a *centered* scheme. These three terms in italic are going to be explained in the related subsections.

For each of these schemes we describe its main properties, advantages and drawbacks. We shall also give some remarks on how easily each scheme can be extended to higher order. All of these schemes have first been developed in the scalar framework, but when possible we will also give their generalization to the system case.

4.4.1 Multidimensional Upwind Schemes

Scalar Case : A *multidimensional upwind* scheme is a scheme that respects the directional nature of the advection. Let us consider the two dimensional scalar advection problem

$$\frac{\partial u}{\partial t} + \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla u} = 0, \quad \mathbf{x} \in \Omega \subset \mathbb{R}^2. \tag{4.47}$$

$\vec{\boldsymbol{\lambda}}$ represents at any point the direction of advection. A *multidimensional upwind* scheme is a numerical scheme that distributes all the information downstream, or equivalently that sends no information to the upstream nodes. An illustration is given on Figure 4.5. On this figure, we also define $\vec{\mathbf{n}}_i$ as the inward normal to the opposite edge of node i, scaled by the length of this edge. Then the quantity

$$k_i = \frac{\vec{\boldsymbol{\lambda}}.\vec{\mathbf{n}}_i}{2} \tag{4.48}$$

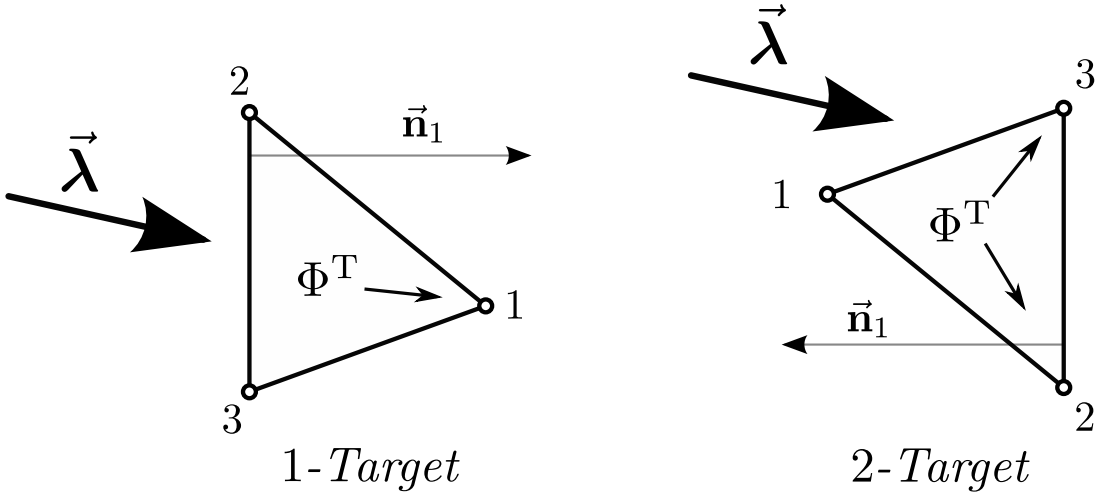


Figure 4.5: Left: 1-*Target* triangle. Node 1 is the only downstream node. It receives the global residual Φ^{T} entirely. Right: 2-*Target* triangle. Node 1 is upstream and receives nothing from the global residual.

tells us if node i is upstream or downstream, depending on its sign. Even though a more general formalism can be developed for a PDS, this geometrical interpretation only applies to the scalar case. In this case, a *multidimensional upwind* scheme is characterized by the following property:

$$\forall \mathrm{T} \in \mathcal{M}_h, \forall i \in \mathrm{T}, \quad k_i \leqslant 0 \Rightarrow \Phi_i^{\mathrm{T}} = 0. \tag{4.49}$$

As one can see on Figure 4.5, there are only 2 possibilities for a $\mathbb{P}^1$ triangle. It could be 1-*Target* as on the left figure. In this case all the *multidimensional upwind* $\mathcal{RD}$ schemes reduce to the same: they all send the totality of the global residual to the unique downstream node. Then $\mathbb{P}^1$ *multidimensional upwind* $\mathcal{RD}$ schemes just differ by the way they distribute the global residual to the downstream nodes in the 2-*Target* triangles (right Figure).

Vectorial Case : In the system case, $\vec{\boldsymbol{\lambda}}$ is a vector of matrices, k_i is thus a $m \times m$ matrix. Because the system is hyperbolic, we have m eigendirections and their associated eigenvalues. The system scheme is now called *multidimensional upwind* if it sends something only on the eigendirections for which the associated eigenvalues are positive. There is no physical stream anymore, as the diagonalization depends on the direction of $\vec{\mathbf{n}}_i$, but numerically, we can consider that in this direction we have m characteristics directed by the m eigenvalues of k_i, and that i should receive no information on the eigendirection for which the characteristic curve is aiming at the opposite side, see Figure 4.6.

Let us introduce some useful notations: in the following, if $\boldsymbol{\Lambda}$ is a diagonal matrix, then $|\boldsymbol{\Lambda}|$ is the diagonal matrix formed by the absolute values of the diagonal elements of $\boldsymbol{\Lambda}$. Now if $\mathbf{K} = \mathcal{R}\boldsymbol{\Lambda}\mathcal{L}$ is a diagonalizable matrix, then

$$|\mathbf{K}| = \mathcal{R}|\boldsymbol{\Lambda}|\mathcal{L}$$

and we now define

$$\mathbf{K}^- = \frac{\mathbf{K} - |\mathbf{K}|}{2}, \quad \text{and} \quad \mathbf{K}^+ = \frac{\mathbf{K} + |\mathbf{K}|}{2}.$$

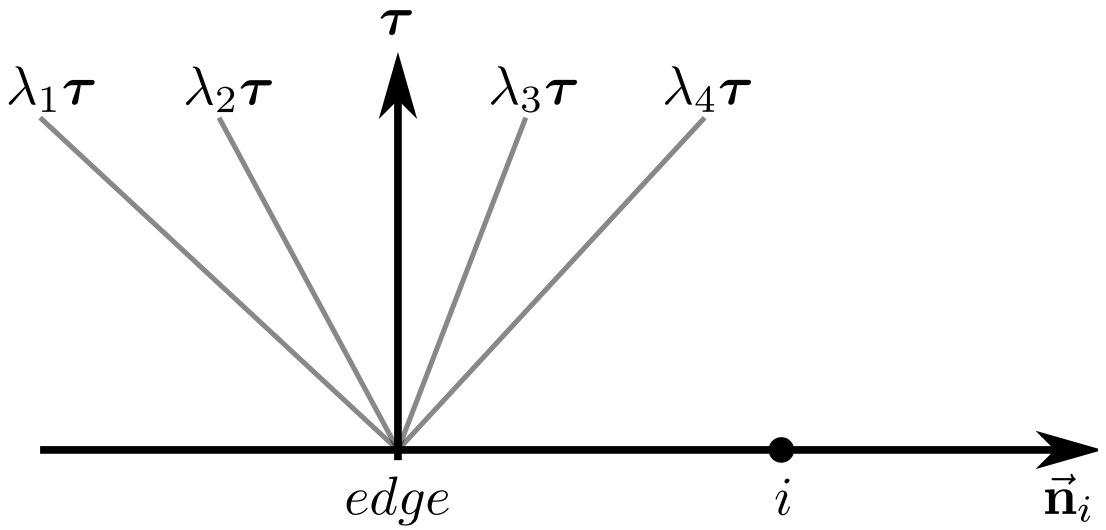


Figure 4.6: **(Multidimensional Upwind)** One dimensional characteristic problem. $\lambda_1, \lambda_2 < 0$, $\lambda_3, \lambda_4 > 0$. Then node i should receive information only on the eigendirections $\vec{r}_3$ and $\vec{r}_4$: $\Phi_i^{\mathrm{T}}.\vec{r}_1 = \Phi_i^{\mathrm{T}}.\vec{r}_2 = 0$.

When the problem is scalar, it is obvious that the absolute value notation coincide with the real absolute value, and

$$k^- = \min(k, 0), \quad k^+ = \max(k, 0).$$

4.4.2 The N-Scheme

The N (Narrow) scheme is a first order scheme, first designed by P.L. Roe ([100, 97], or [89] page 86), very efficient in the case of pure advection equations. It has been since then the basis for the construction of $\mathcal{LP}$ nonlinear positive discretizations (see PSI scheme, Subsection 4.4.5). Moreover, thanks to its *multidimensional upwind* character, it has the lowest numerical dissipation among first-order schemes (see *e.g.* [89] p86). It is defined by the following local nodal residuals:

$$\Phi_i^N = k_i^+(u_i - \tilde{u}), \tag{4.50}$$

where the "average" state $\tilde{u}$ is obtained by recovering the conservation relation. In the $\mathbb{P}^1$ case, this gives

$$\begin{aligned}
\sum_{i\in\mathrm{T}} \Phi_i^N &= \sum_{i\in\mathrm{T}}(k_i^+ u_i) - \tilde{u}(\sum_{i\in\mathrm{T}} k_i^+) \\
&= \Phi^{\mathrm{T}} = \int_{\mathrm{T}} \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla u}\, d\mathbf{x} = \int_{\mathrm{T}} \sum_{i\in\mathrm{T}} u_i \left(\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^1}\right) d\mathbf{x} \\
&= \sum_{i\in\mathrm{T}} k_i u_i.
\end{aligned}$$

And because $k_i = k_i^+ + k_i^-$ and

$$\sum_{i\in\mathrm{T}} k_i = 0 \Rightarrow \sum_{i\in\mathrm{T}} k_i^+ = -\sum_{i\in\mathrm{T}} k_i^-$$

we have:

$$\tilde{u} = \frac{\sum_{i\in \mathrm{T}} k_i^- u_i}{\sum_{i\in \mathrm{T}} k_i^-} \tag{4.51}$$

A big problem of this scheme, is that nothing ensures $\sum_{i\in \mathrm{T}} k_i^-$ to be non null. This appear in particular in the regions where the advection phenomena becomes negligible. For example, the problem is encountered for the Euler equations near stagnation points. These points being isolated, one applies in practice a numerical flux to bypass the problem. Anyway we will use the following notation

$$N = \left(\sum_{i\in \mathrm{T}} k_i^- \right)^{-1}. \tag{4.52}$$

The N scheme is then recast into the form

$$\Phi_i^N = \sum_{j\in \mathrm{T}} k_i^+ N k_j^- (u_i - u_j), \tag{4.50}$$

which shows immediately that the N-Scheme is *monotonicity preserving*. And we have

$$c_{ij}^N = k_i^+ N k_j^- \geqslant 0, \quad \forall i, j \in \mathrm{T}.$$

Finally, there is no way of controlling the bounds of the ratio

$$\beta_i^{\mathrm{T}} = \frac{\Phi_i^{\mathrm{T}}}{\Phi^{\mathrm{T}}},$$

and the N scheme is not $\mathcal{LP}$. The N-Scheme always stays first order accurate, and there is then no need to generalize it to higher order polynomial approximation. All of this will be discussed in Subsection 4.4.5 describing its associated $\mathcal{LP}$ scheme.

Vectorial Case : In the vectorial case, the matrix N is defined easily by equation (4.52) outside the vicinity of the stagnation points, and there is then no difficulty defining the *nodal residuals* by (4.50). Because the sum, product and inversion of matrices conserve the positivity in the sense of (4.34), the vectorial N-Scheme is *monotonicity preserving* but it is still not $\mathcal{LP}$.

4.4.3 The LDA Scheme

The LDA (Low Diffusion A) scheme is a *multidimensional upwind* scheme with bounded distribution coefficients:

$$\Phi_i^{LDA} = \beta_i^{LDA} \Phi^{\mathrm{T}}, \quad \beta_i^{LDA} = -k_i^+ N. \tag{4.53}$$

Because it respects the $\mathcal{LP}$ condition, it is automatically second order. But on the other hand, in can be written as in (4.27) with

$$c_{ij}^{LDA} = -k_i^+ N k_j. \tag{4.54}$$

As one can see, there is no way of determining the sign of the c_{ij}, and the scheme does not verify the *monotonicity preserving* condition. Non physical oscillations appear in the computed

solutions when they show discontinuities. As presented on Figure 4.4 in subsection 4.2.2, the numerical solution *overshoots* or *undershoots* the exact one in the region of the shock. However, it is a very interesting scheme, because it is very little dissipative and gives excellent results on regular enough test cases. This is the reason why this scheme has received a lot of attention in the past decade. The same arguments stay valid in the case of a vectorial problem.

High Order Formulation : Another main drawback of this method is that it is not easy to generalize to $\mathbb{P}^k$ formulation, $k > 1$. Let us keep the example of the scalar advection problem (4.47) to illustrate this. The scheme can easily be extended to 2D $\mathbb{P}^2$ problems, with

$$k_i = \int_{\mathrm{T}} \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^2} d\mathbf{x},$$

φ_i^2 being the $\mathbb{P}^2$ Lagrangian function associated to node i. In that particular case, the scheme is well defined, because $\int_{\mathrm{T}} \overrightarrow{\nabla\varphi_i^2} d\mathbf{x}$ is non null for all the degrees of freedom i. But if we go now to a 3D problem,

$$\varphi_i^2 = \varphi_i^1(2\varphi_i^1 - 1) \Rightarrow \overrightarrow{\nabla\varphi_i^2} = \overrightarrow{\nabla\varphi_i^1}(4\varphi_i^1 - 1), \quad i = 1\dots 3$$

and because $\overrightarrow{\nabla\varphi_i^1}$ is constant over the tetrahedron and $\int_{\mathrm{T}} \varphi_i^1 d\mathbf{x} = \frac{|\mathrm{T}|}{4}$, we have

$$\int_{\mathrm{T}} \overrightarrow{\nabla\varphi_i^2} d\mathbf{x} = 0, \quad i = 1\dots 3. \tag{4.55}$$

Then the values of the solution on the vertices of the tetrahedra do not contribute to the scheme: they can be arbitrary! And we have the same problem if we consider a 2D $\mathbb{P}^3$ problem on triangles. If we look at numbering convention given on Figure 3.7 page 50, because basis function at DoF 10 is symmetric over the triangle, one has:

$$\int_{\mathrm{T}} \overrightarrow{\nabla\varphi_{10}^2} d\mathbf{x} = 0, \tag{4.56}$$

and the value of the solution at the barycentric center of each triangle is useless. In order to bypass this problem, we use today the *sub-triangulation*. Here is the process and its illustration in the case of a 2D $\mathbb{P}^2$ problem.

- Cut the triangle into 4 sub-triangles T_I, T_{II}, T_{III}, T_{IV}, as shown on Figure 4.7;
- For each of sub-triangle T_X, compute a second order global residual

$$\Phi^{\mathrm{T}X} = \sum_{i=1}^{6} u_i \int_{\mathrm{T}_X} \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^2} d\mathbf{x}, \quad X = I, \dots, IV, \tag{4.57}$$

- Compute the first order distribution coefficients in T_X using

$$k_j^{\mathrm{T}X} = \frac{\vec{\boldsymbol{\lambda}}.\overrightarrow{\boldsymbol{n}_j^{\mathrm{T}X}}}{2}, \quad j \in \mathrm{T}_X,\ X = I, \dots, IV, \tag{4.58}$$

- Distribute the global residual

$$\Phi^{\mathrm{T}} = \int_{\mathrm{T}} \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla u_h} d\mathbf{x} = \sum_{X=I}^{IV} \int_{\mathrm{T}_X} \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla u_h} d\mathbf{x}, \tag{4.59}$$

  by sub-triangle, using equation (4.53).

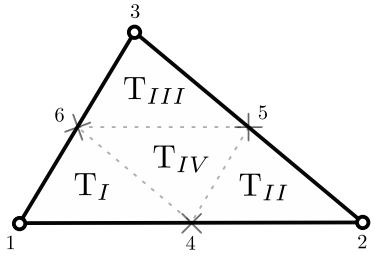


Figure 4.7: Convention of numbering of the $\mathbb{P}^2$ sub-triangles.

Because it uses the first order distribution coefficients by sub-triangles, this method is always defined and takes into account the value of the solution at every degree of freedom. The price to pay is the complexity of the algorithm: instead of computing 1 global residual and distributing it to $\frac{k(k+1)}{2}$ DoFs, one has to interpolate k^2 global residuals on the sub-triangles and distribute each of them to the 3 associated DoFs.

4.4.4 The Blended Scheme

In the last years, there have been many studies trying to create a new class of schemes by blending two types of schemes, one being monotonicity preserving but not $\mathcal{LP}$ (as the N-Scheme), the other one being on the contrary $\mathcal{LP}$ but not monotone (as the LDA-Scheme). One can find good examples of these schemes in [7, 2].

The idea is to define a new scheme by

$$\Phi_i^B = l\Phi_i^N + (1-l)\Phi_i^{LDA}, \tag{4.60}$$

where l of course depends on the solution u_h. Then the challenge is to find the correct criterion defining the blending parameter l, in order to avoid the inconveniences of the schemes one is blending and only keep their advantages. One can also see the blending parameter as a potentiometer that favors the LDA scheme in the regular region and takes advantage of the robustness of the N scheme in the discontinuous areas. Very interesting things have been discovered in this direction, in particular that the PSI scheme (or N-Limited Scheme) we are going to describe in the next paragraph can be seen as an appropriate blending between the N and the LDA schemes (see [2]).

4.4.5 The PSI Scheme

The PSI (Positive Streamline Invariant) scheme of Struijs [113] in certainly the most successful $\mathcal{RD}$ scheme ever designed, for it is *multidimensional upwind*, *conservative*, $\mathcal{LP}$, *monotonicity preserving* and *maximal compact*. It actually comes from the N-scheme, which is why it is often called the *limited N-scheme*. As we have already seen, the N-scheme is *monotonicity preserving* but does not provide bounded distribution coefficients. We then would like to build new distribution coefficients $\beta_i^{\mathrm{T},*}$, $i \in \mathrm{T}$, such that:

- $\sum_{i\in\mathrm{T}} \beta_i^{\mathrm{T},*} = 1$, in order to keep the conservative property;

- $\forall i \in \mathrm{T}$, $\beta_i^{\mathrm{T},*}$ has the same sign as β_i^{T}, in order to conserve the monotonicity preserving property;
- $\exists C \in \mathbb{R}$ such that $\forall \mathrm{T} \in \mathcal{M}_h, \forall i \in \mathrm{T}, \beta_i^{\mathrm{T},*} \leqslant C$, in order the scheme gains the $\mathcal{LP}$ property.

The best way to do that today is to consider, [113]

$$\beta_i^{\mathrm{T},*} = \frac{\left(\beta_i^{\mathrm{T}}\right)^+}{\sum_{j\in\mathrm{T}} \left(\beta_j^{\mathrm{T}}\right)^+}. \tag{4.61}$$

It is called the *limitation technique* and it is the center point of the development of the high order schemes. There are plenty of other limitation techniques and this will be discussed in Section 5.1. But this one is today the best because it is simple to code and always defined, as $\sum_{j\in\mathrm{T}} \left(\beta_j^{\mathrm{T}}\right)^+ \geqslant 1$ when $\Phi^{\mathrm{T}} \neq 0$.

Higher Order Formulation : We have now a scheme that stays compact, and is conservative, $\mathcal{LP}$ and monotonicity preserving. Unfortunately, all this theory is nowadays valid only on the very simple scalar $\mathbb{P}^1$ case. As we said in the paragraph dealing with the N-scheme, there is no way of generalizing directly the N-scheme to $\mathbb{P}^k$ formulation, with $k > 1$. If we consider the direct generalization on the $\mathbb{P}^k$ triangles, we get the same problem expressed in the LDA subsection 4.4.3: some DoFs play no role in the formulation and their value are arbitrary. In order to overcome this problem, the technique consists in formulating the numerical distribution by *sub-triangles*. The practical scheme becomes:

- Compute a second order global residual for each sub-triangle T_X

$$\Phi^{\mathrm{T}_X,2} = \sum_{i=1}^{6} u_i \int_{\mathrm{T}_X} \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^2} d\mathbf{x}, \quad X = I, \ldots, IV,$$

- Compute the first order *upwind* parameters

$$k_j^{\mathrm{T}_X,1} = \frac{\vec{\boldsymbol{\lambda}}.\overrightarrow{\boldsymbol{n}_j^{\mathrm{T}_X}}}{2}, \quad j \in \mathrm{T}_X,\ X = I, \ldots, IV,$$

- Compute the first order distribution coefficients in T_X

$$\begin{aligned}
\Phi^{\mathrm{T}_X,1} &= \sum_{j\in\mathrm{T}_X} k_j^{\mathrm{T}_X,1} u_j, \\
\Phi_i^{\mathrm{T}_X,1} &= \left(k_i^{\mathrm{T}_X,1}\right)^+ (u_i - \tilde{u}_{\mathrm{T}_X}) \ \text{ with } \ \tilde{u}_{\mathrm{T}_X} = \frac{\sum_{j\in\mathrm{T}_X} \left(k_j^{\mathrm{T}_X,1}\right)^- u_j}{\sum_{j\in\mathrm{T}_X} \left(k_j^{\mathrm{T}_X,1}\right)^-} \\
\beta_i^{\mathrm{T}_X,1} &= \begin{cases} \frac{\Phi_i^{\mathrm{T}_X,1}}{\Phi^{\mathrm{T}_X,1}} & \Phi^{\mathrm{T}_X,1} \neq 0 \\ 0 & \text{else} \end{cases}
\end{aligned}$$

- Limit the first order distribution coefficients in T_X

$$\beta_i^{\mathrm{T}_X,*} = \frac{\left(\beta_i^{\mathrm{T}_X,1}\right)^+}{\sum_{j\in\mathrm{T}_X}\left(\beta_j^{\mathrm{T}_X,1}\right)^+}$$

- Distribute the global residuals

$$\Phi^{\mathrm{T}_X,2} = \beta_i^{\mathrm{T}_X,*}\Phi^{\mathrm{T}_X,2}, \quad X = I,\ldots,IV.$$

First this procedure is rather complex, and it is much more difficult to implement than the procedure of generalization of the Lax-Friedrichs scheme to higher order, presented in the following Subsection 4.4.7. The next problem of this algorithm, is that the limited first order distribution coefficients $\beta_i^{\mathrm{T}_X,*}$ are not those of the second order scheme. Therefore, nothing anymore guarantees the scheme to be *monotonicity preserving* and this new PSI scheme has pretty much the same properties as the extended LDA scheme, except it is more complex to deal with. It is nowadays globally agreed that the PSI scheme does not present an easy enough generalization to higher order.

4.4.6 The SUPG Scheme

Let us come to the simply *upwind* schemes. These schemes are not *multidimensional upwind* in the sense they do not verify condition (4.49). But they have an *upwind* character as they take into account the physics of the problem and always give a greater importance to nodes situated downstream. As we have already seen in subsection 4.1.3, the SUPG (Streamline Upwind Petrov Galerkin) scheme can be expressed as an $\mathcal{RD}$ scheme when $\mathbb{P}^1$ formulation is used. The scheme writes:

$$\Phi_i^{SUPG} = \frac{\Phi^{\mathrm{T}}}{3} + \int_{\mathrm{T}} (\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^k})\bar{\boldsymbol{\tau}}(\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla u_h^k})\, d\mathbf{x}, \tag{4.62}$$

which can be seen as a centered homogeneous residual distribution (the Finite Element Galerkin scheme) plus a streamline dissipative term that have of course some *upwind* properties, as explained at the end of the part concerning Petrov-Galerkin formulation in Subsection 4.1.3.

If we give a look to the $\mathbb{P}^1$ case, the matrix $\bar{\boldsymbol{\tau}}$ being defined in subsection 4.1.3, it is classical calculation to determine the distribution coefficients

$$\beta_i^{SUPG} = \frac{1}{3} + \frac{k_i^{\mathrm{T}}\,\bar{\boldsymbol{\tau}}}{|\mathrm{T}|} = \frac{1}{3} + \frac{k_i^{\mathrm{T}}}{\sum_{j\in\mathrm{T}}|k_j^{\mathrm{T}}|}. \tag{4.63}$$

It is then straightforward the β_i^{SUPG} are bounded, and the scheme is $\mathcal{LP}$. But unfortunately, the SUPG in not monotonicity preserving and the scheme provides parasitic oscillations around the regions of discontinuity.

Higher Order Formulation : On the other hand, this scheme is quite easy to generalize to $\mathbb{P}^k$ formulations ($k > 1$) and to three dimensional problems. The only difficulty is to find the right quadrature formula for the dissipative term. This is a point that is discussed further in the manuscript, see Section 5.3 page 103.

4.4.7 The Lax-Friedrichs Scheme

We finally come to the scheme that is going to be used widely in the rest of this thesis. It is called the *Lax-Friedrichs* scheme (LxF) and referred as the *Rusanov* scheme in the literature. It is called a *centered* scheme because it does not give a greater importance to one node or another following some geometrical or physical criteria. Its formulation stays symmetrical relatively to the degrees of freedom of the element. Its convergence is usually slower, because it does not include totally the physics of the problem, and the solution propagates slower in the domain. The main advantage of this scheme is its flexibility and its straightforward generalization to any type of elements (quadrangles, tetrahedra, hexahedra, *aso...*) and any type of discretization ($\mathbb{P}^k$, $\mathbb{Q}^k$, or whatever). As we are going to see, it is also monotone and first order, and can be turned into an $\mathcal{LP}$ scheme easily, using the same technique recasting the first order N-scheme into the $\mathcal{LP}$ PSI scheme. The problem in this case is that when limiting the LxF scheme, the resulting discrete algebraic system may be ill-posed, and the discrete solution of the pseudo time-stepping scheme is not going to converge toward the expected steady solution. We show in the next chapter that this comes from the fact the LxF scheme is totally centered, and that, as in the centered Galerkin case, it needs an additional *upwind* bias to fully converge.

If q denotes again the number of degrees of freedom in the element T, the scheme writes:

$$\Phi_i^{LxF} = \frac{1}{q}\left(\Phi^{\mathrm{T}} + \alpha^{\mathrm{T}} \sum_{j\in \mathrm{T}} (u_i - u_j)\right). \tag{4.64}$$

It is obviously *conservative* and it is *monotonicity preserving* as soon as the scheme parameter α^{T} is large enough. To illustrate this, let us consider the discretization by a $\mathbb{P}^k$ Lagrangian approximation of the steady conservation law in quasi-linear form:

$$\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla u} = 0. \tag{4.65}$$

The unknown u may be scalar or vectorial.

$$\begin{aligned}
\Phi^{\mathrm{T}} &= \int_{\mathrm{T}} \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla u_h} d\mathbf{x} \\
&= \sum_{i\in \mathrm{T}} u_i \left(\int_{\mathrm{T}} \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla \varphi_i^k} d\mathbf{x}\right) \\
&= \sum_{i\in \mathrm{T}} \bar{k}_i^k u_i = -\sum_{j\in \mathrm{T}} \bar{k}_j^k (u_i - u_j).
\end{aligned}$$

Then, we can rewrite the scheme as

$$\Phi_i^{LxF} = \sum_{j\in \mathrm{T}} \frac{\alpha^{\mathrm{T}} - \bar{k}_j^k}{q}(u_i - u_j), \tag{4.66}$$

which is exactly the form of equation (4.27), with

$$c_{ij}^{\mathrm{T}} = \frac{\alpha^{\mathrm{T}} - \bar{k}_j^k}{q}. \tag{4.67}$$

And because $\bar{k}_j^k$ is always diagonalizable, if condition

$$\forall \mathrm{T} \in \mathcal{M}_h, \quad \alpha^{\mathrm{T}} \geqslant \rho\left(\bar{k}_j^k\right), \forall j \in \mathrm{T} \tag{4.68}$$

is met, the scheme is Local Extremum Decreasing, which means monotone when a CFL condition is provided. ρ denotes here the spectral radius in the case of a vectorial problem. If the problem is scalar, one just has to ensure

$$\forall \mathrm{T} \in \mathcal{M}_h, \quad \alpha^{\mathrm{T}} \geqslant \bar{k}_j^k, \forall j \in \mathrm{T} \tag{4.68}$$

Higher Order Scalar Discretization : As one can also see in (4.64), there is absolutely no restriction on q, and the scheme can be applied on any kind of elements. In particular, it works perfectly for higher order discretization. But on the other hand, there is nothing ensuring that the distribution coefficients

$$\beta_i^{\mathrm{T}} = \frac{\Phi_i^{LxF}}{\Phi^{\mathrm{T}}}$$

are bounded. It is well known this scheme is only first order as it is. The *Rusanov* scheme is also very dissipative and this comes from the second term of (4.64). This term tends to diminish everywhere the gradient and thus dissipate very much the solution. One can check that on Figure 5.4 page 106.

However, by limiting this scheme as done for the PSI scheme, one obtains the Limited Lax-Friedrichs scheme (LLxF) that is still compact, very flexible, monotonicity preserving, and this time formally $(k+1)^{\mathrm{th}}$ order accurate. This would be the *ultimate conservative* scheme, if the associated algebraic was not ill-posed. In order to bypass this problem, we are going to add a streamline dissipative term, similar to the one used in the SUPG scheme, and this is one of the main point of the next chapter.

Chapter 5

Construction of a High Order Residual Distribution Scheme

In this chapter, we are going to deal with the general case of a system of conservation laws. As in Chapter 2, m denotes the size of the vector of variables: $\mathbf{U} \in \mathrm{D} \subset \mathbb{R}^m$. The system of conservation laws is usually the Euler system and then $m = d + 2$, where d is the spatial dimension of the problem. We do not allow $\mathbf{U}$ to take any value in $\mathbb{R}^m$ because the physics often add some constraints on the unknowns: the density ρ, the internal energy e, the temperature T, the pressure p, *aso...* must for example stay positive. D represents these constraints.

$$\mathrm{D} = \left\{ \mathbf{U} = (\rho, \rho\vec{\mathbf{u}}, \rho E) \in \mathbb{R}^m ; \rho > 0, E - \frac{|\vec{\mathbf{u}}|^2}{2} > 0 \right\}$$

We are also considering only the steady solution of the PDS and the continuous system writes:

$$\begin{aligned} \text{Find } \mathbf{U} \in \mathrm{D}, \text{ such that} \qquad & \vec{\nabla}.\vec{\boldsymbol{\mathcal{F}}}(\mathbf{U}) = 0, \ \forall \mathbf{x} \in \Omega \\ + \quad & \text{Boundary Conditions.} \end{aligned} \tag{5.1}$$

This chapter mainly focuses on the Lax-Friedrichs scheme presented in Subsection 4.4.7. This is the scheme that has been used in most of the calculations carried out during this thesis. As we have seen in the previous paragraph, the first order LxF scheme, first designed for $\mathbb{P}^1$ triangles, can be easily generalized to higher order polynomial representation in any kind of polyhedral cell. Along the following section, we explain step by step how the steady solution of (5.1) is obtained with this high order scheme. The theory is mainly developed on $\mathbb{P}^2$ triangles, but details could be given for even higher representation of the data in triangles or for $\mathbb{Q}^k$ approximation. In most of cases, the generalization is straightforward. The first section deals with the details of computation of the total and nodal residuals already theoretically seen in Subsection 4.1.1. More details are given about the limitation technique recasting any $\mathcal{RD}$ scheme into an $\mathcal{LP}$ one. In a second section, we speak about the practical resolution of the non linear problem obtained in Section 5.1. We examine the several choices we have to reach the steady state solution of the problem. A third chapter is going to present the main drawback of the LxF method and the way we nowadays get around it. The limited LxF scheme often leads to an ill-posed linear problem that prevents the solution to converge. This problem is cured with an additional stabilization term and we here explain its inconveniences and how we evaluate it numerically. In a last section,

we present the main boundary conditions we need for the simulations of Euler or Navier-Stokes problem, and we detail their practical implementation. Finally this chapter ends by a short summary of the main points of the high order $\mathcal{RDS}$ implementation.

5.1 Total and Nodal Residual - Limitation

5.1.1 Global Residual

The scheme first starts with the evaluation of the *Global Residual* or *Element Residual*, which is given by

$$\Phi^{\mathrm{T}} = \int_{\mathrm{T}} \mathrm{div}\left(\overrightarrow{\boldsymbol{\mathcal{F}}}_h(\mathbf{U}_h)\right) d\mathbf{x} \tag{5.2a}$$

$$= \int_{\partial \mathrm{T}} \overrightarrow{\boldsymbol{\mathcal{F}}}_h(\mathbf{U}_h).\vec{\mathbf{n}}\, ds. \tag{5.2b}$$

As remarked in the preamble, T has not to be a triangle, and this is valid for any kind of numerical approximation. Now, the Lax-Wendroff Theorem of subsection 4.2.1 and Proposition 4.9 enforce conditions on the flux approximation. These conditions are met when approximating the exact flux by its k^{th} order Lagrangian projection

$$\overrightarrow{\boldsymbol{\mathcal{F}}}_h(\mathbf{U}) = \sum_{i \in \mathcal{M}_h} \overrightarrow{\boldsymbol{\mathcal{F}}}_i \varphi_i^k, \tag{5.3}$$

where

$$\overrightarrow{\boldsymbol{\mathcal{F}}}_i = \overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U}_i) = \overrightarrow{\boldsymbol{\mathcal{F}}}\left(\mathbf{U}(\mathbf{x}_i)\right).$$

Then, the approximated flux $\overrightarrow{\boldsymbol{\mathcal{F}}}_h$ is a k^{th} order polynomial over the edges and by construction, see section 3.2, we have the exact number of degrees of freedom on the edges to represent uniquely this polynomial. Formulation (5.2) is thus totally suitable to compute the *Element Residual* by

$$\Phi^{\mathrm{T}} = \sum_{\mathrm{edge} \in \partial \mathrm{T}} \left(\sum_{i \in \mathrm{edge}} \frac{\overrightarrow{\boldsymbol{\mathcal{F}}}_i}{\|\vec{\mathbf{n}}_{\mathrm{edge}}\|} \int_{\mathrm{edge}} \varphi_i^k ds \right) .\vec{\mathbf{n}}_{\mathrm{edge}}, \tag{5.4}$$

which is just a linear combination of the values taken by $\overrightarrow{\boldsymbol{\mathcal{F}}}$ at the DoFs of T, with coefficients $\frac{1}{\|\vec{\mathbf{n}}_{\mathrm{edge}}\|} \int_{\mathrm{edge}} \varphi_i^k ds$. These integrals are simple to evaluate and their values are identical for every triangle. They can be precomputed. Hereafter we report the exact quadrature of the *Global Residual* for $k = 1 \ldots 3$ in a triangle, the numbering being defined on Figure 3.7 page 50, and $\vec{\mathbf{n}}_i$ being the inward normal to the opposite edge of i when it is a vertex of T, or the outward normal to the edge i is belonging to when it is an extra DoF.

- $\mathbb{P}^1$:

$$\Phi^{\mathrm{T}} = \sum_{i=1}^{3} \frac{\overrightarrow{\boldsymbol{\mathcal{F}}}_i.\vec{\mathbf{n}}_i}{2} \tag{5.5}$$

- $\mathbb{P}^2$:

$$\Phi^{\mathrm{T}} = \sum_{i=1}^{3} \frac{\overrightarrow{\boldsymbol{\mathcal{F}}}_i.\vec{\mathbf{n}}_i}{6} + \sum_{i=4}^{6} \frac{2}{3} \overrightarrow{\boldsymbol{\mathcal{F}}}_i.\vec{\mathbf{n}}_i \tag{5.6}$$

- $\mathbb{P}^3$:

$$\Phi^{\mathrm{T}} = \sum_{i=1}^{3} \frac{\vec{\boldsymbol{\mathcal{F}}}_i.\vec{\mathbf{n}}_i}{8} + \sum_{i=4}^{9} \frac{3}{8}\vec{\boldsymbol{\mathcal{F}}}_i.\vec{\mathbf{n}}_i \tag{5.7}$$

All of this is obviously true in the case of quadrangles. The extensions of these interpolations to any kind of configuration is obvious. As one can notice, for $\mathbb{P}^3$ triangle, the value of the *global residual* does not depend of the value of $\vec{\boldsymbol{\mathcal{F}}}_{10}$. This is however not really a problem as node 10 will still play a role in the LxF *nodal residual* and receive a part of the *global residual* after limitation. This remark is general for all the extra DoFs that are situated inside the elements.

5.1.2 Local Nodal Residual

Now we have computed the *global residual*, we wish to distribute it to the nodes via the first order Lax-Friedrichs *nodal residuals*. In fact, these signals are only used to build the higher order Limited Lax-Friedrichs scheme. We recall first order LxF *nodal residual* for q degrees of freedom

$$\Phi_i^{LxF} = \frac{1}{q}\left(\Phi^{\mathrm{T}} + \alpha^{\mathrm{T}} \sum_{j\in \mathrm{T}} (\mathbf{U}_i - \mathbf{U}_j)\right), \tag{5.8}$$

which is obviously *conservative*. The big deal here is to compute well the parameter α^{T}. As we have seen in Subsection 4.4.7, α^{T} ensures monotonicity preserving condition when it is large enough. But on the other hand, if it is too large, the centered term $\frac{\Phi^{\mathrm{T}}}{q}$ will become insignificant compared to the second term $\sum_{j\in \mathrm{T}}(\mathbf{U}_i - \mathbf{U}_j)$, related to the local gradient of the solution. The larger α^{T} is, the less related to the physics of the problem the scheme is. One wishes then to find the finest criterion to define α^{T}. As we have seen in Subsection 4.4.7, a necessary condition is

$$\alpha^{\mathrm{T}} \geqslant \rho\left(\bar{k}_j^k\right), \quad \forall j \in \mathrm{T}. \tag{5.9}$$

Fortunately, the eigenvalues of the k_i matrices are known in the case of the Euler System (see Subsection 2.2.9) and this condition is recast into:

$$\alpha^{\mathrm{T}} = \max_{i\in \mathrm{T}} \left(\|\vec{\mathbf{u}}_i\| + c_i\right) . \max_{\mathrm{edge}} |\mathrm{edge}| \tag{5.10}$$

where c_i denotes the speed of the sound at point i.

5.1.3 Limitation Techniques

Finally, the LxF scheme is only first order and we wish to obtain a higher order one. Which means we need to get at least the $\mathcal{LP}$ condition. In 4.4.5, we have already presented a procedure turning the first order N scheme into the impressive high order PSI scheme. We first begin by adapting this algorithm to the case of the vectorial LxF scheme and then discuss other possibilities of *limitations*.

Scalar Case : In the scalar case, we begin by defining the first order *Distribution Coefficients*:

$$\beta_i^{\mathrm{T}} = \begin{cases} \frac{\Phi_i^{LxF}}{\Phi^{\mathrm{T}}}, & \text{if } \Phi^{\mathrm{T}} \neq 0, \\ 0, & \text{else.} \end{cases} \tag{5.11}$$

and use the limitation technique already presented in (4.61) to get the $(k+1)^{\text{th}}$ order *Distribution Coefficients*:

$$\beta_i^* = \frac{\left(\beta_i^{\mathrm{T}}\right)^+}{\sum_{j\in\mathrm{T}}\left(\beta_j^{\mathrm{T}}\right)^+}. \tag{5.12}$$

We recall that this formula is always defined as $\sum_{j\in\mathrm{T}}\left(\beta_j^{\mathrm{T}}\right)^+ \geqslant 1$ if $\Phi^{\mathrm{T}} \neq 0$. Using this procedure, the new limited scheme with β_i^* distribution coefficients has the following properties:

- The scheme is **conservative**
$$\sum_{i\in\mathrm{T}} \beta_i^* = 1. \tag{5.13}$$

- The scheme is **linearity preserving**. β_i^* is always defined because $\sum_{j\in\mathrm{T}}\left(\beta_j^{\mathrm{T}}\right)^+ \geqslant 1$ when $\Phi^{\mathrm{T}} \neq 0$ and:
$$0 \leqslant \beta_i^* \leqslant 1. \tag{5.14}$$

- If the first order scheme is **monotonicity preserving** then the $(k+1)^{\text{th}}$ order one is as well because
$$\forall i \in \mathrm{T}, \quad \beta_i^*.\beta_i^{\mathrm{T}} \geqslant 0. \tag{5.15}$$
If one has
$$\Phi_i^{LxF} = \beta_i^{\mathrm{T}}\Phi^{\mathrm{T}} = \sum_{j\in\mathrm{T}} c_{ij}(u_i - u_j),$$
with positive c_{ij} coefficients, one obtains
$$\Phi_i^* = \beta_i^*\Phi^{\mathrm{T}} = \sum_{j\in\mathrm{T}} \frac{\beta_i^*}{\beta_i^{\mathrm{T}}} c_{ij}(u_i - u_j),$$
where $\frac{\beta_i^*}{\beta_i^{\mathrm{T}}} c_{ij} \geqslant 0, \quad \forall i,j \in \mathrm{T}$.

System Case : As soon as the residuals are multidimensional, the *Distribution Coefficients* become matrices, and the procedure is much more complex. Of course, one could limit the residual line by line (or equivalently one unknown after another) and this works quite well (see [8, 92]). The main advantage of this choice is to be able to maintain some constraints directly on the variables, for example positivity for the density. But in the case of the Euler equations, it works actually much better to limit the *characteristic variables* ([10] page 106). To do so, we first project the *nodal residuals* on the left eigenvectors $\mathcal{L}_i$ of the hyperbolic problem (5.1), evaluated using the average state:
$$\bar{\mathbf{U}} = \frac{1}{q}\sum_{i\in\mathrm{T}} \mathbf{U}_i,$$
and in the direction tangential to the stream $\vec{\mathbf{n}}_{\vec{\mathbf{u}}} = \frac{\vec{\mathbf{u}}}{\|\vec{\mathbf{u}}\|}$. $\vec{\mathbf{u}}$ denotes here the mean velocity in the triangle *ie.* the velocity vector associated to $\bar{\mathbf{U}}$. The left eigenvectors are defined in Subsection 2.2.9. The q projected residuals for a given linear form $\mathcal{L}_i\left(\Phi_j^{LxF}\right)$ are then limited using scalar formula (5.12), with
$$\beta_{ij} = \frac{\mathcal{L}_i\left(\Phi_j^{LxF}\right)}{\sum_{j\in\mathrm{T}}\mathcal{L}_i\left(\Phi_j^{LxF}\right)} = \frac{\mathcal{L}_i\left(\Phi_j^{LxF}\right)}{\mathcal{L}_i\left(\Phi^{\mathrm{T}}\right)}.$$

This gives q limited coefficients x_{ij}, $j = 1 \dots q$. The limited vector Φ_j^* is then reconstructed as the vector having coordinates $(x_{ij})_{i=1\dots m}$ in the basis of the m right eigenvectors $\mathcal{R}_i$, duals of the $\mathcal{L}_i$s. This last paragraph dealing with the limitation of multidimensional $\mathcal{RD}$ scheme is summarized in algorithm 1.

Algorithm 1 Vectorial Limitation

for $i = 0$ to m **do**
  for all $j \in \mathrm{T}$ **do**
    $\beta_{ij} \longleftarrow \frac{\mathcal{L}_i\left(\Phi_j^{LxF}\right)}{\mathcal{L}_i(\Phi^{\mathrm{T}})}$
    $x_{ij} \longleftarrow \frac{(\beta_{ij})^+}{\sum_{j\in\mathrm{T}} (\beta_{ij})^+}$
  end for
end for
for all $j \in \mathrm{T}$ **do**
  $\Phi_j^* \longleftarrow \sum_{i=1}^{m} x_{ij}\mathcal{R}_i$
end for

Geometrical Representation in the Scalar 2D $\mathbb{P}^1$ Case : Ideally, one would like the limitation also takes into account the *Upwind* property. This would provide a stable $(k+1)^{\text{th}}$ order scheme, a perfect scheme. There exists such a limitation technique in the scalar 2D $\mathbb{P}^1$ case and we need a geometrical representation to illustrate it, see Figure 5.1. On the left part of the figure is represented the Struijs limitation (5.12) for $\mathbb{P}^1$ triangles. In the scalar case, the three distribution coefficients β_i^{T} define a unique point B in $\mathbb{R}^2$ by its barycentric coordinates in T. For the Struijs limitation, there are three main regions for B. B can be first situated inside the triangle (zone 1). In that case, all the β_i^{T} are positive and smaller than 1, and if we denote B^* the image of B by the limitation process, one has: $B^* = B$. B can also be in zone 2,3 or 4. In that case, one β_i^{T} is positive and the to other are negative. Then $\beta_i^* = 1$ and $\beta_j^* = 0, \forall j \neq i$. B^* is limited toward the closest vertex to B. Finally, the most complex situation is when B is in zone 5,6 or 7. In that case, one β_i^{T} is negative and the two other are positive. Then, the limitation provides $\beta_i^* = 0$ and B^* is situated on the edge opposite to node i. Furthermore, Struijs limitation technique conserves the ratio between the two strictly positive distribution coefficients:

$$\frac{\beta_j^*}{\beta_k^*} = \frac{\beta_j^{\mathrm{T}}}{\beta_k^{\mathrm{T}}}.$$

As shown on the left on Figure 5.1, B is limited along the straight line joining B and node i and B^* is then situated at the intersection between this straight line and the edge opposite to i. Unfortunately, nothing ensures the new distribution point B^* to be downstream. In the case of Figure 5.1 for example, it is thoroughly possible B stays in region 4, as point B_1. B^* is then node 3 which is the upstream node, and this is exactly the opposite situation of the *Upwind* property (4.49).

An Upwind Limitation : If we want to turn the scheme into an upwind scheme, the limitation technique has to depend somewhere of $\vec{\boldsymbol{\lambda}}$, the direction of advection. One possibility

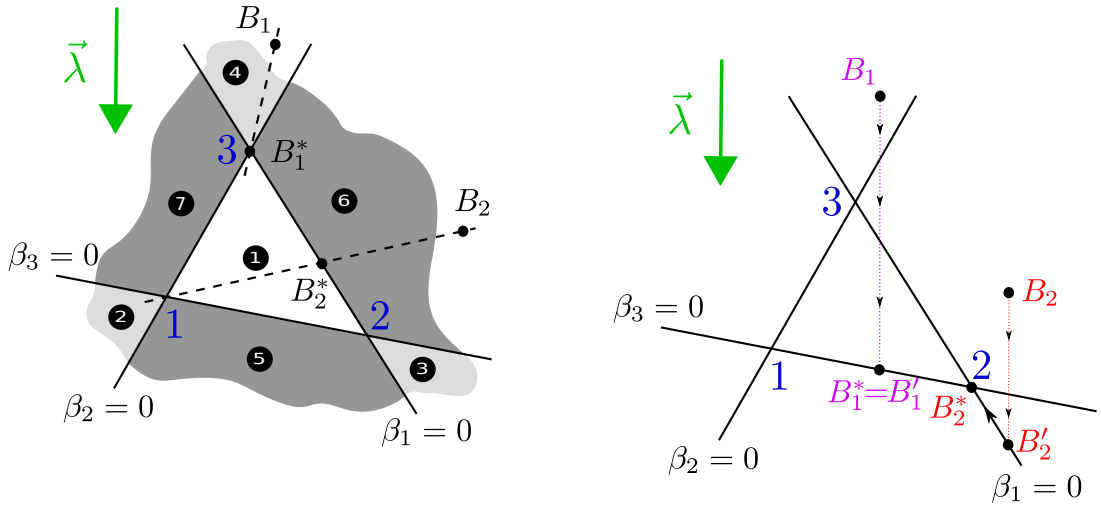


Figure 5.1: Geometrical interpretation of the limitation technique. Point B has barycentric coordinates $\left(\beta_1^{\mathrm{T}}, \beta_2^{\mathrm{T}}, \beta_3^{\mathrm{T}}\right)$. The geometrical transformation $B \mapsto B^*$ depends of the area in which lies B. On the right is presented the classical Struijs limitation technique while on the right figure we illustrate a try for an upwind limitation.

is the following: in the scalar $\mathbb{P}^1$ case, if one considers the unique line defined by B and direction vector $\vec{\boldsymbol{\lambda}}$, it crosses the straight lines defined by the edges of T at 2 or 3 points. If the advection speed is parallel to one edge, we consider that the intersection point is situated at $-\infty$. We then define B' as the one of the three intersection points that is situated the farther downstream from B. If all the intersection points are situated upstream with respect to B, we set $B' = B$. Then B^* is obtained as the Struijs limitation of the barycentric point B'. This is shown on the right part of Figure 5.1. This gives a very efficient scheme in $\mathbb{P}^1$ and for a two dimensional domains. The iterative convergence is as fast as for a classical upwind scheme (N Scheme, LDA scheme) and the result is good whereas no stabilization have been used. To assess this we have computed a very simple pure advective problem on the unit square $[0;1]^2$ for constant vertical advection $\vec{\boldsymbol{\lambda}} = (0,1)$:

$$\begin{cases} \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla u} = 0 \\ u(x,0) = \sin^2(5\pi x) \\ u(0,y) = u(1,y) = 0 \end{cases} \tag{5.16}$$

The upper boundary is let free. We have run this second order test case on 5 different triangular grids having 10, 20, 40, 80 and 100 nodes on each boundary respectively. On Figure 5.2, we have represented above the isolines of the solutions on the finest grid and the iterative convergence. The solution is nice and the iterative convergence is fast. Below is presented the grid convergence. The slope is indeed only 1.45. But if we compare these results with the ones that will be presented in Subsection 6.2.1, we see they are everywhere better. The result is clear: the *upwind* limitation is much faster and gives better results. Moreover, this new limitation technique does not fulfill condition (5.15), because the barycentric point B is allowed to change zone (for example from zone (4) to zone (5) for point B_1 on the right part of Figure 5.1). Then, it should not be monotonicity preserving anymore. But in practice, we observe that the solution is smooth and stays bounded between its initial extremal values.

Unfortunately, its generalization to other cases that $\mathbb{P}^1$ scalar problems is not easy at all. We

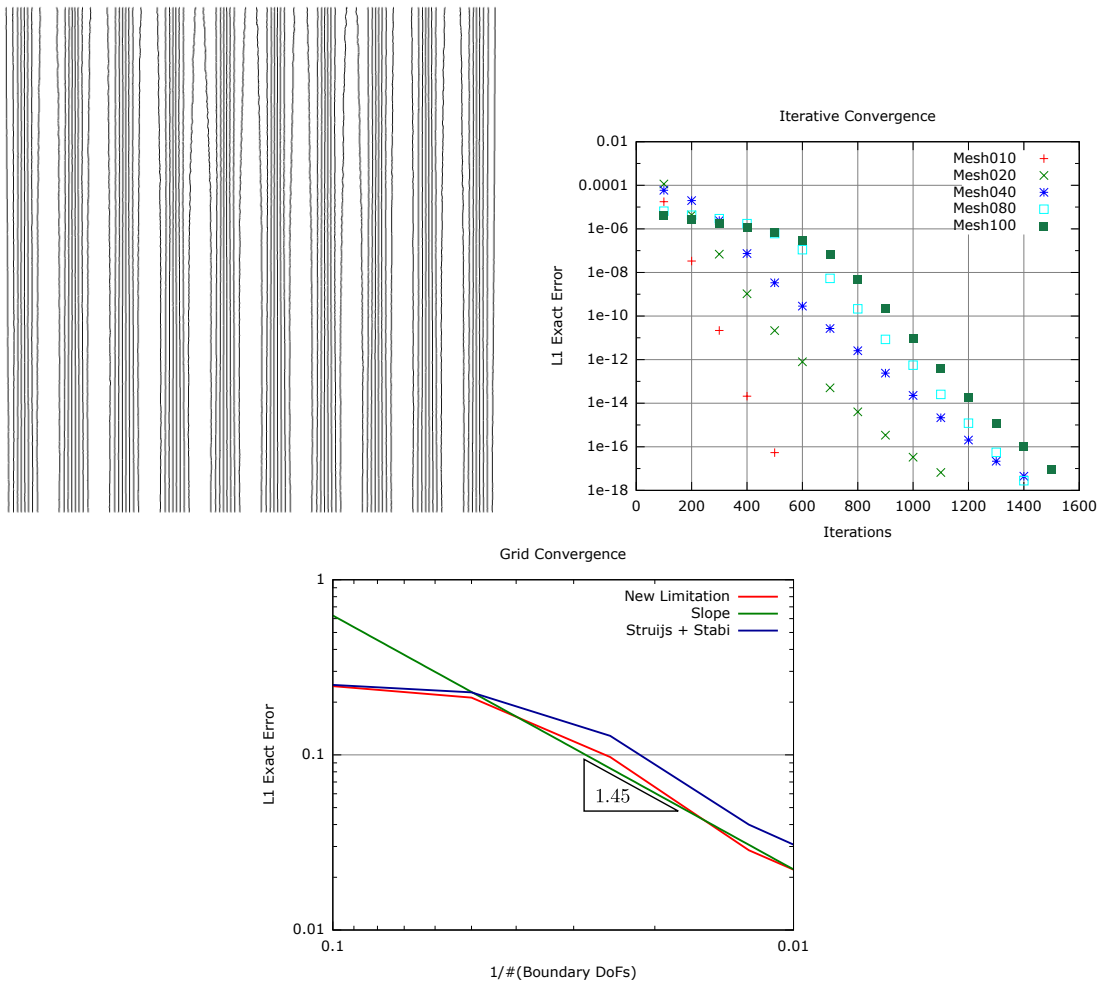


Figure 5.2: $\mathbb{P}^1$ results for scalar problem (5.16) obtained with LxF scheme limited by the limitation technique illustrated on the right part of Figure 5.1. Above are given the isolines of the solution on the finest grid as well as its iterative convergence. Below is shown a comparison in term of grid convergence between this new scheme and the classical one that is going to be detailed next.

have tried to apply it to vectorial problems, but it has not given any interesting results at that moment. The difficulty being to handle with the distribution coefficients β_i^{T} that are now matrices of size m. Furthermore, it is absolutely not possible to define the barycentric point B when using a higher order polynomial representation, because there are $\frac{k(k+1)}{2}$ distribution coefficients. Some more interest should be given to this limitation technique as it is very promising.

Other Limitation Techniques : During this thesis, we have been trying many other limitation possibilities. Another technique would be to allow some β_i^{T} to be negative, while the $\beta_j^T,\ j \in \mathcal{M}_h$ stay globally bounded. The main trend is the lower the negative bound on the β_i^{T} is, the more dissipative the limited scheme becomes. If the bound is to low, the $\beta_j^T,\ j \in \mathcal{M}_h$ stay as they are and the scheme is so dissipative that it becomes first order. This direction of research is really exciting but has not given any interesting result so far, and the best limitation technique still remains the Struijs one.

5.2 Solution of the Algebraic Equation

As we have already seen several times, the steady state vectorial solution $\mathbf{U}_h$ verifies the non linear equation

$$\sum_{\mathrm{T}\in\mathcal{D}_i} \Phi_i^{\mathrm{T}}(\mathbf{U}_h) = 0,\ \forall i \in \mathcal{M}_h. \tag{5.17}$$

This section aims at explaining the different options we have to solve this problem. In fact, all the solutions come from the same common idea. As seen in Subsection 4.2.2, $\mathbf{U}_h$ is seen as the steady numerical state of the pseudo-unsteady problem

$$|\mathcal{D}_i|\frac{\partial \mathbf{U}_i}{\partial \tau} + \sum_{\mathrm{T}\in\mathcal{D}_i} \Phi_i^{\mathrm{T}}(\mathbf{U}_h) = 0, \quad \forall i \in \mathcal{M}_h, \tag{5.18}$$

where $|\mathcal{D}_i|$ is only here to make the equation dimensionally correct and τ is a pseudo time used to reach the steady state of (5.18), which is obviously the solution of (5.17). We then discretize this continuous problem by finite differences and obtain the pseudo-time stepping numerical scheme:

$$|\mathcal{D}_i|\frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta\tau} + \sum_{\mathrm{T}\in\mathcal{D}_i} \Phi_i^{\mathrm{T}}(\mathbf{U}_h^\chi) = 0, \quad \forall i \in \mathcal{M}_h. \tag{5.19}$$

The parameter χ represents the time step at which the residual is estimated. We have two cases:

- $\chi = n$: new solution at time $n+1$ can be computed explicitly. That is why this scheme is called the *explicit scheme*,
- $\chi = n+1$: new solution at time $n+1$ cannot be computed directly. Its value is the implicit solution of an non linear equation. That is why this scheme is called the *implicit scheme*.

5.2.1 The Explicit Scheme

The solution at time $\tau = n+1$ is updated via the formula

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \omega_i^n \sum_{\mathrm{T}\in\mathcal{D}_i} \Phi_i^{\mathrm{T}}(\mathbf{U}_h^n), \quad \forall i \in \mathcal{M}_h, \tag{5.20}$$

with ω_i^n being a pseudo time stepping parameter which dimension is

$$[\omega_i^n] = \frac{\text{time}}{\text{area}}.$$

This parameter is useful to ensure the $\mathcal{L}^\infty$ stability of the scheme, as we will now see.

Scalar Case : If one uses formulation (4.27) on page 72, one has:

$$\forall i \in \mathcal{M}_h, \quad u_i^{n+1} = \left(1 - \omega_i^n \sum_{j \in \mathcal{D}_i} \tilde{c}_{ij}\right) u_i^n + \omega_i^n \sum_{j \in \mathcal{D}_i} \tilde{c}_{ij} u_j^n, \tag{5.21}$$

$\tilde{c}_{ij}$ being defined like in (4.31), page 72 as:

$$\tilde{c}_{ij} = \sum_{\mathrm{T} \in \mathcal{D}_i \cap \mathcal{D}_j} \gamma_i^{\mathrm{T}} c_{ij}^{\mathrm{T}}, \tag{5.22}$$

with c_{ij}^{T} coming from the first order scheme and $\gamma_i^{\mathrm{T}} = \frac{\beta_i^*}{\beta_i^{\mathrm{T}}} \geqslant 0$ when $\beta_i^{\mathrm{T}} \neq 0$ or $\gamma_i^{\mathrm{T}} = 0$ else, representing the limitation process. Because equation (5.10) ensures all $\tilde{c}_{ij}$ to be positive and the sum of the barycentric coefficients being 1, u_i^{n+1} is a mean value of the $\left(u_j^n\right)_{j \in \mathcal{D}_i}$ if and only if

$$0 \leqslant \omega_i^n \leqslant \frac{1}{\sum_{j \in \mathcal{D}_i} \tilde{c}_{ij}}. \tag{5.23}$$

It is then sure

$$\forall i \in \mathcal{M}_h, \quad \min_{j \in \mathcal{M}_h} u_j^n \leqslant u_i^{n+1} \leqslant \max_{j \in \mathcal{M}_h} u_j^n,$$

and therefore

$$\forall n \in \mathbb{N}, \forall i \in \mathcal{M}_h, \quad \inf_{\mathbf{x} \in \Omega} u_0(\mathbf{x}) \leqslant u_i^n \leqslant \sup_{\mathbf{x} \in \Omega} u_0(\mathbf{x}),$$

which is the $\mathcal{L}^\infty$ stability of the numerical solution.

In practice, it is complex and not needed to compute the $\tilde{c}_{ij}$ though, because we have a stronger but non necessary criterion that ensures $\mathcal{L}^\infty$ stability. As seen in (4.67), page 86, for the LxF scheme the first order monotonicity coefficients verify $\sum_{j \in \mathrm{T}} c_{ij} = \alpha^{\mathrm{T}}$ and because $0 \leqslant \gamma_i^{\mathrm{T}} \leqslant 1$,

$$\frac{1}{\sum_{j \in \mathcal{D}_i} \tilde{c}_{ij}} \geqslant \frac{1}{\sum_{\mathrm{T} \in \mathcal{D}_i} \alpha^{\mathrm{T}}} \geqslant 0.$$

Then a good and easy estimation of the pseudo time stepping parameter ω_i^n to ensure the monotonicity of the scheme is

$$\omega_i^n = \frac{1}{\sum_{\mathrm{T} \in \mathcal{D}_i} \alpha^{\mathrm{T}}}. \tag{5.24}$$

System Case : Unfortunately, the same reasoning cannot be done in the system case, because the $\tilde{c}_{ij}$ are now matrices. We then keep the stability criterion (5.24) and use it as it is in the mutidimensional problem because α^{T} are scalar quantities. In practice, the explicit LxF scheme applied to a vectorial problem has always given stable results so far.

Advantages and Drawbacks of the Explicit Formulation : The main advantages of the explicit method are that it is very robust and easy to implement. As soon as condition (5.23) is fulfilled, the scheme starts to converge. Very complex cases with very sharp discontinuities can be easily computed. And the explicit scheme can be coded in a couple of hundred lines. One just has to: read the mesh and do the geometry (elements areas, edges normals, extra DoFs,...), initialize the solution, and at each time step compute the local nodal residuals and update the solution, taking into account the boundary conditions. An iteration is then computationally costless. But on the other hand, the convergence is very slow and one has to perform a lot of iterations to reach the steady state of equation (5.18). The convergence rate is measured by a norm of vector $\left(\sum_{\mathrm{T}\in\mathcal{D}_i}\Phi_i^{\mathrm{T}}(\mathbf{U}^n)\right)_{i\in\mathcal{M}_h}$. We usually use the $\mathcal{L}^2$ norm. For a same given problem, the explicit version of the scheme requires 10 to 100 times more iterations than the implicit version to fully converge. The difference comes mainly from the pseudo time step. While explicit scheme time step is restricted for stability, we show the implicit scheme is unconditionally positive. At the end of an implicit simulation, the pseudo time steps can be arbitrarily large. Furthermore, the domain of influence of a node during an iteration of an explicit scheme is just its direct neighbors. The solution propagates inside the domain at the speed of the advection. Whereas in the implicit scheme the solution is updated globally and nodes far from the boundaries are already updated at iteration 2.

5.2.2 The Implicit Scheme

At each time step, the solution of the numerical scheme is updated using:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \omega_i^n \sum_{\mathrm{T}\in\mathcal{D}_i} \Phi_i^{\mathrm{T}}(\mathbf{U}_h^{n+1}), \quad \forall i \in \mathcal{M}_h. \tag{5.25}$$

Scalar Case : We first start by demonstrate that this scheme in its scalar version is unconditionally positive. As for the explicit scheme, we suppose it can be put into the form (4.27).

Property 5.1 (*Unconditional Positivity*)

For any pseudo time step $\Delta\tau$, if the nodal residuals can be expressed as in (4.27)*, the scheme* (5.25) *in its scalar form verifies the global discrete maximum principle*

$$\forall i \in \mathcal{M}_h, \quad \min_{j\in\mathcal{M}_h} u_j^n \leqslant u_i^n \leqslant \max_{j\in\mathcal{M}_h} u_j^n \tag{5.26}$$

Proof: We start by defining the vector of unknown $\mathbf{U}^n$ by

$$\forall i \in \mathcal{M}_h, (\mathbf{U}^n)_i = u_i^n,$$

and the two constant vectors $\mathbf{U}_{min}^n$ and $\mathbf{U}_{max}^n$ by

$$\forall i \in \mathcal{M}_h, \quad (\mathbf{U}_{min}^n)_i = \min_{j\in\mathcal{M}_h} u_j^n,\ (\mathbf{U}_{max}^n)_i = \max_{j\in\mathcal{M}_h} u_j^n.$$

Then one can write $\mathbf{U}^n_{min} \leqslant \mathbf{U}^n \leqslant \mathbf{U}^n_{max}$.

If one considers equation (4.27), scheme (5.25) is reformulated into:

$$\mathcal{A}\mathbf{U}^{n+1} = \mathcal{B}\mathbf{U}^n \tag{5.27}$$

with

$$\begin{array}{cc} \mathcal{A}_{ii} = \frac{1}{\omega_i^n} + \sum_{j\in\mathcal{D}_i} \tilde{c}_{ij} & \mathcal{A}_{ij} = -\tilde{c}_{ij} \\ \mathcal{B}_{ii} = \frac{1}{\omega_i^n} & \mathcal{B}_{ij} = 0 \end{array}$$

$\tilde{c}_{ij}$ being defined by (4.31), page 72. Matrix $\mathcal{B}$ has only positive coefficients, then

$$\mathcal{A}\mathbf{U}^{n+1} = \mathcal{B}\mathbf{U}^n \geqslant \mathcal{B}\mathbf{U}^n_{min} = \mathcal{A}\mathbf{U}^n_{min}. \tag{5.28}$$

If the scheme is *Local Extremum Decreasing*, the $\tilde{c}_{ij}$ are all positive and $\mathcal{A}$ is diagonal dominant. This implies $\mathcal{A}$ is invertible and $\mathcal{A}^{-1}$ has only positive coefficients [118]:

$$\mathcal{A}^{-1}_{ij} \geqslant 0, \quad \forall i,j \in \mathcal{M}_h.$$

We can then multiply both sides of (5.28) by $\mathcal{A}^{-1}$ and obtain the lower part of equation (5.26). A similar reasoning for the upper part gives the complete result. ■

Vectorial Case : Once more, this demonstration can not be extended to the system case at that moment. In fact, all the reasonning can be generalized to vectorial unknowns except one thing. Let us explain this point and start the generalization of the proof.

We suppose the system has m unknowns and the mesh has n degrees of freedom. Then the problem has size $n.m$, the vector of unknowns having n components, each one of them being a vector of size m. We build then $\mathbf{U}^n_{min}$ and $\mathbf{U}^n_{max}$ such that

$$\forall i \in \mathcal{M}_h,\ (\mathbf{U}^n_{min})_i \leqslant (\mathbf{U}^n)_i \leqslant (\mathbf{U}^n_{max})_i.$$

Equation (5.25) is recast into

$$\mathcal{A}\mathbf{U}^{n+1} = \mathcal{B}\mathbf{U}^n \tag{5.29}$$

with

$$\begin{array}{cc} \mathcal{A}_{ii} = \frac{\mathbf{I}}{\omega_i^n} + \sum_{j\in\mathcal{D}_i} \tilde{c}_{ij} & \mathcal{A}_{ij} = -\tilde{c}_{ij} \\ \mathcal{B}_{ii} = \frac{\mathbf{I}}{\omega_i^n} & \mathcal{B}_{ij} = 0 \end{array}$$

where $\mathbf{I}$ is the identity matrix and $\tilde{c}_{ij}$ are $m \times m$ positive matrices in the sense of (4.34), because the scheme is supposed to be *Local Extremum Decreasing*. Thus, equation (5.28) is still true, with $\mathcal{A}$ being a diagonal block dominant matrix. What is missing is a theorem showing that $\mathcal{A}$ must be invertible and that $\mathcal{A}^{-1}$ has only positive blocks.

Anyway, by experience the implicit scheme behaves perfectly in the system case. The initial extrema are maintained troughout the simulation whatever the pseudo time step could be.

Practical Computation : Of course, as only $\mathbf{U}^n$ is known, it is impossible to compute $\Phi_i^{\mathrm{T}}(\mathbf{U}_h^{n+1})$. But the residuals depend continuously of the values of the solution and it is then possible to linearize the values of the local nodal residuals by

$$\Phi_i^{\mathrm{T}}(\mathbf{U}^{n+1}) \approx \Phi_i^{\mathrm{T}}(\mathbf{U}^n) + \sum_{j\in\mathcal{M}_h} \frac{\partial \Phi_i^{\mathrm{T}}(\mathbf{U}^n)}{\partial \mathbf{U}_j} \left(\mathbf{U}_j^{n+1} - \mathbf{U}_j^n\right). \tag{5.30}$$

Thus, if one uses notation

$$\Delta \mathbf{U}_j^n = \mathbf{U}_j^{n+1} - \mathbf{U}_j^n, \tag{5.31}$$

and the fact that the Φ_i^{T} only depends on the values of the solution at the degrees of freedom of T, equation (5.25) is rewritten into

$$\left(\frac{\mathbf{I}}{\omega_i^n} + \sum_{\mathrm{T}\in\mathcal{D}_i} \frac{\partial \Phi_i^T(\mathbf{U}^n)}{\partial \mathbf{U}_i}\right)\Delta \mathbf{U}_i^n + \sum_{\substack{j\in\mathcal{D}_i \\ j\neq i}} \left(\sum_{\mathrm{T}\in\mathcal{D}_i\cap\mathcal{D}_j} \frac{\partial \Phi_i^{\mathrm{T}}(\mathbf{U}^n)}{\partial \mathbf{U}_j}\right)\Delta \mathbf{U}_j^n = \sum_{\mathrm{T}\in\mathcal{D}_i} \Phi_i^{\mathrm{T}}(\mathbf{U}_h^n), \tag{5.32}$$

which is a matrix system in $\Delta\mathbf{U}^n$. $\mathbf{I}$ is the $m = d+2$ identity matrix and the right hand side ($\mathcal{RHS}$) is the *explicit* residual.

The main point is at this time to compute the Jacobians of the nodal residuals: $\dfrac{\partial \Phi_i^T(\mathbf{U}^n)}{\partial \mathbf{U}_j}$. For example, limitation formula (5.12) is not everywhere differentiable. Once more we have here several solutions, each one of them having its advantages and drawbacks. To understand well why many possibilities are offered, let us give a look to the huge matrix of problem (5.32), defined by $d+2$ blocks. Because the scheme is unconditionally stable, we look at the matrix for different values of $\omega_i^n \in \mathbb{R}^+$. This matrix is sparse. We have $\frac{1}{\omega_i^n}$ everywhere on the diagonal and the $(d+2)\times(d+2)$ block at line i and row j is non null if and only if node i and j are direct neighbors (belonging to a same element). The smaller the time steps ω_i^n are, the more dominant the diagonal coefficients are. Thus at the limit $\omega_i^n \to 0$, we obtain the fully *explicit* scheme. On the other hand, if we consider ω_i^n going to infinity, the scheme turns into something looking as

$$u_{n+1} = u_n - \left(f'(u_n)\right)^{-1}.f(u_n),$$

which is the global formulation of a Newton scheme. It is well known that the Newton scheme does not always converge. But when it does, it converges very well (in a quadratic manner). We need to be close enough to the solution to be in its basin of attraction. For this reason, in the *implicit* case ω_i^n does not ensure the stability but can be seen as a potentiometer between robust but slow fully explicit scheme and powerful, fast but possibly unadapted Newton scheme. Then the Jacobians forming the big matrix are descent directions, and because we just aim for the steady state, these directions do not need to be exact. This is very interesting because computing the Jacobians exactly is expensive. We present here the different ways to approximate these Jacobians.

5.2.3 First Order Jacobians

In a first approach, we approximate the exact Jacobians by the Jacobians of the first order nodal residuals (5.8) page 91, where α^T is considered to be constant. The matrices of the vector of matrices $\dfrac{\partial \overrightarrow{\boldsymbol{\mathcal{F}}}}{\partial \mathbf{U}}$ have been given in the case of a 2D domain in Subsection 2.2.9. Let us compute line i of the linearized problem. The Jacobians write

$$\frac{\partial \Phi_i^{\mathrm{T}}(\mathbf{U}^n)}{\partial \mathbf{U}_j} \approx \begin{cases} \frac{1}{q}\left(w_i \dfrac{\partial \overrightarrow{\boldsymbol{\mathcal{F}}}}{\partial \mathbf{U}}(\mathbf{U}_i).\vec{\mathbf{n}}_i + (q-1)\alpha^{\mathrm{T}}\mathbf{I}\right), & \text{if } j = i \\ \frac{1}{q}\left(w_j \dfrac{\partial \overrightarrow{\boldsymbol{\mathcal{F}}}}{\partial \mathbf{U}}(\mathbf{U}_j).\vec{\mathbf{n}}_j - \alpha^{\mathrm{T}}\mathbf{I}\right), & \text{if } j \neq i \end{cases} \tag{5.33}$$

where the vector w is the set of coefficients of the linear combination of the $\vec{\boldsymbol{\mathcal{F}}}_j.\vec{\mathbf{n}}_j$ in the computation of Φ^{T}, see equations (5.5), (5.6) and (5.7). We recall that $\vec{\mathbf{n}}_i$ is the inward normal to the opposite edge of i when it is a vertex of T, or the outward normal to the edge i is belonging to when it is an extra DoF. We give here the vector w in the $\mathbb{P}^k$ case

- $k = 1$:
$$w = [\frac{1}{2}, \frac{1}{2}, \frac{1}{2}]$$
- $k = 2$:
$$w = [\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}]$$
- $k = 3$:
$$w = [\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{3}{8}, \frac{3}{8}, \frac{3}{8}, \frac{3}{8}, \frac{3}{8}, \frac{3}{8}, 0]$$

Remark 5.2

In the fourth order case, we can notice the zero at the last component of w corresponding to the 10^{th} node situated inside the triangle. This will be also the case for all the degrees of freedom that do not lie on the edges of T. It is however not a bad news, because the diffusive part of the Lax-Friedrichs scheme is still distributing something to these nodes. The value of these nodes being involved in the global scheme they cannot be arbitrary.

Because the $\dfrac{\partial \vec{\boldsymbol{\mathcal{F}}}}{\partial \mathbf{U}}$ are known, these Jacobians are easy to compute and this method is relatively fast. The problem is that the descent direction is really too different from the exact Newton one. The quadratic convergence of the Newton method is never met in that case. But compared to the *explicit* scheme, the method is really efficient in terms of the number of iterations and of the CPU time. One starts with small time steps in order to be sure to go toward the steady solution and as soon as the residual $\varepsilon_2 = \| \left(\sum_{\mathrm{T} \in \mathcal{D}_i} \Phi_i^{\mathrm{T}}(\mathbf{U}^n) \right)_{i \in \mathcal{M}_h} \|_2$ is enough reduced, one increases the time steps an switches to the pseudo Newton method.

A practical study of the different methods of resolution is done on the 3D Bump test case presented in Subsection 7.3.1, page 151. In particular, we compare the efficiency of these linear Jacobians with the ones we are presenting next, that are a bit more complex to compute, but that tremendously help to reach the Newton quadratic convergence.

5.2.4 Finite Difference Jacobians

Another approach that has been developed during this thesis is to evaluate the Jacobian by finite differences. The problem is that it is 2 to 3 times more expensive than the previous method. In this case the quadratic convergence can be met and the steady state is obtained much faster, especially when machine zero is sought. In the case of the first order Jacobian, the convergence rate usually slows down when approaching the machine zero ($\varepsilon_2 \leqslant 10^{-6}$), whereas in the case of finite differences, it tends to accelerate. All the following discussion is illustrated by the 3D bump problem presented in subsection 7.3.1, page 151. One can especially give a look to Figures 7.11 and 7.12 page 153, for a comparison between this Jacobian approximation and the one described in last subsection.

The Jacobian matrices are filled in line by line. Line $l \in [\![1, m]\!]$ (l^{th} variable) of the $(d+2) \times (d+2)$ block situated at line i and row j is filled in with

$$\left(\frac{\Phi_i^{\mathrm{T}}(\mathbf{U}^n + \delta_l \mathbf{V}_{jl}) - \Phi_i^{\mathrm{T}}(\mathbf{U}^n)}{\delta_l} \right)^{\mathbf{T}}, \tag{5.34}$$

where $\mathbf{V}_{jl}$ is a vector having the same size as $\mathbf{U}^n$, having 1 on the line corresponding to the l^{th} variable of node j, and zeros everywhere else. $^{\mathbf{T}}$ represents the vector transposition. δ_l is the finite difference parameter. Its value determines the precision of the approximation and depends on the variable considered. It should not be too small in order to avoid round off problems, and not too big in order to obtain an accurate Jacobian. In our computations, we usually use the following heuristic formula

$$\delta_l = \max(10^{-10}, 10^{-8}. \max_{j \in \mathcal{M}_h} |\mathbf{U}_{jl}^n|). \tag{5.35}$$

As one can see, this method requires to compute $\frac{mk(k+1)}{2}$ times more nodal residuals than the explicit scheme. It is expensive, but Figures 7.11 and 7.12 page 154 shows it is worth it, in terms of CPU time or iterations. The main drawback of this method is pretty much the same as the one of the Newton method. At the beginning of a simulation, the domain is usually initialized with a homogeneous constant solution which is far away from the steady solution. One has then to start with very small time steps in order to converge robustly. Then why use a complex expensive method to finally use a scheme equivalent to the explicit one ? That is why, in some cases we start with the first order Jacobian implicit method until the global residual has been divided by a certain amount (between 10 and 100), and then switch to the faster finite difference method.

5.2.5 Exact Jacobians

Finally, we have investigated a third method which is nowadays a total failure. We have not found so far the reasons why this method is not working, even if it seems promising on the paper. It should be faster than the finite differentiate and cost less in term of calculations. The idea is to differentiate the program that generates the residual with respect to some input variables (the nodal value of the solution in our case). This can be done automatically with the INRIA software TAPENADE[4], see [62]. To explain quickly how it works, here is an example with the following Fortran 95 code:

```
SUBROUTINE test(x,f)
  REAL, DIMENSION(:), INTENT(in)  :: x
  REAL,               INTENT(out) :: f
  f=SUM(x**2)
END SUBROUTINE test
```

then TAPENADE sends back

[4] http://tapenade.inria.fr:8080/tapenade/index.jsp

```
SUBROUTINE TEST_D(x, xd, f, fd)
  IMPLICIT NONE
  REAL, DIMENSION(:), INTENT(IN) :: x
  REAL, DIMENSION(:), INTENT(IN) :: xd
  REAL, INTENT(OUT) :: f
  REAL, INTENT(OUT) :: fd
  REAL, DIMENSION(SIZE(x)) :: arg1
  REAL, DIMENSION(SIZE(x)) :: arg1d
  INTRINSIC SUM
  arg1d(:) = 2*x*xd
  arg1(:) = x**2
  fd = SUM(arg1d(:))
  f = SUM(arg1(:))
END SUBROUTINE TEST_D
```

which still compute f as a function of $\mathbf{x}$, but also the directional derivatives $\frac{\partial f}{\partial \mathbf{x}}.\mathbf{xd}$. Then the following main program

```
PROGRAM main
  REAL, DIMENSION(5)              :: x
  REAL                            :: f,fd
  x=(/ 1.0, 5.0, 3.0, 1.0, 6.0 /)
  CALL test_d(x,(/1.0,0.0,0.0,0.0,0.0/),f,fd)
  PRINT*, f,fd
END PROGRAM main
```

prints on the screen

$$72.000000 \quad 2.0000000$$

and if one uses $(/0.0, 2.0, 0.0, 0.0, 0.0/)$ for $\mathbf{xd}$, one gets

$$72.000000 \quad 10.0000000$$

We have applied this software to the procedure that computes the nodal residuals and asked to differentiate it exactly with respect to vector $\mathbf{U}^n$. The critical non differentiable points have been regularized. For example, the absolute value function is replaced by

$$|x| \approx \begin{cases} |x|, & \text{if } |x| \leqslant \varepsilon \\ \frac{x^2+\varepsilon^2}{2\varepsilon}, & \text{else} \end{cases} \tag{5.36}$$

Unfortunately, we have not been able to compute one single simple case with this method. The simulation crashes after a finite number of iterations. It would be interesting to go further into this approach, as it is less expensive that the *finite differences* and should show some better convergence.

5.3 Convergence Problems and Stabilization Term

The main reason we have been looking for a "*upwinding Limitation*" is that it is a sure cure to the main flaw of the Limited Lax-Friedrichs scheme (LLxF). In order to illustrate this flaw, we make use of the two following scalar problems:

1. **Circular Advection:** the domain is the square $[0;1]^2$ and the scalar solution verifies

$$\begin{cases} -y\dfrac{\partial u}{\partial x} + x\dfrac{\partial u}{\partial y} = 0, & \forall (x,y) \in [0;1]^2 \\ u(0,y) = \cos^2(\pi y), & \forall y \in [0;1]. \end{cases} \tag{5.37}$$

   The advection speed $\vec{\boldsymbol{\lambda}} = \begin{pmatrix} -y \\ x \end{pmatrix}$ is circular and the exact solution is just the rotation of the entering profile at $x = 0$.

2. **Burger Equation:** the domain is $\Omega = [0;1]^2$ and the scalar problem writes

$$\begin{cases} \dfrac{\partial u}{\partial y} + u\dfrac{\partial u}{\partial x} = 0, & \forall (x,y) \in \Omega \\ u(x,0) = 1 - 2x, & \forall x \in [0;1] \\ u(0,y) = 1, & \forall y \in [0;1] \\ u(1,y) = -1, & \forall y \in [0;1] \end{cases} \tag{5.38}$$

   The exact solution is given by a fan in region

$$\{(x,y) \in \Omega; y \leqslant x \text{ and } y \leqslant 1 - x\},$$

   a vertical shock starting at point $(0.5, 0.5)$ and two constant plateau at value 1 and -1 on both sides.

As one can see on Figure 5.3, the convergence rate of the LLxF scheme for problem (5.37) is really poor compared to the first order LxF scheme or the PSI one. And if we look at the solution on Figure 5.4, the isolines are all wiggled. It is absolutely not a problem of stability, because we have shown the scheme is $\mathcal{L}^\infty$ stable. It is a problem of convergence: we can see that through the fact that the scheme has not reached the steady state. What is even more interesting is looking to the solution of (5.38) that shows discontinuities and that is also represented on Figure 5.4. Here we see that the shock is well resolved, in one cell, and that the wiggles only appear in the smooth regions. They apparently do not come from the discontinuity but from some spurious modes the scheme is not able to dump. This is a general remark about this problem, as the discontinuities are always well handled and the wiggles always occur in the smooth parts of the flow. Then the full convergence is never reached and, even if the limited version of the LxF scheme is theoretically second order, only first order is observed in practice. We are next going to see qualitatively the origin of these spurious modes and describe concretely the way we overcome this problem.

5.3.1 Nature of the Problem

The problem we are encountering is a difficult problem for which we can unfortunately provide only qualitative answers. Let us come back to a **scalar problem** for sake of simplicity. If we first neglect the boundary conditions or consider them included into the *nodal residuals*, we have already seen the scheme reads

$$\sum_{\mathrm{T} \in \mathcal{D}_i} \Phi_i^{\mathrm{T}}(u_h) = 0, \ \forall i \in \mathcal{M}_h. \tag{5.39}$$

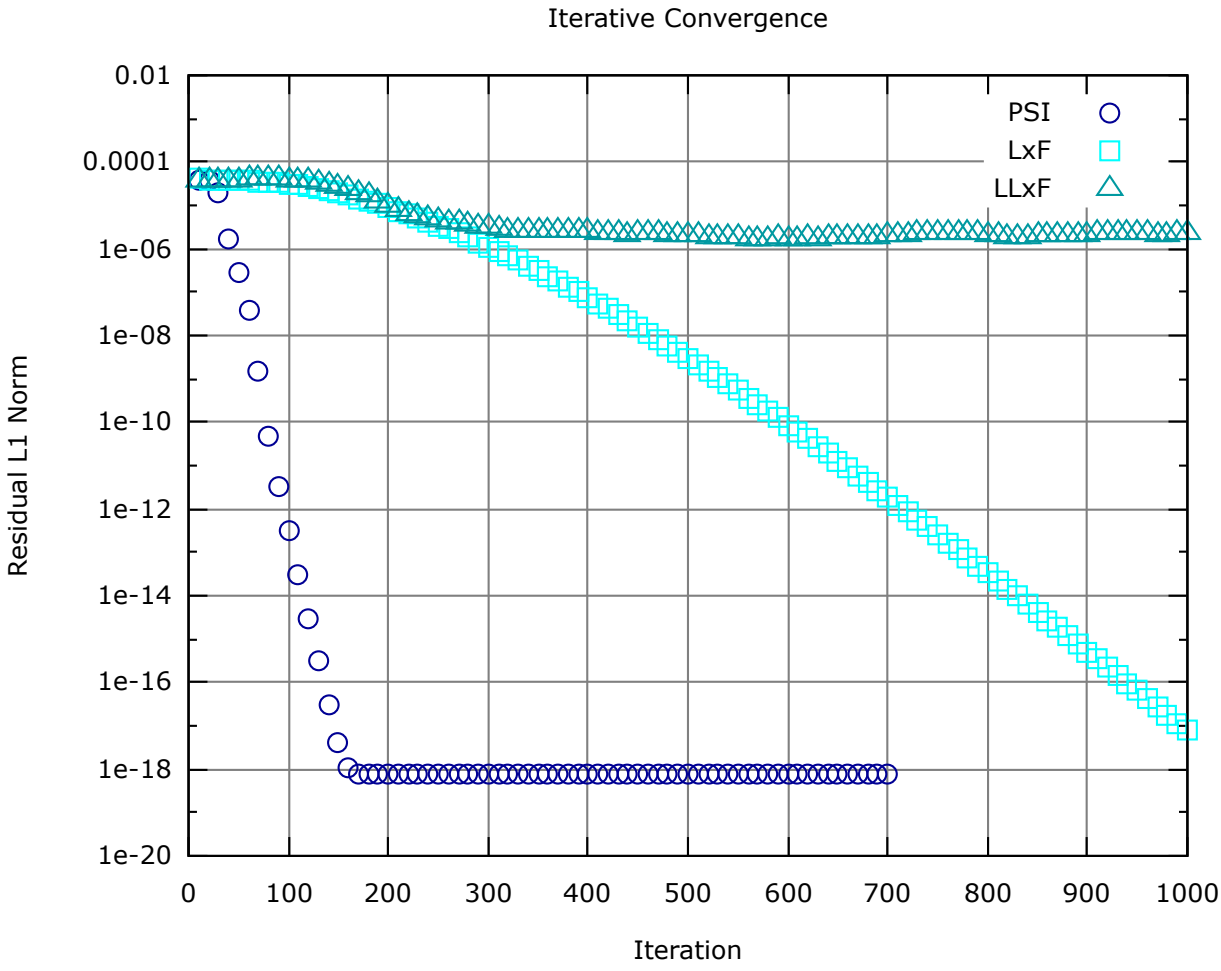


Figure 5.3: Iterative convergence curve for problem (5.37) treated with the second order PSI scheme, the first order Lax-Friedrichs scheme and the theoretically second order limited version of the Lax-Friedrichs scheme.

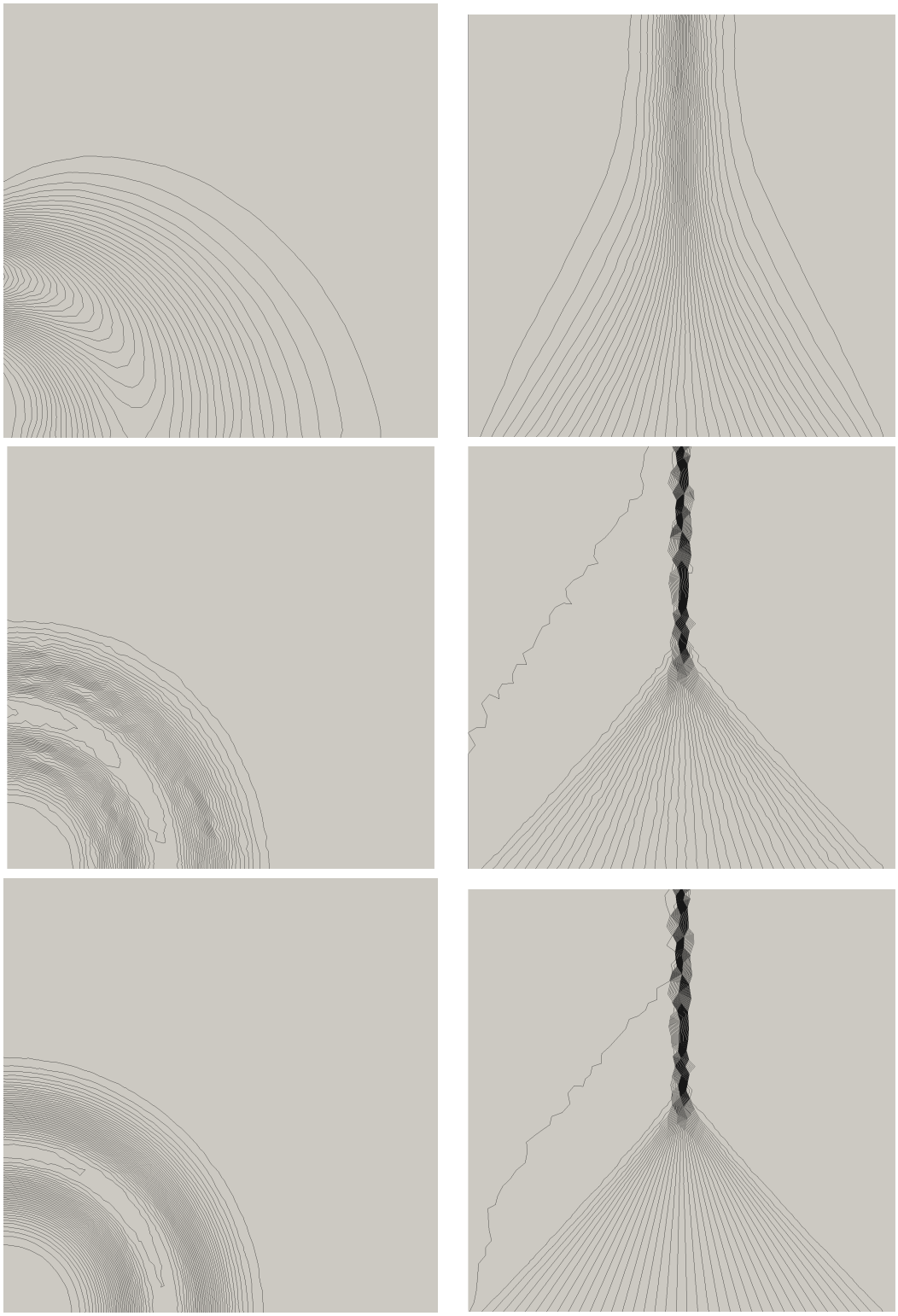


Figure 5.4: Isolines of the solution of problem (5.37) and (5.38) obtained with the non limited (first row) and the limited (second row) version of the Lax-Friedrichs scheme, and with the second order PSI scheme (third row). It is clear the non limited version of the LxF scheme is very dissipative and thus first order. The limited version should be second order, but because of the appearance of spurious modes, we do not get convergence to machine zero and the solution is finally first order. The PSI solution is used as a reference.

If we use form (4.27) and separate the influence of the boundary conditions, we get

$$\sum_{\mathrm{T}\in\mathcal{D}_i}\sum_{j\in\mathrm{T}} c_{ij}^{\mathrm{T},*}\,(u_i-u_j) = l(u_h),\ \forall i\in\mathcal{M}_h, \tag{5.40}$$

which can be put into the non linear matrix form

$$A(\mathbf{U})\mathbf{U} = \mathcal{L}(\mathbf{U}), \tag{5.40}$$

where $\mathbf{U}$ is the vector of unknowns and $\mathcal{L}(\mathbf{U})$ is the contribution of the boundary conditions. From the previous discussions, it comes for the LxF scheme

$$c_{ij}^{\mathrm{T},*} = \gamma_i^{\mathrm{T}} c_{ij}^{\mathrm{T}} = \frac{\beta_i^*}{\beta_i^{\mathrm{T}}}\frac{(\alpha^{\mathrm{T}}-k_j)}{q} \geqslant 0. \tag{5.41}$$

Then matrix $A=(a_{ij})_{i,j\in\mathcal{M}_h}$ which coefficients are

$$a_{ii} = \sum_{\mathrm{T}\in\mathcal{D}_i}\sum_{j\in\mathrm{T}} c_{ij}^{\mathrm{T},*}, \quad a_{ij} = \begin{cases} -\sum_{\mathrm{T}\in\mathcal{D}_i\cap\mathcal{D}_j} c_{ij}^{\mathrm{T},*}, & j\in\mathcal{D}_i \\ 0, & \text{else} \end{cases}$$

is diagonally dominant. Its coefficients are positive on the diagonal and negative elsewhere. This is a very good start to show well-posedness. Unfortunately, in the case of the Lax-Friedrchs scheme, it is absolutely possible there exists a node in $\mathcal{M}_h$ such that

$$\beta_i^{\mathrm{T},*} = 0, \quad \forall \mathrm{T}\in\mathcal{D}_i, \tag{5.42}$$

and the associated equation writes $0=0$. The value of u_h at node i is not determined but can usually exist only in a certain interval. The non linear associated algebraic problem is then ill-posed, has no fully determined solution, and there is no chance the time stepping method (5.19) converges. This explanation matches exactly what we observe on the convergence curves on Figure 5.3 and 5.4. In a first part, the scheme converges well, ensuring the global constraints of the problem. After some iterations, the steady state solution appears, and the value of the node where the algebraic problem occurs have reach their intervals of constraint. The solution has now enough degrees of freedom to let some spurious mode appear and the convergence stops.

To understand well why such a situation as (5.42) may exist, one has to remark that the Limited Lax-Friedrchs scheme is a totally centered scheme. There is nothing in its construction that gives a greater importance to one node than to another, for any physical reason. The direction of distribution is mainly given by the solution gradient in T, above all when the coefficient α^{T}, ensuring the *monotonicity preserving property*, is big. It can then occur that the signal is not necessarily sent in the direction of the advection, and situation illustrated on Figure 5.5 is plausible: node i receives absolutely no information from its neighbours. In the case of an *upwind* scheme, this situation cannot occur because every node i is situated downstream in at least one element of $\mathcal{D}_i$. It is therefore sure that this node will receive a part of the global residual of this element. We can then write:

$$a_{ii} > Ch, \quad \forall i\in\mathcal{M}_h, \tag{5.43}$$

and the associated algebraic problem is well-posed.

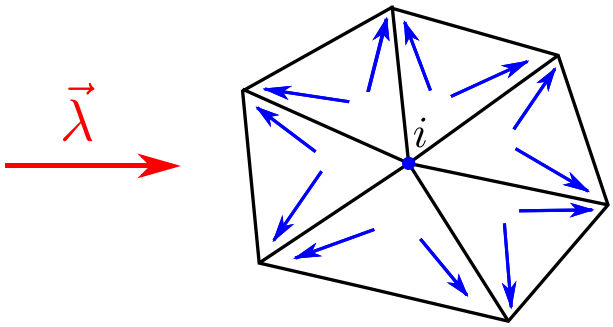


Figure 5.5: This figure illustrate equation (5.42). In the case of the simply limited LxFscheme, it can occur that some node i receives no information from its direct neighbours.

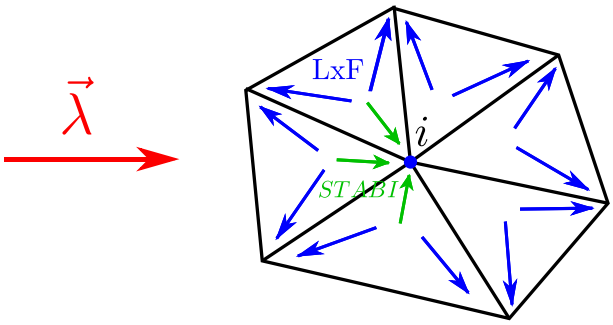


Figure 5.6: The SUPG-like term ensures every node to receive a certain signal by its upwind property.

5.3.2 Cure

The cure for this problem of ill-posedness comes from the SUPG scheme. As we have already seen in Subsection 4.4.6, the SUPG scheme is built with a centered signal $\frac{\Phi^{\mathrm{T}}}{3}$ plus a streamline dissipation

$$D_i^{\mathrm{T}} = \int_{\mathrm{T}} (\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^k})\bar{\boldsymbol{\tau}}(\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla u_h^k})\, d\mathbf{x}. \tag{5.44}$$

This last term has a dissipative property that actually stabilize the Galerkin scheme, and it has also an *upwind character* which is exactly what we are looking for, see Figure 5.6. If we consider a $\mathbb{P}^1$ scheme applied to a constant advection scalar problem, this term adds

$$\int_{\mathrm{T}} \bar{\boldsymbol{\tau}}(\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^1})^2 d\mathbf{x} > 0$$

on the diagonal of the matrix A, described in previous subsection. Then condition of well-posedness (5.43) is met and the scheme is going to converge.

As already seen in Subsection 4.1.3, matrix $\bar{\boldsymbol{\tau}}$ is needed for local nondimensionalization, in order the formulation stays consistent. Its characteristic size must be $\frac{h}{(\|\vec{\mathbf{u}}\|+c)}$, which is exactly the dimension of matrix N given by equation (4.52) page 81. When available, we will then use matrix N for $\bar{\boldsymbol{\tau}}$, and we have observed that the results were slightly better when using this option. Sometimes, there is no need to compute matrix N though, and because it is computationally costly (one has to compute the k_i, find their negative parts, and invert $\sum_{i\in\mathrm{T}}(k_i)^-$) and dangerous because we have seen N is not always defined, we rather use the term:

$$D_i^{\mathrm{T}} = \frac{h}{\alpha^{\mathrm{T}}} \int_{\mathrm{T}} (\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^k})(\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla u_h^k})\, d\mathbf{x}. \tag{5.45}$$

What we furthermore need is that this extra term does not destroy the properties of the LxF scheme. One has first to notice that because $\sum_{i\in\mathrm{T}} \varphi_i^k = 1$,

$$\sum_{i\in\mathrm{T}} D_i^{\mathrm{T}} = 0.$$

The global conservation of the scheme is then always maintained. We also absolutely need that this term is of the same order of accuracy as the global residual. We recall (see Property 4.9) that a necessary condition for a scheme to be of order $(k+1)$, is that the distributed signals to the nodes are of order $(k+2)$. Let u^* be the exact solution of the continuous scalar problem and $\pi_h^k u^*$ its $\mathbb{P}^k$ projection. We have

$$\begin{aligned}
D_i^{\mathrm{T}}(\pi_h^k u^*) &= \int_{\mathrm{T}} (\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^k})\bar{\boldsymbol{\tau}}(\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\pi_h^k u^*})\, d\mathbf{x} \\
&= \int_{\mathrm{T}} (\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^k})\bar{\boldsymbol{\tau}}\left(\vec{\boldsymbol{\lambda}}.\left(\overrightarrow{\nabla\pi_h^k u^*} - \overrightarrow{\nabla u^*}\right)\right) d\mathbf{x} \\
&= |\mathrm{T}| \times \mathcal{O}(h^{-1}) \times \mathcal{O}(h) \times \mathcal{O}(h^k) \\
&= \mathcal{O}(h^{k+2}),
\end{aligned}$$

which is expected, because $\bar{\boldsymbol{\tau}}$ is built to destroy the physical dimension of $\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^k}$. $D_i^{\mathrm{T}}(\pi_h^k u^*)$ has the same size as $\Phi^{\mathrm{T}}(\pi_h^k u^*)$. In the case one would not like to compute a matrix having the same

properties as $\bar{\boldsymbol{\tau}}$, it is conceivable to use a constant instead, as $\frac{h}{(\|\vec{\mathbf{u}}\|+c)}$ or simply h. What have been observed numerically is that the more effort is done, the more efficient the stabilization term is. A scheme using matrix N for $\bar{\boldsymbol{\tau}}$ will converge faster that its twin using h instead. However, for simplicity, we are usually going to consider that $\bar{\boldsymbol{\tau}} = h$ in the following.

Finally, as we have seen through the examples given in Subsection 5.3.1, the spurious modes occur only in the smooth regions. And the price to pay to converge with help of this new dissipative term is to loose the formal monotonicity. We can explain that quickly in the scalar explicit case. The scheme writes now:

$$\begin{aligned} u_i^{n+1} &= \left(1 - \omega_i^n \sum_{j\in\mathcal{D}_i} \gamma_i^{\mathrm{T}} \tilde{c}_{ij} + h\int_{\mathrm{T}} (\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^k})^2 d\mathbf{x}\right) u_i^n \\ &+ \sum_{j\in\mathcal{D}_i} \omega_i^n \sum_{\mathrm{T}\in\mathcal{D}_i\cap\mathcal{D}_j} \left(\gamma_i^{\mathrm{T}} c_{ij}^{\mathrm{T}} + h\int_{\mathrm{T}} (\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^k})(\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_j^k}) d\mathbf{x}\right) u_j^n \end{aligned} \tag{5.46}$$

u_i^{n+1} is a barycenter of the u_j^n, $j \in \mathcal{D}_i$ and the sum of the barycentric coefficients is 1. The scheme verifies a maximum principle if and only if

$$\sum_{\mathrm{T}\in\mathcal{D}_i\cap\mathcal{D}_j} \left(\gamma_i^{\mathrm{T}} c_{ij}^{\mathrm{T}} + h\int_{\mathrm{T}} (\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^k})(\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_j^k}) d\mathbf{x}\right) \geqslant 0, \quad \forall j \neq i.$$

This condition is unreachable as there must exist an element T in which $\beta_i^{\mathrm{T}} < 0 \Rightarrow \beta_i^{\mathrm{T},*} = 0 \Rightarrow \gamma_i^{\mathrm{T}} = 0$, and as soon as $\int_{\mathrm{T}} (\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^k})^2 d\mathbf{x} > 0$, there exists $j \in \mathrm{T}$ such that $\int_{\mathrm{T}} (\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_i^k})(\vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla\varphi_j^k}) d\mathbf{x} < 0$.

Now, there are two things: the stabilized scheme is not positive anymore, which is preoccupying for problems with shocks, and the limited first order scheme behaves well around the discontinuities. The solution is thus to stabilize the scheme only in the smooth regions. This is done by multiplying the dissipation term (5.44) by a *shock-capturing* function $\theta_{\mathrm{T}}(\mathbf{x}, u_h)$, defined by

$$\theta_{\mathrm{T}} = \begin{cases} 1, & \text{where } u_h \text{ is smooth} \\ h, & \text{in the discontinuities} \end{cases} \tag{5.47}$$

There are many possible choices for the parameter θ_{T}. The best choice we have experimented so far is

$$\theta_{\mathrm{T}} = 1 - \max_{i\in\mathrm{T}} \left(\max_{\mathrm{T}\in\mathcal{D}_i} \max_{j\in\mathrm{T}} \frac{|u_j - \bar{u}_{\mathrm{T}}|}{|u_j| + |\bar{u}_{\mathrm{T}}| + \varepsilon}\right), \tag{5.48}$$

where $\varepsilon = 10^{-12}$ or any positive number near to machine zero, and $\bar{u}_{\mathrm{T}} = (\sum_{j\in\mathrm{T}} u_j)/(\sum_{j\in\mathrm{T}} 1)$. One could notice this formulation is not compact anymore, as the value at node i does not depend only on the values at its direct neighbours. In fact, there is a way of computing this formula that maintains the maximal compactness of the scheme. This is presented in Algorithm 2. The trick is to add an extra variable $\tilde{\theta}_\sigma$ that allows to compute the compact part inside the parenthesis of Equation (5.48). The rest of the formula is evaluated only at next time step by copying $\tilde{\theta}_\sigma$ into θ_σ and using (5.49).

In the case of a multidimensional problem, Algorithm 2 can however not be used as it is. Equations (5.50) and (5.51) are only valid in the case of a scalar problem. For vectorial problems, the shock capturing is then only based on one variable, and we usually compute it by replacing the quantity u by the density or the entropy component.

Algorithm 2 Sketch of the implementation of one of the possible shock capturing function. The evaluation of θ_T (cf. equation (5.48)) is kept compact by updating and swapping the monitors θ_σ and $\tilde{\theta}_\sigma$.

Initialize by $\theta_\sigma = 1$ for all DoFs,
Set $\varepsilon = 10^{-12}$,
for each iteration k **do**
  Set $\tilde{\theta}_\sigma = 0$ for each σ,
  for each element T **do**
    Evaluate the local shock capturing coefficient θ_T, with

$$\theta_T = 1 - \max_{\sigma \in T} \theta_\sigma, \tag{5.49}$$

    Evaluate a mean value in T

$$\bar{u}_{\mathrm{T}} = \frac{\sum_{j \in \mathrm{T}} u_j}{\sum_{j \in \mathrm{T}} 1} \tag{5.50}$$

    Evaluate

$$\xi_{\mathrm{T}} = \max_{\sigma \in \mathrm{T}} \left(\frac{|u_\sigma - \bar{u}_{\mathrm{T}}|}{|u_\sigma| + |\bar{u}_{\mathrm{T}}| + \varepsilon} \right) \tag{5.51}$$

    for each $\sigma \in$ T **do**
      $\tilde{\theta}_\sigma = \max(\tilde{\theta}_\sigma, \xi_{\mathrm{T}})$
    end for
  end for
  Swap : $\theta_\sigma = \tilde{\theta}_\sigma$,
end for

5.3.3 Stabilization Term Computation

The goal of this section is to explain the practical computation of term (5.44). One first looks for an exact quadrature formula. If one uses a $\mathbb{P}^k$ polynomial representation, the integrand is of polynomial order $(k-1)^2$, and one needs a quadrature formula of at least $(k-1)^2$-th order of accuracy. Term (5.44) is computed as

$$D_i^{\mathrm{T}} = h|\mathrm{T}|\theta_T \sum_{\mathbf{x}_q \in \text{ quad}} \omega_q \left(\vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla\varphi_i^k}(\mathbf{x}_q)\right)\left(\vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla u_h^k}(\mathbf{x}_q)\right). \tag{5.52}$$

The problem is that a quadrature formula of $(k-1)^2$-th order of accuracy represents quickly a tremendous amount of quadrature points when k is growing. Then the question is: do we really need an exact quadrature, and if not, what is the criterion on the quadrature formula ensuring the dissipation term to play its role ? To answer this question, we need to define what the necessary properties of this term are. First, the term has to be of the same magnitude of accuracy as the nodal residuals. As we have already seen in the previous subsection, if we inject the $\mathbb{P}^k$ projection of the solution of the continuous problem into the dissipation term, all the terms of the quadrature sum will be of the desired order of accuracy.

$$h|\mathrm{T}|\theta_T \left(\vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla\varphi_i^k}(\mathbf{x}_q)\right)\left(\vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla\pi_h^k u^*}(\mathbf{x}_q)\right) = \mathcal{O}(h^{k+2}), \quad \forall \mathbf{x}_q \in \text{ quad}.$$

Second, we have to ensure the term has some dissipative properties, because we want it to distribute some information toward the ill-posed nodes and then dump the spurious modes. In other word, we need the following bilinear form

$$D_i^{\mathrm{T}}(u,v) = h|\mathrm{T}|\theta_T \sum_{\mathbf{x}_q \in \text{ quad}} \omega_q \left(\vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla u}(\mathbf{x}_q)\right)\left(\vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla v}(\mathbf{x}_q)\right) \tag{5.53}$$

to be positive definite. This reduces to ensure

$$D_i^{\mathrm{T}}(u,u) = 0 \quad \Longrightarrow \quad \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla u} \equiv 0. \tag{5.54}$$

This condition is met when all the weight coefficients ω_q are positive and the quadrature formula uses enough quadrature points to define uniquely the $(k-1)^{\text{th}}$ order polynomial $\overrightarrow{\nabla u_h}$. The computation of the stabilization term is sumed up in the three following points:

- The formal order of accuracy is unconditionally met;
- $\forall q \in \text{quad}, \omega_q > 0$, for example, ω_q is always 1 or $\frac{1}{\#\{\text{quad}\}}$;
- Quadrature formula uses $\frac{k(k+1)}{2}$ quadrature points:

$$\#\{\text{quad}\} = \frac{k(k+1)}{2}$$

and if we finally consider the general case of a vectorial problem, the practical computation of the stabilization term writes:

$$D_i^{\mathrm{T}} = h|\mathrm{T}|\theta_T \sum_{q=1}^{k(k+1)/2} \sum_{j \in \mathrm{T}} \mathbf{U}_j \left(\vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla\varphi_i^k}(\mathbf{x}_q)\right)\left(\vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla\varphi_j^k}(\mathbf{x}_q)\right) \tag{5.55}$$

Order	2	3	4	5
DoF	3	6	10	15
φ^{T}	3	6	9	12
D_i^t	1	3	6	10
Consistent	1	6	16	>>

Table 5.1: This tabular shows the number of quadrature points needed to compute the global residual and the dissipation term. Line D_i^t shows the number of points needed in our formulation, and line "Consistent" shows the number of points needed when an exact quadrature would have been used. The bottom right box just tells this number is very big in the 5^{th} order case. We have not find a quadrature rule integrating exactly a 2D polynomial of order 16!

One can compare on Tabular 5.1 the number of quadrature points needed in an exact quadrature formula with the number of quadrature point strictly necessary. With this small trick, we have very much reduced the computational cost of this dissipation term.

In the case of an implicit scheme, one wishes to find the Jacobian matrix associated to this extra term. That for, we make the hypothesis that the advection is constant (or at least not depending on the value of the solution) and the Jacobian is straightforward. The contribution of the dissipation to the i^{th} line and j^{th} row of the left hand side matrix is given by

$$\left(J_{Dissip}\right)_{ij} = h|\mathrm{T}|\theta_T \sum_{q=1}^{k(k+1)/2} \sum_{j\in\mathrm{T}} \left(\vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla\varphi_i^k}(\mathbf{x}_q)\right)\left(\vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla\varphi_j^k}(\mathbf{x}_q)\right) \tag{5.56}$$

Finally, one can look at Figures 5.7, and 5.8 to observe the effects of this additional term on the isolines of the solution, as well as on the associated convergence curve. The convergence is completed to machine zero and the obtained solution is much better. The results are of the same quality as those obtained with the PSI scheme.

5.4 Boundary Conditions

At this stage, we have not been much speaking about the boundary conditions. They have been mostly neglected for sake of simplicity. It is a difficult topic because their construction is often intuitive and their explanation never totally rigorous. In CFD, there are two types of boundary conditions: the strong and the weak ones. The strong boundary conditions are bound to the Dirichlet condition: $u_h(\mathbf{x}) = 0,\ \mathbf{x} \in \partial\omega$. A value is strongly imposed to one or several variables of the solution. This is the case of the *supersonic inflow* or the *solid wall* boundaries. They are interesting because the boundary condition is reliably exactly imposed. Nevertheless, these conditions are not very much appreciated because they are not fully consistent with the global formulation of the scheme. The scheme comes from the weak formulation of the continuous problem and one needs then to start from here to build the boundary conditions. What we generally obtain is an extra boundary flux to distribute to the degrees of freedom lying on the border of Ω. This is what we call the *weak* boundary conditions.

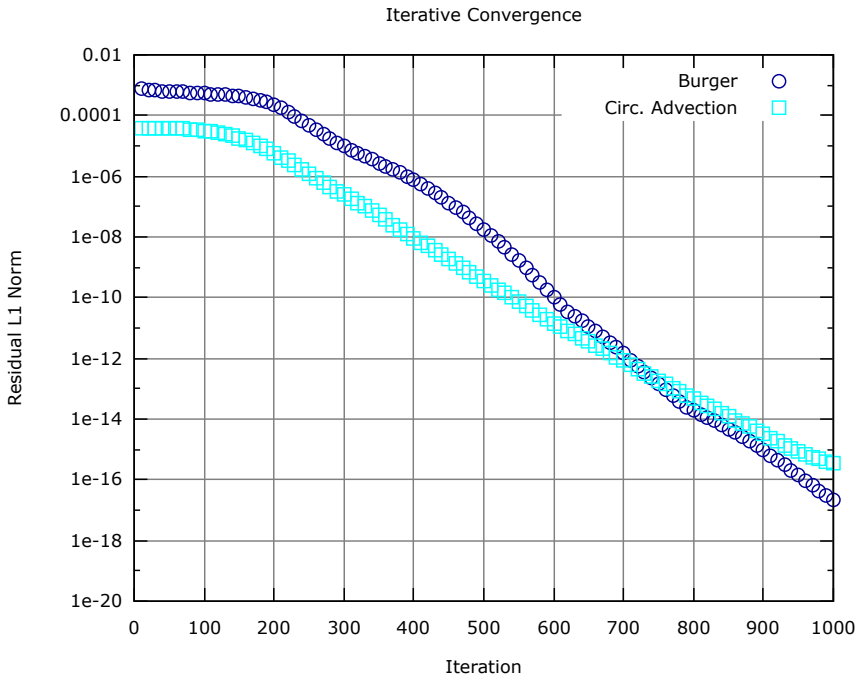


Figure 5.7: Iterative convergence for the stabilized Lax-Friedrichs scheme. The machine zero is reached and the theoretical second order of the scheme is met, as illustrated below.

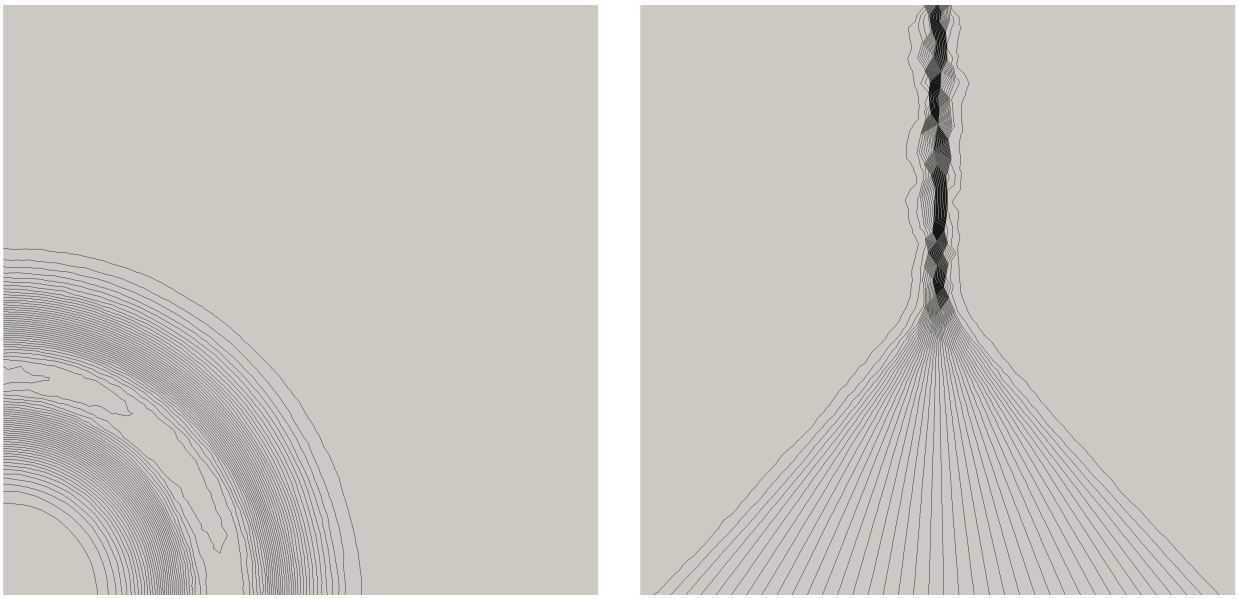


Figure 5.8: Iterative convergence for the stabilized Lax-Friedrichs scheme. The machine zero is reached and the theoretical second order of the scheme is met, as illustrated below.

5.4.1 Supersonic In/Out-Flow

The big advantage of the supersonic flows is that all the characteristics of the problem point toward the same direction. All the information is advected is the direction of the velocity. For a supersonic inflow, we are then sure nothing will go upstream and try to leave the domain. For the elements on the border, the nodal values completely depend on the values on the boundary. It is therefore possible to impose the input values strongly, without fearing any reflexion phenomenon. Because all the characteristics are entering the domain, we need to impose all the m variables in order the problem is well-posed (see Subsection 2.1.9), and the boundary condition is applied by ensuring

$$\mathbf{U}_i^n = \mathbf{U}_{out}, \quad \forall i \in \partial\Omega, \forall n. \tag{5.57}$$

For an explicit scheme, one covers all the boundary nodes and just nullify their received residuals. If condition (5.57) is fulfilled at time step $n = 0$, it will be ensured at any time step. In the case of an implicit scheme, we do the same for the residuals, plus we nullify the whole i^{th} line of blocks in the system matrix, just letting the $(d+2) \times (d+2)$ identity block at row i. Then one has

$$\Delta\mathbf{U}_i^n = 0 \Longrightarrow \mathbf{U}_i^{n+1} = \mathbf{U}_i^n = \mathbf{U}_i^0 = \mathbf{U}_{out}.$$

In the case of the outflow, it is even simpler. At any time step, all the information on the outflow border are radically blown out by the supersonic flow. This is what we want and this is exactly what happens numerically. Then we have to do nothing:

Proposition 5.3 (*Supersonic Outflow*)

The supersonic outflow boundary condition is applied by doing nothing more to the numerical scheme.

5.4.2 Solid Wall Boundary Conditions

Solid wall boundary conditions are useful in the case of a viscous fluid, which means when using a Navier-Stokes model. In the case of an Euler simulation, this conditions are usually replaced by the *slip wall* conditions, see bellow. When a fluid is viscous, the friction makes boundary layers appear in the vicinity of the solid wall, because the flow sticks to the surface. Then one wishes to ensure $\vec{\mathbf{u}} = \vec{0}$ on the boundary. This is done as in the previous subsection just by nullifying the residual linked to the speed of the flow for the degrees of freedom lying on the boundary. In the explicit case, we then just maintain the initial values of the speed on the wall (which must be 0). In the implicit case, we obtain the same result by moreover replacing the corresponding lines with the identity lines in the matrix of the system.

This method is however true only for a still walls. What if the wall is moving, as in a Couette flow, or a Stokes flow ? One wishes at that time to impose $\vec{\mathbf{u}} = \vec{\mathbf{u}}_{\text{wall}}$ on the boundary. The problem with nullifying the velocity residuals is that one maintains the momentum value and not the velocity, whereas the value of the density changes. The solution is to replace the velocity residuals by $\Delta\rho\,\vec{\mathbf{u}}_{\text{wall}}$. Thus, one has

$$\vec{\mathbf{u}}_i^{n+1} = \frac{(\rho\vec{\mathbf{u}})_i^{n+1}}{\rho_i^{n+1}} = \frac{(\rho\vec{\mathbf{u}})_i^n + \Delta\rho\,\vec{\mathbf{u}}_{\text{wall}}}{\rho_i^{n+1}} = \vec{\mathbf{u}}_{\text{wall}}.$$

This works in the implicit case with the appropriated matrix lines, but we also have a second possibility. Instead of changing the right hand side, we can maintain it to zero and replace the line of the diagonal block of the matrix corresponding to the velocity at i by

$$\left[\begin{array}{cccc} -u_x^{\text{wall}} & 1 & 0 & 0 \\ -u_y^{\text{wall}} & 0 & 1 & 0 \end{array}\right]. \tag{5.58}$$

This has exactly the same effect.

5.4.3 Slip Wall Boundary Conditions

As we have already said in the previous subsection, in the case of Euler simulations the fluid is considered to be non viscous, and it is not stuck to the walls. The fluid is nevertheless still not able to pass through the walls and the no-slip condition is changed into the slip condition $\vec{\mathbf{u}}.\vec{\mathbf{n}} = 0$.

As explained in Subsection 2.1.5, page 20, $\mathbf{U}$ is the solution of problem (5.1) with boundary conditions, if it verifies, for any $\varphi \in \mathcal{C}^1(\Omega)$

$$\begin{aligned} -\int_\Omega \overrightarrow{\nabla\varphi}.\overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U})d\mathbf{x} + \int_{\partial\Omega} \varphi\overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U}).\vec{\mathbf{n}}\, ds &= 0, \qquad (5.59)\\ \Leftrightarrow \sum_{\mathrm{T}\in\mathcal{M}_h}\left(-\int_{\mathrm{T}} \overrightarrow{\nabla\varphi}.\overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U})d\mathbf{x} + \int_{\partial\mathrm{T}\cap\partial\Omega} \varphi\overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U}).\vec{\mathbf{n}}\, ds\right) &= 0 \end{aligned}$$

with $\vec{\mathbf{n}}$ being the outward unit normal to the boundary. We here consider that the same boundary condition is applied to the whole edge of Ω. In the reality, there are usually many different boundary conditions to apply to the problem, and one has then to split the contour integral into the right pieces. Now, $\mathbf{U}_h$ approximates the exact solution as the unique solution of $\mathscr{W}_h^k = \text{Span}_{i\in\mathcal{M}_h}\left\{\varphi_i^k\right\}$ verifying (5.59) for any shape function φ_i associated to node i. If i is situated inside Ω, φ_i has a compact support in Ω and the right integral in (5.59) is zero. The scheme reduces to gather the signals coming from the different elements of $\mathcal{D}_i$. But if i lies on the boundary, the right integral is not null anymore and its role is to enforce the *slip wall boundary flux*, which is given for the Euler equations by

$$\overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U})_{|(\vec{\mathbf{u}}.\vec{\mathbf{n}}=0)}.\vec{\mathbf{n}} = \begin{pmatrix} 0 \\ pn_x \\ pn_y \\ 0 \end{pmatrix}. \tag{5.60}$$

Then for a DoF on the boundary, after applying the Green formula inside T to the left integral, the weak formulation over the mesh $\mathcal{M}_h$ reads:

$$\begin{aligned} \sum_{\mathrm{T}\in\mathcal{D}_i} \Big(&\int_{\mathrm{T}} \varphi_i^k.\mathrm{div}\left(\overrightarrow{\boldsymbol{\mathcal{F}}}_h(\mathbf{U}_h)\right) d\mathbf{x} \qquad (5.61)\\ &+ \int_{\partial\mathrm{T}\cap\partial\Omega} \varphi_i^k\left(\overrightarrow{\boldsymbol{\mathcal{F}}}_h(\mathbf{U}_h)_{|(\vec{\mathbf{u}}.\vec{\mathbf{n}}=0)} - \overrightarrow{\boldsymbol{\mathcal{F}}}_h(\mathbf{U}_h)\right).\vec{\mathbf{n}}\, ds\Big) = 0, \end{aligned}$$

which is the residual distribution plus a additional boundary term enforcing flux

$$\overrightarrow{\boldsymbol{\mathcal{F}}}_{\text{slip}}(\mathbf{U},\vec{\mathbf{n}}) = \left(\overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U})_{|(\vec{\mathbf{u}}.\vec{\mathbf{n}}=0)} - \overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U})\right).\vec{\mathbf{n}} = \begin{pmatrix} -\rho\vec{\mathbf{u}}.\vec{\mathbf{n}} \\ -\rho u\vec{\mathbf{u}}.\vec{\mathbf{n}} \\ -\rho v\vec{\mathbf{u}}.\vec{\mathbf{n}} \\ -\rho h\vec{\mathbf{u}}.\vec{\mathbf{n}} \end{pmatrix} \tag{5.62}$$

on the boundary edges. $h = E + p/\rho$ denotes the specific *enthalpy*.

Without any further explanation, this is exactly what we do in the case of a $\mathcal{RDS}$. We first compute the global residuals and distribute them to their respective DoFs. Afterward, we go all over the edges of $\mathcal{M}_h$ lying on the boundary, compute the terms

$$B_i^{\text{edge}} = \int_{\text{edge}} \varphi_i^k \overrightarrow{\boldsymbol{\mathcal{F}}}_{\text{slip}}(\mathbf{U}^n, \vec{\mathbf{n}})\, ds, \tag{5.63}$$

and add them to the residual of the corresponding boundary DoFs. One has to remark that as $\overrightarrow{\boldsymbol{\mathcal{F}}}_h$ is built as the $\mathbb{P}^k$ projection of the continuous flux $\overrightarrow{\boldsymbol{\mathcal{F}}}$, the computation of this term is just a linear combination of the values of the enforced flux at the degrees of freedom of the edge, which coefficients are the i^{th} line of the symmetric mass matrix

$$\left(\mathcal{M}^k\right)_{ij} = \int_0^1 \varphi_i^k \varphi_j^k\, ds. \tag{5.64}$$

The computational formula writes:

$$B_i^{\text{edge}} = \sum_{j \in \text{edge}} \left(\mathcal{M}^k\right)_{ij} \left(\overrightarrow{\boldsymbol{\mathcal{F}}}_h(\mathbf{U}_j)_{|(\vec{\mathbf{u}}.\vec{\mathbf{n}}=0)} - \overrightarrow{\boldsymbol{\mathcal{F}}}_h(\mathbf{U}_j)\right).\vec{\mathbf{n}}_{\text{edge}}, \tag{5.63}$$

where $\vec{\mathbf{n}}_{\text{edge}}$ is still the outward normal to the boundary but its norm is the length ($|\text{edge}|$) of the considered edge.

5.4.4 Far-field Conditions

In CFD, we are often simulating problems that require infinite large domains. We can of course not consider these domains entirely and we then use large computational domains such that the boundaries are far enough from the simulated aerodynamic object. It is therefore usual to consider these external boundaries as if they were situated at the infinity and that the solution is almost constant around these boundaries. We wish then to impose a far-field flux on these edges, as if the domain were drown in a infinite space filled with a homogeneous steady state. Because the equations are invariant by Galilean transformation, this will act as if the aerodynamic object was moving at the speed at infinity in a steady domain.

We have seen in Subsection 2.1.9 that the good way of treating boundary conditions is to enforce the external conditions only on the entering characteristics, and to let the solution be on the outgoing characteristics. In the case of the two dimensional Euler equations and for a subsonic flow, there are usually 3 entering characteristics and 1 outgoing one. Furthermore, we assume that the solution is constant enough on the vicinity of the boundary such that the advection is constant, and the flux can be approximated by

$$\overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U}) \approx \frac{\partial \overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U})}{\partial \mathbf{U}} \mathbf{U} = \vec{\boldsymbol{\lambda}}(\mathbf{U})\mathbf{U}. \tag{5.65}$$

Now the flux crossing an edge has two components. Because the problem is hyperbolic, if n_{edge} is the outward normal scaled by the length of the edge, one has

$$\begin{aligned} \overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U}).\vec{\mathbf{n}}_{\text{edge}} &\approx \vec{\boldsymbol{\lambda}}(\mathbf{U}).\vec{\mathbf{n}}_{\text{edge}}\, \mathbf{U} \\ &= \mathbf{K}_{(\mathbf{U},\vec{\mathbf{n}}_{\text{edge}})}\mathbf{U} \\ &= \mathbf{K}^+_{(\mathbf{U},\vec{\mathbf{n}}_{\text{edge}})}\mathbf{U} + \mathbf{K}^-_{(\mathbf{U},\vec{\mathbf{n}}_{\text{edge}})}\mathbf{U}. \end{aligned} \tag{5.66}$$

The last two terms represent the outgoing and ingoing flux respectively. Following, what has just been said, we want the ingoing flux to be the flux at infinity and the outgoing one to be the flux related to the solution. This is called the Steger-Warming flux and it is defined by

$$\overrightarrow{\boldsymbol{\mathcal{F}}}_{SW}(\mathbf{U},\mathbf{U}_\infty,\vec{\mathbf{n}}) = \mathbf{K}^-_{(\mathbf{U},\vec{\mathbf{n}})}\mathbf{U}_\infty + \mathbf{K}^+_{(\mathbf{U},\vec{\mathbf{n}})}\mathbf{U}. \tag{5.67}$$

If we follow the arguments in previous subsection 5.4.3, one needs to add the contributions of the edges sharing i to the residuals of a node i of the boundary. They write

$$\begin{aligned} B_i^{\text{edge},SW} &= \int_{\text{edge}} \varphi_i^k \left(\overrightarrow{\boldsymbol{\mathcal{F}}}_{SW}(\mathbf{U}^n,\mathbf{U}_\infty,\vec{\mathbf{n}}_{\text{edge}}) - \overrightarrow{\boldsymbol{\mathcal{F}}}_h(\mathbf{U}^n).\vec{\mathbf{n}}_{\text{edge}} \right) ds \\ &= \int_{\text{edge}} \varphi_i^k \left(\mathbf{K}^-_{(\mathbf{U},\vec{\mathbf{n}}_{\text{edge}})}(\mathbf{U}_\infty - \mathbf{U}) \right) ds. \end{aligned} \tag{5.68}$$

Once more the flux is supposed to be of the same polynomial order as the solution, and the Steger-Warming contribution is computed as

$$B_i^{\text{edge},SW} = \sum_{j\in\text{edge}} \left(\mathcal{M}^k\right)_{ij} \left(\mathbf{K}^-_{(\mathbf{U}_j,\vec{\mathbf{n}}_{\text{edge}})}(\mathbf{U}_\infty - \mathbf{U}_j) \right) \tag{5.69}$$

Boundary Condition Jacobians : In the case of an implicit scheme, one needs to compute the Jacobians of these boundary contributions and add them at the right place in the matrix of the problem. For the Steger-Warming boundary condition, it is not a difficult task, as the additional Jacobian at line i and row j is

$$\left(\mathcal{M}^k\right)_{ij} \mathbf{K}^-_{(\mathbf{U}_j,\vec{\mathbf{n}}_{ij})}. \tag{5.70}$$

This is also valid for the previous slip wall boundary condition. In this case, one has first to compute the Jacobian of the imposed flux,

$$J_{\text{slip}} = \frac{\partial \overrightarrow{\boldsymbol{\mathcal{F}}}_{\text{slip}}}{\partial \mathbf{U}},$$

and the Jacobian of the boundary contributions at line i and row j writes

$$\left(\mathcal{M}^k\right)_{ij} J_{\text{slip}}(\mathbf{U}_i,\vec{\mathbf{n}}). \tag{5.71}$$

5.5 Summary of the Effective Implementation

Here is a quick summary of this chapter. The goal is to fully describe in a couple of lines the way the Limited Stabilized Lax-Friedrichs scheme is implemented is $\mathbb{P}^2$. $\mathbf{U}$ represents the numerical solution at pseudo time-step n. The proposed method is implicit. For explicit scheme, just remove the items dealing with the left hand side matrix. The solution is either scalar or vectorial. Difference will be given when needed. Except $\mathcal{RHS}$ which represents the Right Hand Side (also called the explicit residual), all the notation have been already presented.

For all the elements T of the mesh do:

- Compute the **Global Residual** along the edges of T

$$\Phi^{\mathrm{T}} = \sum_{i=1}^{3} \frac{\overrightarrow{\boldsymbol{\mathcal{F}}}_i.\vec{\mathbf{n}}_i}{6} + \sum_{i=4}^{6} \frac{2}{3}\overrightarrow{\boldsymbol{\mathcal{F}}}_i.\vec{\mathbf{n}}_i$$

- Compute α^{T} as

$$\alpha^{\mathrm{T}} = \max_{i\in\mathrm{T}} \left(\|\vec{\mathbf{u}}_i\| + c_i\right) . \max_{\text{edge}} |\text{edge}|$$

  and for each degree of freedom of T, compute the **Nodal Residual**

$$\Phi_i^{\mathrm{T}} = \frac{1}{6}\left(\Phi^{\mathrm{T}} + \alpha^{\mathrm{T}} \sum_{j\in\mathrm{T}} (\mathbf{U}_i - \mathbf{U}_j)\right)$$

- In the case of a vectorial problem, apply algorithm 1 page 93. In the scalar case, compute the first order **Distribution Coefficients**

$$\beta_i^{\mathrm{T}} = \frac{\Phi_i^{\mathrm{T}}}{\Phi^{\mathrm{T}}},$$

  limit them

$$\beta_i^* = \frac{\left(\beta_i^{\mathrm{T}}\right)^+}{\sum_{j\in\mathrm{T}} \left(\beta_j^{\mathrm{T}}\right)^+}$$

  and get the second order **Nodal Residual**

$$\Phi_i^* = \beta_i^* \Phi^{\mathrm{T}}.$$

- Compute the **Stabilization Term**

$$D_i^{\mathrm{T}} = |\mathrm{T}|^2 \theta_T \sum_{q=1}^{k(k+1)/2} \sum_{j\in\mathrm{T}} \mathbf{U}_j \left(\vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla\varphi_i^k}(\mathbf{x}_q)\right) \left(\vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla\varphi_j^k}(\mathbf{x}_q)\right)$$

- Assemble the left hand side matrix, using either the first order Jacobians or the finite difference Jacobians with the matrix associated to the stabilization term

$$\left(J_{Dissip}\right)_{ij} = |\mathrm{T}|^2 \theta_T \sum_{q=1}^{k(k+1)/2} \sum_{j\in\mathrm{T}} \left(\vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla\varphi_i^k}(\mathbf{x}_q)\right) \left(\vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla\varphi_j^k}(\mathbf{x}_q)\right)$$

- Gather the received signals

$$\forall i \in \mathrm{T}, \quad \mathcal{RHS}(i) += \Phi_i^* + \mathbf{D}_i^{\mathrm{T}}$$

For all the edges lying on the boundary do:

- Compute and distribute to the DoFs of the edge the associated **Boundary Flux**, in the case of a weak boundary condition. Add the boundary flux Jacobians to the left hand side matrix. In the case of a strong boundary condition, do nothing. These conditions must be treated after all the weak boundary conditions have been covered.
- Apply the strong boundary conditions and their effects on the matrix.

Solve the obtained system, update the solution and go to next time step!

Part III

New Developments and Illustrations

Chapter 6

Hybrid Meshes

One of the main advantages of the $\mathcal{RD}$ Lax-Friedrichs scheme we are presenting in this thesis, is its easy generalization to any type of polyhedral element. Using the $\mathbb{Q}^k$ basis functions defined in Chapter 3 on any convex quadrangle, we discuss in this chapter the extension of the LLxF to the computations on hybrid meshes. As we shall see, the use of such meshes presents some interest when looking at the accuracy of the obtained solution and the computational time. So far, the method has only been developed for 2D problems, but we are convinced the results we are showing stay valuable for 3D meshes containing hexahedra.

6.1 Formulation of the Stabilized LLxF Scheme on Quadrangles

6.1.1 Global and Nodal Residuals

We recall that for any convex quadrangle Q there exists a unique $\mathbb{Q}^1$ diffeomorphism φ transforming the reference element $\widehat{\mathrm{Q}} = [0;1]^2$ into Q, completely described by formula (3.11). The $\mathbb{Q}^k$ basis functions defined on the reference element are transported to Q thanks to φ and we obtain $(k+1)^2$ basis functions on Q that are polynomial of order k along the edges of Q and that verify:

$$\forall i, j \in \mathrm{Q}, \quad \mathscr{Q}_i^k(\mathbf{x}_j) = \delta_{ij}.$$

The fact that the restriction of our approximated function is polynomial of the right order on the edges is very useful, because one just has to use the degrees of freedom of the edges and the right weight coefficients to compute the **Global Residual** of Q as a contour integral. This is shown on Figure 6.1.

We now have all the necessary elements to formulate the Lax-Friedrichs scheme on quadrangles, thus obtaining the first order distribution coefficients that we *limit* in order to obtain the $(k+1)^{\text{th}}$ order distribution coefficients. As one can see, nothing really changes compared to the triangular formulation, and the extension is straightforward. Concerning the *Stabilization Term*, there are some differences with respect to the $\mathbb{P}^k$ case. The next paragraph is devoted to this aspect.

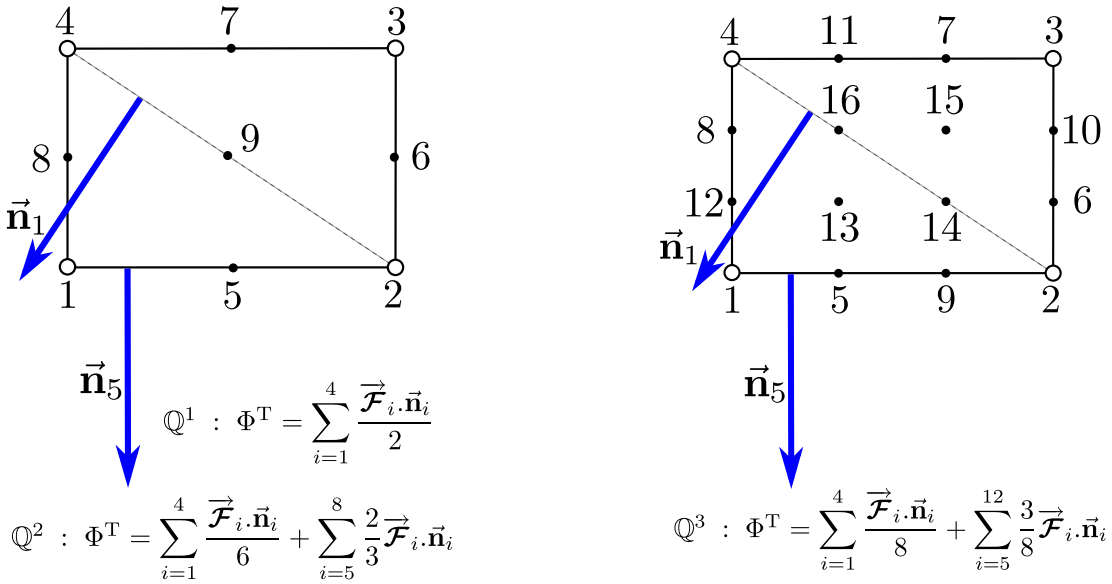


Figure 6.1: Global Residual computation in $\mathbb{Q}^1$, $\mathbb{Q}^2$ and $\mathbb{Q}^3$ quadrangles.

6.1.2 Stabilization Term Computation

As we have seen in Subsection 5.3.3, the *Stabilization Term* is calculated via a quadrature formula. In order to be efficient, we need enough quadrature points to define the gradient of the solution uniquely in the quadrangle. The problem is that the form functions are defined as the $\mathbb{Q}^k$ functions over the reference quadrangle composed with the $\mathbb{Q}^1$ transformation φ. We recall that the Jacobian of this transformation is denoted by J. Moreover, the gradient of a $\mathbb{Q}^k$ function does not have to be $\mathbb{Q}^{k-1}$. The only thing that is sure is that the gradient of the solution is a $\mathbb{Q}^k$ function and we are going to use all the DoFs of the quadrangle as quadrature points, in order for the *Stabilization Term* to have some dissipative properties. The *Stabilization Term* is computed as follows:

$$\begin{aligned} D_i^{\mathrm{Q}} &= h\theta^{\mathrm{Q}} \int_{\mathrm{Q}} \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla \mathscr{Q}_i}\ \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla u}\, d\mathbf{x} \\ &= h\theta^{\mathrm{Q}} \int_{\hat{\mathrm{Q}}} \vec{\boldsymbol{\lambda}}\left(\varphi(\hat{\mathbf{x}})\right).\overrightarrow{\nabla \mathscr{Q}_i}\left(\varphi(\hat{\mathbf{x}})\right)\ \vec{\boldsymbol{\lambda}}\left(\varphi(\hat{\mathbf{x}})\right).\overrightarrow{\nabla u}\left(\varphi(\hat{\mathbf{x}})\right)\ |J(\hat{\mathbf{x}})|\, d\hat{\mathbf{x}} \\ &\approx h\theta^{\mathrm{Q}} \sum_{q=1}^{(k+1)^2} \sum_{j \in \mathrm{Q}} u_j\ \vec{\boldsymbol{\lambda}}\left(\varphi(\hat{\mathbf{x}}_q)\right).\overrightarrow{\nabla \mathscr{Q}_i}\left(\varphi(\hat{\mathbf{x}}_q)\right)\ \vec{\boldsymbol{\lambda}}\left(\varphi(\hat{\mathbf{x}}_q)\right).\overrightarrow{\nabla \mathscr{Q}_j}\left(\varphi(\hat{\mathbf{x}}_q)\right)\ |J(\hat{\mathbf{x}}_q)|. \end{aligned}$$

If ψ denotes the inverse function of φ, one has

$$\mathscr{Q}_i = \hat{\mathscr{Q}} \circ \psi.$$

Then

$$\overrightarrow{\nabla \mathscr{Q}_i} = J^{-1}.\overrightarrow{\nabla \hat{\mathscr{Q}}_i} \circ \psi,$$

and

$$D_i^{\mathrm{Q}} = h\theta^{\mathrm{Q}} \sum_{q=1}^{(k+1)^2} \sum_{j \in \mathrm{Q}} u_j\ \vec{\boldsymbol{\lambda}}(\mathbf{x}_q).J^{-1}\overrightarrow{\nabla \hat{\mathscr{Q}}_i}(\hat{\mathbf{x}}_q)\ \vec{\boldsymbol{\lambda}}(\mathbf{x}_q).J^{-1}\overrightarrow{\nabla \hat{\mathscr{Q}}_j}(\hat{\mathbf{x}}_q)\ |J|(\hat{\mathbf{x}}_q) \tag{6.1}$$

In practice, we consider that J is constant over $\widehat{\mathrm{Q}}$ and of the same order as $|\mathrm{Q}|$. Then, equation (6.1) is usually computed as

$$D_i^{\mathrm{Q}} = \frac{h\theta^{\mathrm{Q}}}{|\mathrm{Q}|} \sum_{q=1}^{(k+1)^2} \sum_{j \in \mathrm{Q}} u_j \ \vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla \widehat{\mathscr{Q}}_i}(\hat{\mathbf{x}}_q) \ \vec{\boldsymbol{\lambda}}(\mathbf{x}_q).\overrightarrow{\nabla \widehat{\mathscr{Q}}_j}(\hat{\mathbf{x}}_q), \tag{6.2}$$

and the associated Jacobian matrix is obvious.

6.2 Numerical Results

6.2.1 Constant Advection

We start this chapter of results by a very simple scalar case. The domain Ω is the unit square $[0;1]^2$ and the advection is constant and vertical, $\vec{\boldsymbol{\lambda}} = (0,1)$. The problem reads:

$$\begin{cases} \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla u} = 0 \\ u(x,0) = \sin^2(5\pi x) \\ u(0,y) = u(1,y) = 0 \end{cases} \tag{6.3}$$

The values on the upper boundary are let free. The values on the other boundaries are imposed strongly at the beginning of the computation and never updated. The unique solution is obviously the transport of the input function:

$$u^*(x,y) = \sin^2(5\pi x).$$

We have computed this problem on many different grids. The characteristic mesh size h is here the inverse of the number of vertices lying on one boundary (the edges of the domain are homogeneously discretized). For different values of h, we have generated different meshes, ones with triangles only, the other ones being hybrid (contain triangles and quadrangles). The hybrid grids are generated with GMSH [34, 33] using the "recombine" function that combine as much triangles of the triangular grid as possible into convex quadrangles. For each values of h, the triangulation has thus exactly the same number of vertices as the hybrid mesh and we usually have 2 times more elements in the triangulation than in the hybrid mesh. This is summarized in Table 6.1. We are going to study the h-convergence of the *Stabilized Lax-Friedrichs* scheme in triangulation, hybrid mesh and of course compare the efficiency of one approximation with respect to the other.

On Figure 6.2, one can see the coarser hybrid mesh used on the left side, and the isolines of the 4^{th} order solution obtained on the finest hybrid mesh on the right. On Figure 6.3 we have represented the h-convergence curves for the hybrid meshes, the triangular ones and the mean square straight lines for the points corresponding to the hybrid grids. The desired order is met for all the polynomial approximations. In the case of the second order, we obtain indeed a slope of 1.3 which is far from the slope of 2 expected, but we can see that the two points for the two first grids are almost at the same error level. The explanation is simple: the meshes are so coarse that the input function is advected only as far as one or two elements from the bottom boundary and the measured error is roughly the $\mathcal{L}^1$ norm of the exact solution. We can consider that the first point is not relevant for 1^{st} order simulation and if we use just the four

h	**Vertices**	**Triangles**		**Quadrangles**
0.1	114	190	36	77
0.05	468	858	128	365
0.025	1784	3410	480	1465
0.0125	7777	15236	1982	6627
0.01	11454	22510	2858	9826

Table 6.1: Number of vertices, triangles and quadrangles constituting the different meshes used for the grid convergence. The left number in the column **Triangles** corresponds to the number of triangles in the triangular mesh, while the right one is the number of triangles in the hybrid grid. Hybrid grids have then about two times less elements than the triangular twin ones.

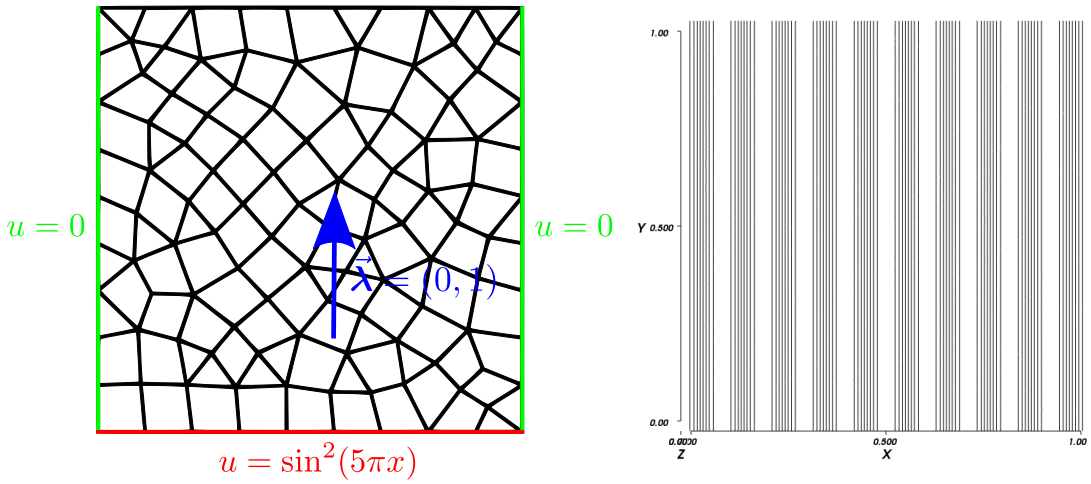


Figure 6.2: Coarser hybrid grid and the 4^{th} order solution obtained on the finest hybrid grid for problem (6.3).

other points, the slope of the mean square straight line is now 1.8, which is far better. Another very interesting remark is that for the same number of vertices and the same sought order of accuracy, the hybrid grid is generally doing a better job. This being true above all for the finest grid ($h \in \{0.0125, 0.01\}$). We explain that the following way: if we consider a convex quadrangle, we can divide it into two triangles. If we make use of a $\mathbb{P}^k$ approximation on the triangles, we are going to add extra DoFs on the edges and inside the triangles. But if we now recombine these two triangles, we obtain exactly the quadrangle with its $\mathbb{Q}^k$ DoFs. And in the case of triangles the approximation of the exact solution is piecewise polynomial of order k, while in the case of the quadrangle, for the same number of DoFs, we have the approximation of polynomial order k, plus the mixed terms coming from the $\mathbb{Q}^k$ framework. Then the global finite dimensional subspace of approximation for the triangular mesh is included in the subspace of approximation for the quadrangular grid, and it is correct that the approximation is better with quadrangles than with triangles.

Finally, one would also like to compare the two simulations in term of computational time. The CPU time (in seconds) needed for 1000 iterations are reported on Table 6.2. The computation on the hybrid grid is almost always faster, except for the 4^{th} order approximation on the

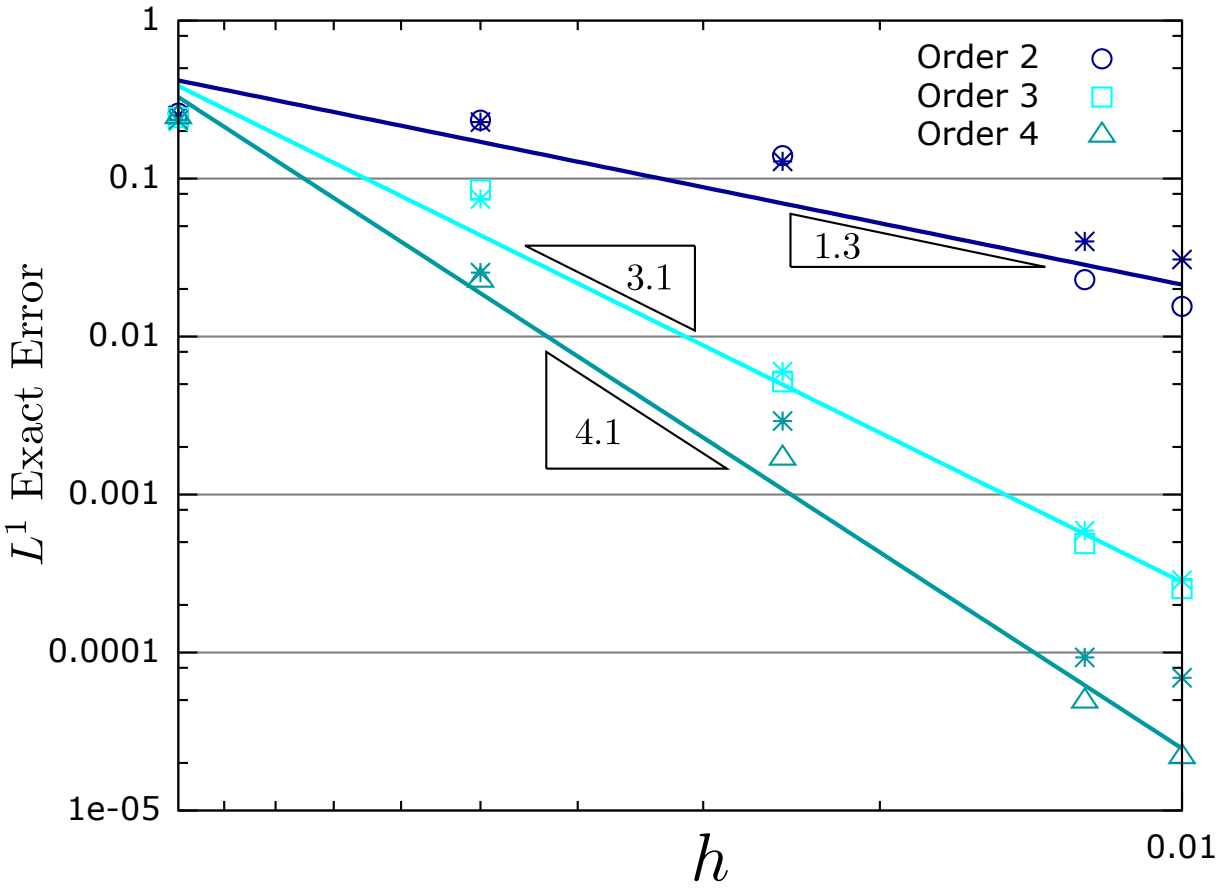


Figure 6.3: Mesh convergence for the simple constant advection problem (6.3). The mean square slope are calculated with the errors measured on the hybrid meshes (represented by circles, squares and triangles). The star points correspond to the same simulations on triangular grids (same problem, same number of vertices).

finest grid. This was expected, as the hybrid grid has roughly two times less elements than its triangular associated mesh. For 2^{nd} order approximation, we have 4 DoFs per quadrangles while triangles have only 3. The ideal speed ratio is then 1.5 which is not so far from the 1.41 obtained. But as soon as we use higher order approximation, we recall that the number of DoFs in a quadrangle is $(k-1)^2$, while there are only $\frac{k(k+1)}{2}$ DoFs in a triangle. Then using higher order approximation brings about 2 times more work in a quadrangle than in a triangle. Plus, the computation of the dissipation term uses all the DoFs in a quadrangle, while it uses only $\frac{k(k-1)}{2}$ degrees of freedom in a triangle. That explains why the speed ratio goes to 1 for larger k, and we are pretty sure this ratio would be smaller than 1 for 5^{th} order approximation. However, the use of quadrangles remains interesting since they give a lower error compared to the one obtained on triangles.

h	$\mathbb{P}^1$	$\mathbb{Q}^1$	$\mathbb{P}^2$	$\mathbb{Q}^2$	$\mathbb{P}^3$	$\mathbb{Q}^3$
0.1	0.286	0.206	0.369	0.31	0.53	0.47
0.05	1.32	0.927	1.68	1.43	2.43	2.24
0.025	5.42	3.76	7.1	6.04	10.55	9.81
0.0125	25.05	17.75	34.02	30.59	51.9	50.78
0.01	37.24	27.06	53.34	46.22	72.95	76.81
	1.41		1.16		1.05	

Table 6.2: Computational time in seconds for 1000 iterations for the different meshes and order of approximation. The last line gives the mean speed ratio for the considered order of approximation.

6.2.2 Circular Advection

Consider a solid body rotation speed $\vec{\boldsymbol{\lambda}} = (-y, x)$, and the resulting inner equation:

$$x\frac{\partial u}{\partial y} - y\frac{\partial u}{\partial x} = \vec{\boldsymbol{\lambda}}.\overrightarrow{\nabla u} = 0. \tag{6.4}$$

We solve this problem on the computational domain $[-1;1]\times[0;1]$. Let $\vec{\mathbf{n}}$ be the outward normal to the boundaries of the domain. It will be useful to classify the boundaries as follows:

- $\vec{\boldsymbol{\lambda}}.\vec{\mathbf{n}} < 0$: the flow is entering the domain along the edge. We say that the edge belongs to Γ^+, the set of the *inflow boundaries*.
- $\vec{\boldsymbol{\lambda}}.\vec{\mathbf{n}} > 0$: the flow is going out of the domain along the edge. We say that the edge belongs to Γ^-, the set of the *outflow boundaries*.

As we have seen in Chapter 2, we need to impose the solution on the *inflow boundaries*, while the solution can be let free on Γ^-.

We are going to use three different types of grids. The domain is divided in two by the straight line $x = 0$. The first grid is a *triangulation* of both sub-domains. This mesh will be called "TriTri". The second one is a *triangulation* of left side combined with a hybrid mesh on the right side. It is called "TriQua". Finally, the last mesh is a hybrid grid on both sides and we call it "QuaQua". These meshes as well as the Γ^+ and Γ^- boundaries are represented on Figure

6.4. For all the test cases, the solution is going to be null outside the disk of radius $\frac{3}{4}$. Then, the advected form will be imposed only on boundary (1) and the value 0 will be maintained on boundaries (2) and (3).

We are going to impose a shape function on boundary (1), with compact support in $[-\frac{3}{4};0]$, and observe the advected function on the output boundary $[0;1]$. We start by the regular function

$$\sin^8\left(\frac{4\pi x}{3}\right), \tag{6.5}$$

on boundary (1). If value 0 is maintained on the other *inflow boundaries*, the exact solution is obviously

$$\begin{cases} \sin^8\left(\frac{4\pi r}{3}\right), & \text{if } r=\sqrt{x^2+y^2}<\frac{3}{4} \\ 0, & \text{else} \end{cases} \tag{6.6}$$

The value of the solution at the degrees of freedom of the output edge $x=[0;1]$ are represented on Figure 6.5 for 2^{nd} and 3^{rd} order simulations. First thing, even if the mesh is rather coarse, the 3^{rd} order simulation gives a very fine result for all the grids. There is no big difference between the meshes in that case. It is much interesting to look at the 2^{nd} order approximation. In all cases, the scheme is diffusive. But what is clear is that the more quadrangles are used in the grid, the less diffusive the output function is. This confirms the remarks made in the previous subsection: the quadrangle approximation uses a wider space of approximation and is then more accurate.

We now consider a discontinuous solution. The input form function on boundary (1) is the characteristic function of interval $[\frac{1}{4};\frac{3}{4}]$, $\xi_{[\frac{1}{4};\frac{3}{4}]}(|x|)$ and the exact solution is given by

$$\begin{cases} 1.0, & \text{if } \frac{1}{4}<r=\sqrt{x^2+y^2}<\frac{3}{4} \\ 0.0, & \text{else} \end{cases} \tag{6.7}$$

The output degrees of freedom are plotted on Figure 6.6. As before, the solutions on grids containing quadrangles are very slightly better. The discontinuities are a bit better resolved. But we have been testing this case above all to check the behaviour of the scheme in presence of discontinuities. As we said in Subsection 5.3.2, the stabilization term destroys the monotonicity preserving property of the LLxF scheme, and we should use a shock capturing function θ to annihilate the effects of this term in the vicinity of discontinuities. Here we have set θ uniformly equal to 1. However, the 2^{nd} order simulation is very good and we can not really see any spurious oscillations. On the 3^{rd} order simulation, we can see that some over- and undershoots appear at points 1, 2 and 3. These oscillations could have been almost completely eliminated with a good shock capturing function θ. But, the global behaviour of the stabilized limited Lax-Friedrichs scheme is rather good, the oscillations are almost insignificant. Eventually, it is important to notice that the formulation on triangles seems to be a bit more stable as the overshoot at point 2 is nonexistent for triangular grid.

6.2.3 Higher Order Efficiency

We now come to the system case. We consider an Eulerian Mach 0.3 flow around a unit sphere. The computational domain is $[-10;10]^2$. It is maybe not big enough, as we are going to see in the following. We have built many different grids for this problem. They are built on the approximation of the sphere boundary with 10, 20, 40, 80 and 100 points respectively.

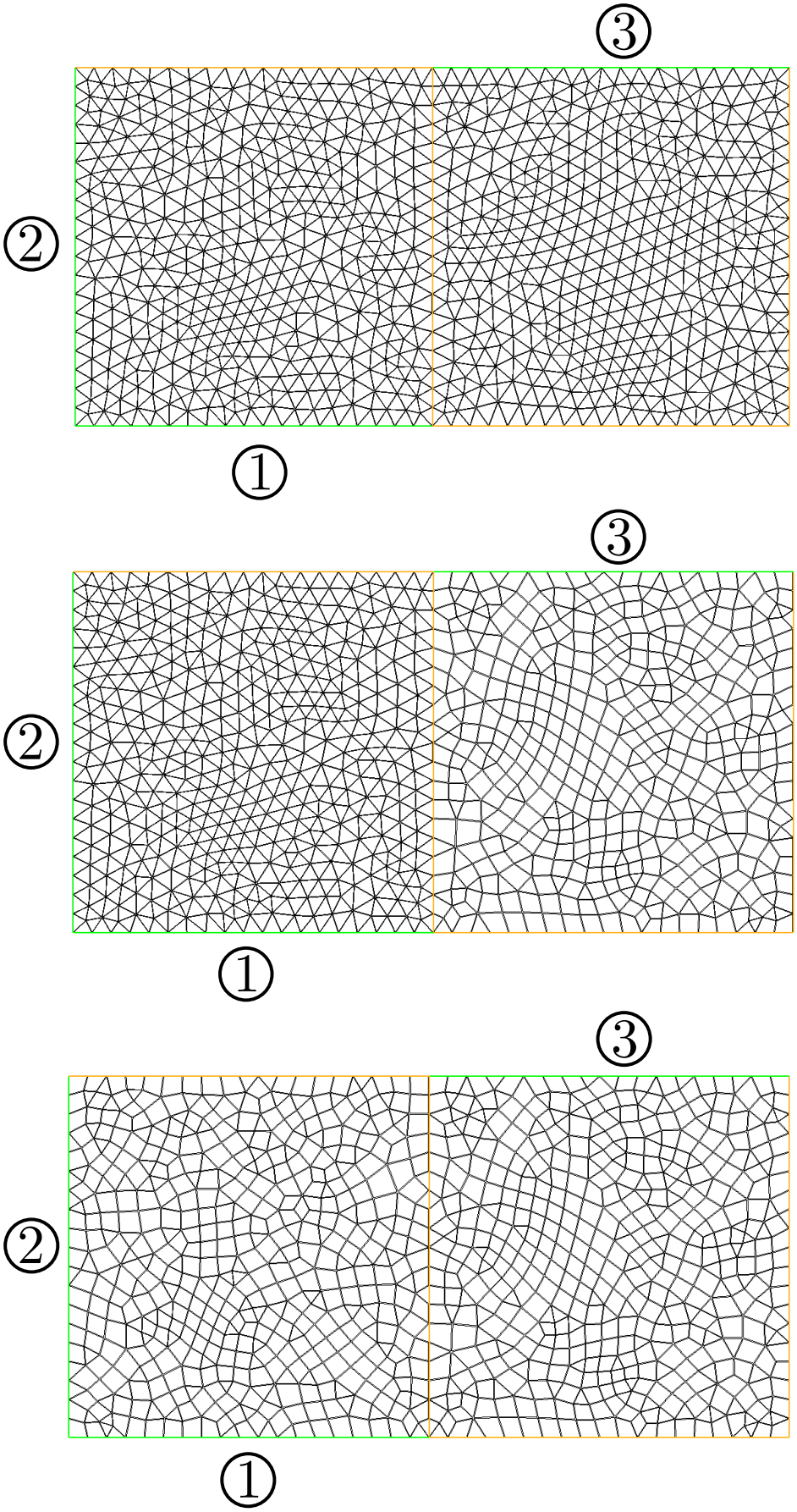


Figure 6.4: "TriTri", "TriQua" and "QuaQua" meshes used for Problem (6.4). The green edges (1), (2) and (3) are the *inflow boundaries*.

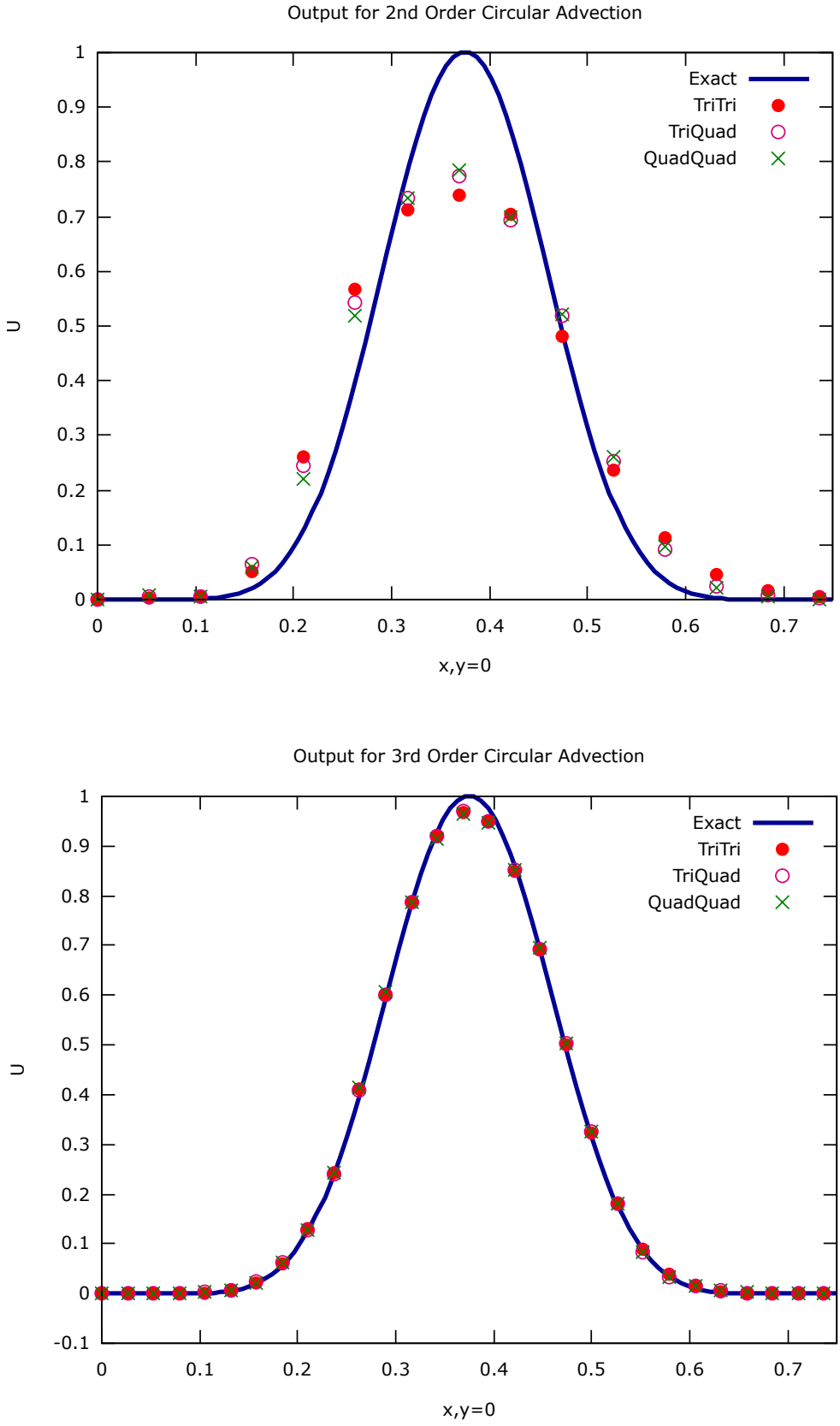


Figure 6.5: Value of the solution at the DoFs situated on the output boundary for 2^{nd} and 3^{rd} order approximation. The input function is given by (6.5).

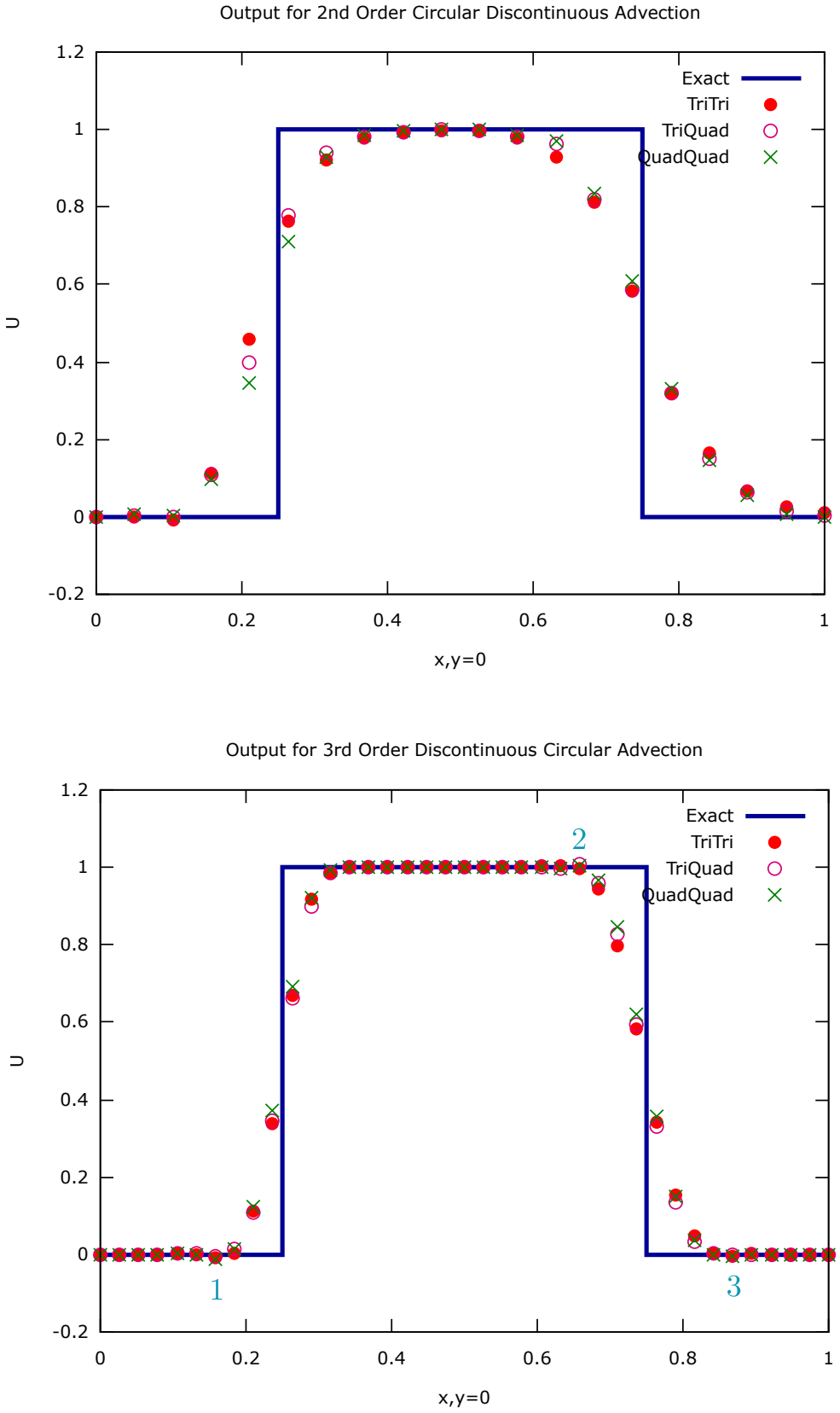


Figure 6.6: Value of the solution at the DoFs situated on the output boundary for 2^{nd} and 3^{rd} order approximation. The input function is $\xi_{[\frac{1}{4};\frac{3}{4}]}(|x|)$.

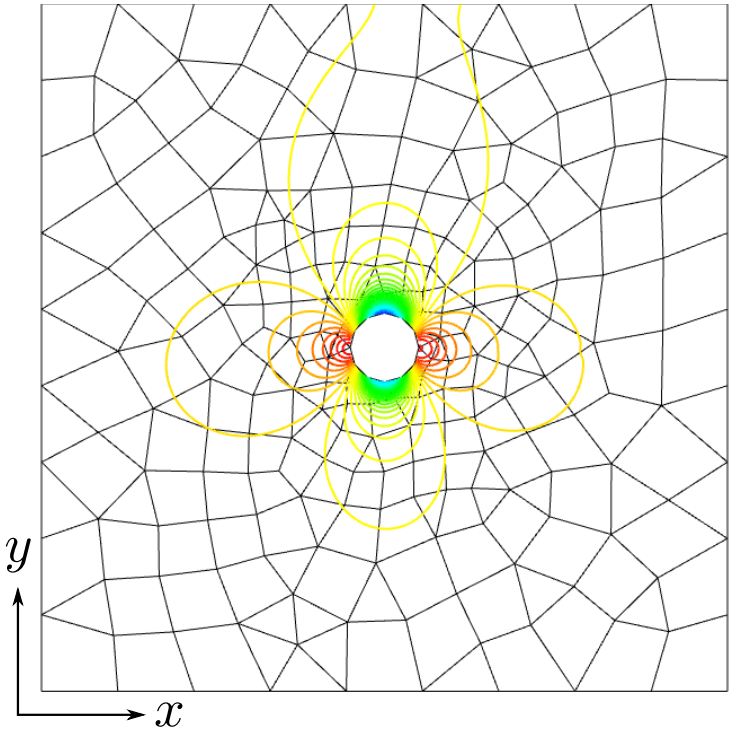


Figure 6.7: Density isolines of the third order solution obtained on the finest hybrid grid represented over the coarser hybrid mesh.

The inner domain is discretized either with triangles only or by a hybrid grid containing mostly quadrangles, such that the hybrid grid has roughly two times less elements than the associated *triangulation*. All of these meshes have been generated thanks to the free software GMSH [33, 34] developed by Christophe Geuzaine (University of Liège) and Jean-Francois Remacle (Catholic University of Louvain). 2^{nd} and 3^{rd} order simulations have been computed on these meshes.

The coarser hybrid grid as well as the isolines of the solution obtained with the finest hybrid grid with a 3^{rd} order scheme are given on Figure 6.7. From the isolines, we can see that the solution is not perfect, especially on the rear of the cylinder. Even if the test case should be isentropic, numerical entropy is created on the boundaries, mostly at the stagnation point and it spoils the solution elsewhere. We will see further that this is actually a way of measuring the order of accuracy of the used numerical scheme. Another way is by measuring the global lift or drag around the sphere. As one can see on Figure 6.7, the mesh has no symmetry and the solution is then not going to be symmetrical around the sphere boundaries. We can then measure the numerical lift coefficient as the contour integral of the *pressure* around the sphere boundary:

$$\begin{aligned} C_l &= \int_{\partial \text{sphere}} p\, \vec{\mathbf{n}}.\vec{\mathbf{dy}} \\ &= \sum_{\text{edges}} \frac{p_A + p_B}{2}\, \vec{\mathbf{n}}^{\text{edge}}.\vec{\mathbf{dy}}, \qquad \text{in the } \mathbb{P}^1 \text{ case} \\ &= \sum_{\text{edges}} \frac{p_A + 4p_C + p_B}{6}\, \vec{\mathbf{n}}^{\text{edge}}.\vec{\mathbf{dy}}, \qquad \text{in the } \mathbb{P}^2 \text{ case.} \end{aligned}$$

Here p_A, p_B and p_C stand for the values of the pressure at both ends and in the middle of the edge respectively. This lift coefficient should converge toward zero as the mesh gets finer. We can see the convergence curves on Figure 6.8. The two different colors denote the two different orders of representation of the data. The circle and square points are the lift coefficients obtained with the hybrid grids. All the points represented by stars are the lift coefficients corresponding to the triangular meshes. Finally, the lines are the mean square straight lines for the set of points obtained with the hybrid grids. For second order simulation, the result matches exactly what have been observed on scalar problems: the slope of the mean square straight line is almost the perfect one and the quadrangular result is always slightly better than the triangular one. Conversely, the result obtained with the 3^{rd} order code is less clear. There are several reasons for this. First of all, we note that the simulations on the hybrid grids are globally worse than the ones on the *triangulations*. But for these 3^{rd} order computations, the iterative convergence have not been reached. We have represented the iterative convergence curves for 2^{nd} and 3^{rd} order simulations on Figure 6.9. The scheme is implicit with first order Jacobians. Whereas all the simulations of second order of accuracy converge to 10^{-12}, the third order computations refuse to converge lower than 10^{-5}. We cannot explain why at that moment, but we hope we are just facing an implementation error in the code. What we are however sure of, is that this lack of iterative convergence influences the lift convergence because the steady state is not fully reached. We also observe on Figure 6.8 that the slope of the 3^{rd} order computations is a bit far from the expected one. Even though it is still better than the 2^{nd} order one. Indeed, the lack of iterative convergence could explain this, but there is another thing: in all these calculations, the boundary are represented linearly. The edges are straight lines and the lift coefficient is a parameter that is local to the boundary. Even if the scheme is third order accurate inside the domain, its accuracy could be locally reduced. The obtained slope is a combination of the third order expected accuracy and the second order accuracy of the boundary representation. We will see further that not only the solution can be represented with higher order, but also the edges of the mesh. We call this representation *isoparametrical*.

Concerning the entropy production, we can see on Figure 6.10 that some entropy calculated as

$$s = \ln\left(\frac{p}{p_\infty}\right) - \gamma \ln\left(\frac{\rho}{\rho_\infty}\right) \tag{6.8}$$

is created in the vicinity of the sphere. On this figure, we have represented the isolines of entropy generated with second order scheme (top left), with third order scheme (bottom) and with the second order scheme applied on the degrees of freedom of the third order scheme (top right). It is clear that the creation of numerical entropy is much reduced by using the third order scheme, even when comparing with the mesh having the same number of DoFs. The quality of the solution is indeed also improved. It is even much more interesting to give a look at the computational time. For the triangular grid, the second order simulation has a cost of about $0.96s$ per iteration, the third order one $3.36s$ per iteration and the second order simulation on the third order DoFs about $3.44s$ per iteration. This is expected because each element of the second order mesh is split into 4 to obtain the second order mesh equivalent to the third order one. The 3^{rd} order simulation computationally costs about the same than the second order simulation on the split mesh, but the obtained result is much more accurate.

This subsection has shown the high order formulation is doing a very good job on the hybrid meshes, as it is seriously improving the solution for about the same computational cost. Unfortunately, we are not observing the expected mesh convergence slope, this being essentially due to the linear representation of the boundary. The next subsection is dealing with a higher order

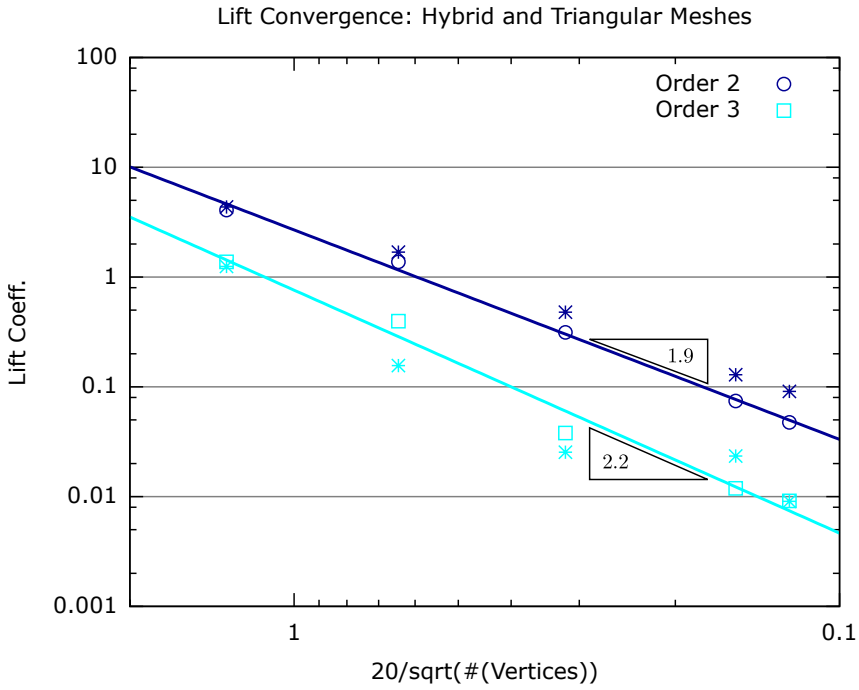


Figure 6.8: Convergence of the lift coefficient. Each color denotes an order of accuracy, stars are the triangular grids, circles and squares the hybrid ones and lines are the mean square straight lines of the circles and squares set.

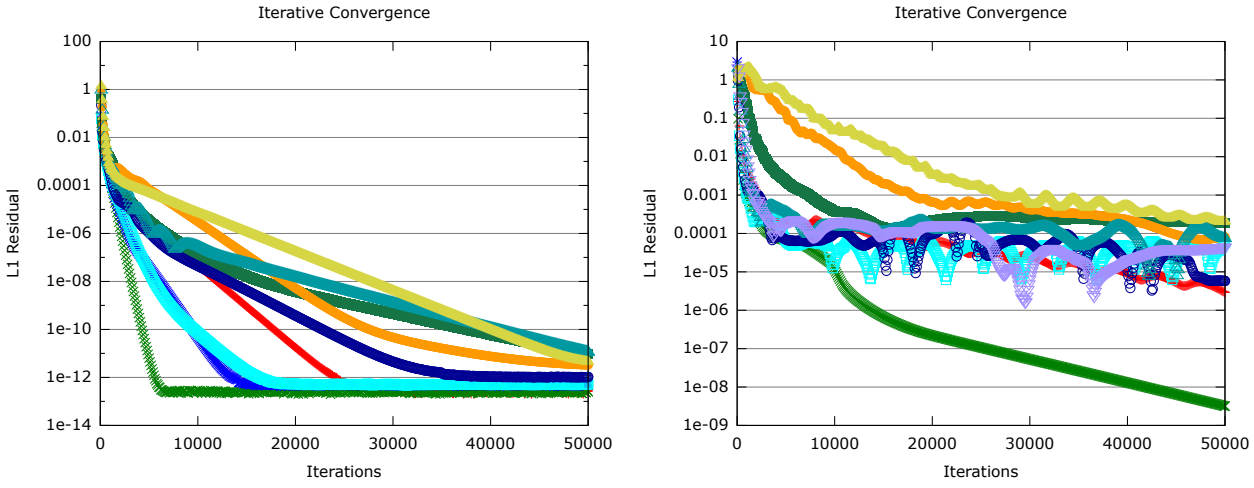


Figure 6.9: Iterative convergence for all the meshes of the sphere problem. On the left are the iterative curves of the second order simulation whereas the right figure corresponds to the third order ones.

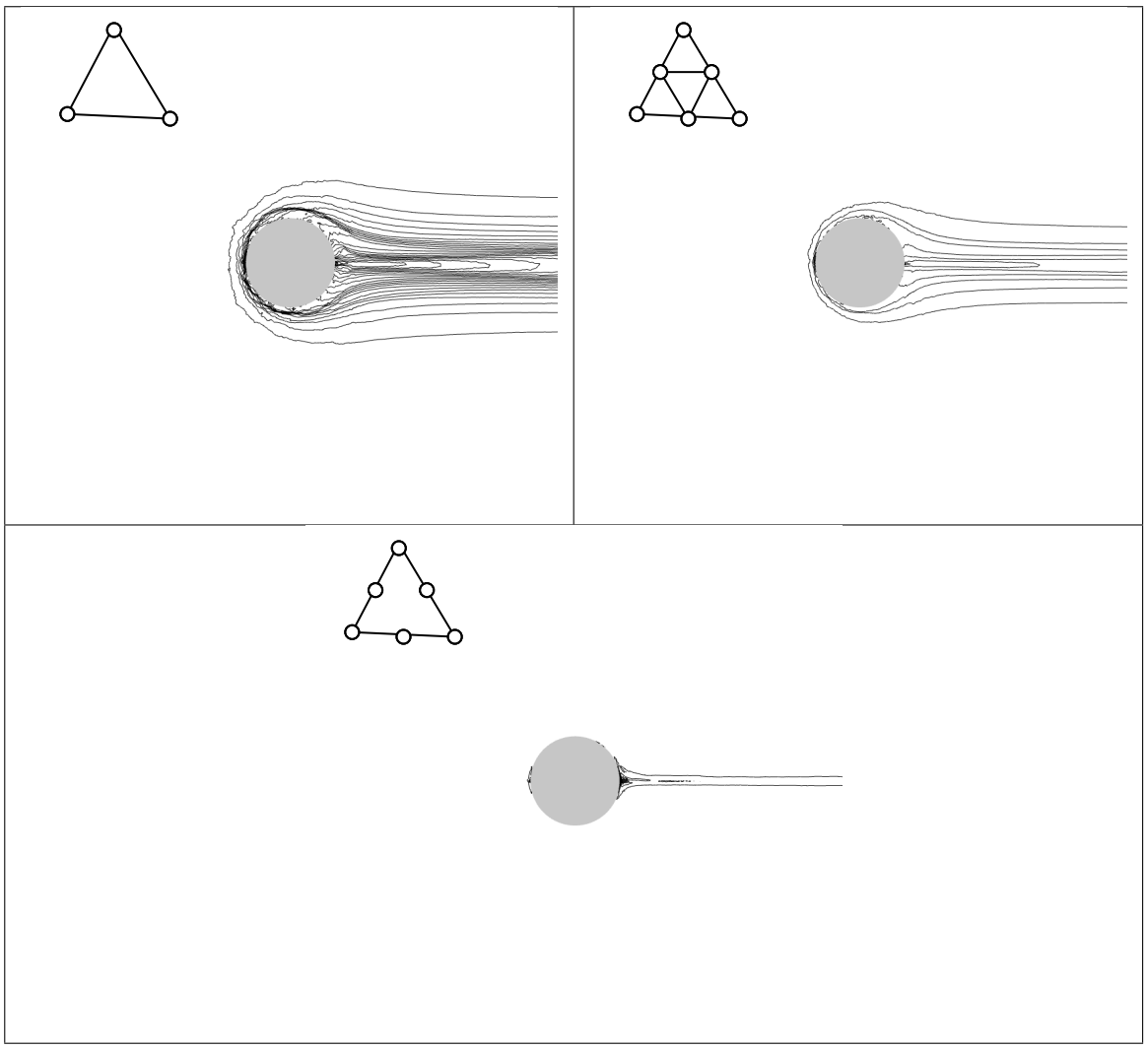


Figure 6.10: Same 60 isolines of created numerical entropy for second order scheme (up-left), second order scheme on the third order sub-triangulated mesh (up-right) and third order scheme (below).

of representation of the boundary edges. We will see how this improves even more the computed solution but, for the time being does not give exactly the expected results.

6.2.4 Isoparametrical Elements

The goal is here to represent the boundary edges with quadratic parametric curves. Let us consider two neighbour vertices of the boundary, A and B, and the real boundary edge linking them. Until now we have been approximating this curved edge by the segment [AB]. We want to improve this approximation. To do that, we look for three vectors of size $n = 2$, the number of spatial dimensions, such that the parametric curve

$$\mathbf{X}(t) = \vec{a}t^2 + \vec{b}t + \vec{c} \tag{6.9}$$

is an approximation of order $\mathcal{O}\left(\|\mathrm{AB}\|^2\right)$ of the real boundary. We then have 6 degrees of freedom to define the new edge. First, we have to ensure the curve passes through A and B. We thus redefine (6.9) as a Bézier curve and we get:

$$\mathbf{X}(t) = \mathbf{X}_\mathrm{A}(1-t)^2 + \mathbf{X}_\mathrm{B}t^2 + \vec{c}t(1-t), \tag{6.10}$$

where $\mathbf{X}_\mathrm{A}$ and $\mathbf{X}_\mathrm{B}$ are the coordinates of A and B respectively and $\vec{c}$ is two extra degrees of freedom.

One could choose and extra point on the real edge (for example the orthogonal projection C$'$ of C the middle of [AB]) and compute $\vec{c}$ such that the quadratic approximation of the curve also passes through C$'$. Unfortunately, because we do not impose the direction of the derivatives at A and B, the global reconstructed boundary profile is not $\mathcal{C}^1$ and we even observe some Gibbs phenomenon in the region where the curvature of the profile is strong (the stagnation point of an airfoil for example). We have done several tests with this configuration, and the quality of the global solution is not improved at all and even deteriorated at some times. The idea is then to impose just the exact direction of the first derivatives at A and B (2 extra constraints) and to take the middle edge C* as

$$\mathrm{C}^* = \mathbf{X}\left(\frac{1}{2}\right). \tag{6.11}$$

If $\vec{V_\mathrm{A}}$ and $\vec{V_\mathrm{B}}$ are two exact tangents to the boundary profile at points A and B, whatever be their norms, the last coefficient of equation (6.10) is the unique solution of

$$\begin{cases} \vec{c} \wedge \vec{V_\mathrm{A}} &= 2\mathbf{X}_\mathrm{A} \wedge \vec{V_\mathrm{A}} \\ \vec{c} \wedge \vec{V_\mathrm{B}} &= 2\mathbf{X}_\mathrm{B} \wedge \vec{V_\mathrm{B}} \end{cases} \tag{6.12}$$

Even if only the edges of the boundary are modified, we consider that the whole mesh is isoparametric, and we have to redefine all the quantity that have been calculated previously in the case of "straight triangles". The global residual is still calculated as a contour integral, but because both the flux and the normal to the edges are quadratic functions of the coordinates, the integrand is of order 4. The classical Simpsons rule does not integrate the global residual exactly anymore. We then use Gauss quadrature points and integrate the residual just by evaluating the needed quantities at the quadrature points. This is what have been done in the following simulation. However, by experience, using Simpsons rule does not seem to destroy the accuracy. It is then a cheaper solution as it does not require to reconstruct the unknowns at

the quadrature points. For the dissipation term, the reasoning is the same than in Subsection 5.3.3. The accuracy of the scheme is always maintained and the term is dissipative if and only if we have enough quadrature points to define the gradients a unique way. Equation (5.55) is still valid, but the gradients of the basis functions are different and have to be recomputed. Finally, the slip wall boundary contribution on the sphere edge is calculated as (5.63), with a 4^{th} order quadrature because once more the boundary fluxes and the normals are quadratic functions of the coordinates.

We have plotted on Figure 6.11 the same entropy contours for the second order, the third order and the third order with parametric boundaries solutions as well as the lift convergence curve. For the entropy isolines, the result is pretty clear: compared to second order, the third order simulation reduces the numerical entropy production, even more when using the isoparametric representation of the boundaries. In the last case, the entropy production is almost insignificant compared to the $\mathbb{P}^1$ computation. Unfortunately, things do not improve as far as the convergence of the lift coefficient is concerned. The 3^{rd} order slope is not reached as expected, and the slope of the mean square straight line is even worse than in the case of the linear representation of the boundaries. However, except for the finest grid, all the point for the isoparametrical simulation are situated beneath those of the previous 3^{rd} order simulation. As in the case of the linear representation of the boundaries, the scheme has not fully converged, and this may be due to a lack of maturity of the hybrid scheme.

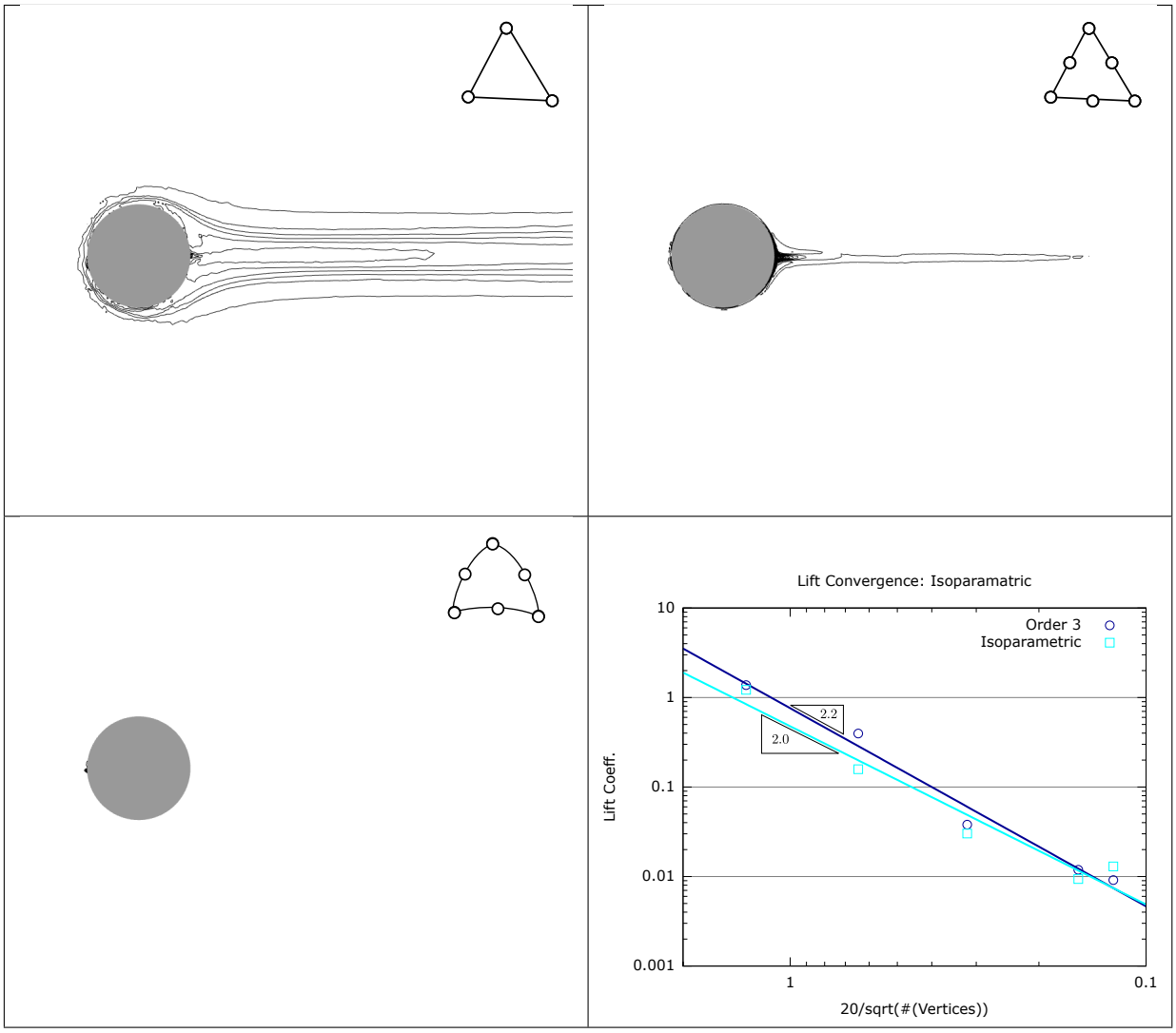


Figure 6.11: Entropy isolines and lift coefficient convergence for the sphere problem. Up-Left figure is the entropy contours for second order simulation, Up-Right is for third order simulation with linear representation of the boundaries. Down-Left is for third order simulation with isoparametrical elements. Each of these figures represents the same 50 levels of isolines. Finally, the down right figure compares the lift coefficient between the linear and the isoparametrical representation of the boundaries.

Chapter 7

3D Simulations

This chapter is devoted to the simulation of the Euler equation in three dimensions. Even if we are going to treat only steady Euler test cases, we first start by generalizing the construction of the unsteady Navier-Stokes system done for a two dimensional domain in Section 2.2. The three dimensional steady Euler system is obtained by ignoring the time dependent terms and remove the viscous effects. The speed has now three components u, v and w and the vector of unknowns is

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}. \tag{7.1}$$

The three dimensional unsteady Navier-Stokes equations read:

$$\frac{\partial \mathbf{U}}{\partial t} + \operatorname{div}\left(\vec{\boldsymbol{\mathcal{F}}}(\mathbf{U})\right) = \left(K_{ij}\mathbf{U}_{,j}\right)_{,i} = \operatorname{div}\left(\mathbb{K}.\overrightarrow{\nabla \mathbf{U}}\right). \tag{7.2}$$

where, using $\boldsymbol{\delta}_i$ to denote the i^{th} column of the 3×3 identity matrix,

$$\vec{\boldsymbol{\mathcal{F}}} = (\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3), \quad \mathbf{F}_i = \begin{pmatrix} \rho u_i \\ \rho u_i \vec{\mathbf{u}} + p\boldsymbol{\delta}_i \\ (\rho E + p)u_i \end{pmatrix}, \ i = 1 \ldots 3$$

is the *advection flux* and $\mathbb{K}$ is a $d \times d$ diffusive matrix of $m \times m$ ($m = d + 2$) matrices that are detailed in Appendix A. In Appendix B we have also reported the Jacobians of the advective flux $\mathrm{A} = \dfrac{\partial \mathbf{F}_1}{\partial \mathbf{U}}$, $\mathrm{B} = \dfrac{\partial \mathbf{F}_2}{\partial \mathbf{U}}$ and $\mathrm{C} = \dfrac{\partial \mathbf{F}_3}{\partial \mathbf{U}}$. The diagonalization of the 3D advection speed in any direction $\vec{\mathbf{n}}$ is also given. The left and right eigenvectors as well as the eigenvalues are needed for example to define the limitation over the characteristic components of the residual.

3D computations are much more complex compared to the 2D ones. First of all, the result is harder to analyze. It is much more complicated to find a local irregularity (for example a problem on the boundary) in a three dimensional solution than in a 2D one. In 2D, one can represent and see all the points of the domain globally. But in 3D, the only thing we can watch are slices of the solution. In a second time, it is really much easier to reach the limit of a processor capacity with a 3D computation. It is not uncommon that a node has 100 neighbours

in a $\mathbb{P}^2$ simulation on tetrahedra. Then each line of the matrix needs about 40kBytes of RAM. Multiplied by the approximately $3n$ DoFs (n being the number of vertices), this represents $0.1n$ MBytes to load just the matrix of the linear system in the RAM of the computer. Then if n is larger than 10^5 the computation cannot be done on a single processor. In order to distribute this memory load between several processors, we have been developing a parallelized version of the code. We make here a small parenthesis to present the implementation and the performances of the parallelization of the $\mathcal{RD}$ schemes.

7.1 Parallelization

Parallel computing is a form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved concurrently ("in parallel") [11]. In our case, one of the good feature of the Residual Distribution Schemes is they are *compact*. That means that at each time step, the value of a degree of freedom is updated using only the values of its direct neighbours (the DoFs sharing the same elements). If we have the possibility to use n processors, we can then divide the mesh into n load balanced sub-domains (containing approximately the same number of DoFs) and ask to each of the processors to update the values of the DoFs of one single domain only. We will call *inner degrees of freedom*, the set of DoFs of a sub-domain whose direct neighbours are all lying in this sub-domain. For these DoFs, their values can be updated independently of the values of the DoFs of the other sub-domains. As we said in the beginning: "they are solved concurrently". The problem comes from the DoFs lying on the vicinity of the edge of each sub-domain. For these nodes, the processors have to share some data in order their values are correctly updated. If this is not done a smart enough manner, the computation is certainly not going to be n time faster, which is one of the main goals of the parallelization. For example, if we do the so called *synchronized parallelization*, each processor waits for the others when he is done with his task, and the memory sharing is realized only when all the processors have finished their computing. This is not an efficient technique at all. In fact, the size of the problem is usually very big compared to the number of processors available. This means that the number of *inner degrees of freedom* is very large compared to the quantity of data the processor has to share. Then, one can renumber the elements of the sub-domains such that the elements having a node on the edge of the sub-domain have the larger number. When the processor starts the iteration, it can simultaneously update the values of the *inner degrees of freedom* and share the needed updated values (during the previous iteration). This is possible because on modern processors, the algebra unit is always separated from the communication one. This technique is called the *asynchronized parallelization* and provide a much better *speedup*.

7.1.1 Domain Decomposition

For the domain decomposition, we have been using Scotch, which is a *"Software package and libraries for sequential and parallel graph partitioning, static mapping, and sparse matrix block ordering, and sequential mesh and hypergraph partitioning"* [5], developed at INRIA Bordeaux Sud-Ouest by François Pellegrini [77, 78, 79]. It is available under the CeCILL-C free/libre

[5] `http://www.labri.fr/perso/pelegrin/scotch/scotch_en.html`

software license [29], which has basically the same features as the GNU LGPL (*"Lesser General Public License"*). The main characteristics of Scotch for domain decomposition are the following:

- Balance of the computation load across processors,
- Minimization of the inter-processor communication cost,
- Treatment in $\mathcal{O}(n_{\text{edges}})$.

As we have seen in the previous section the load balancing is a very important step. During a computation, it is not really to be desired some processor has one or more iterations in advance compared to the others. To prevent such a situation, we still have to synchronize all the processors at the end of an iteration. If the load balancing is well done, the computational cost of such a procedure is negligible. But it is costly when a processor is much slower than the others. In this case, all the processors are going to compute globally at the same speed as the slowest one. The quality of the domain decomposition is also quantified by the inter-processor communication cost. This results from the exchange between the processors of the values lying on DoFs whose direct neighbours are not all in the same domain. Because the $\mathcal{RDS}$ are *compact*, all these special DoFs are situated in a stripe which width does not exceed one element. We will call this region the *overlap*. Then minimizing the inter-processor communication cost is equivalent to minimize the number of DoFs situated in the *overlap*, which can be simply done by minimizing the length of the separating surface between the domains.

In a first attempt of parallelization, we have not chosen a good solution, though. We have decomposed the mesh element by element, and balanced the processors load by taking into account only the vertices of the mesh. This is not the best choice as soon as we want to execute a higher order simulation, because we were generating the higher order mesh on the already decomposed domain. Nothing ensures the load balancing is maintained and it is pretty sure there exist splitting ways using some extra DoFs that minimize much better the overlapping areas. Thanks to the work of Cédric Lachat, during his Master degree internship at INRIA Bordeaux, we are today first generating the higher order mesh and only then do the domain decomposition with Scotch. However, this work is too recent and all the results presented in this chapter are using the previous solution. That is also why the next Subsection about the *overlap* treatment assumes that the domain decomposition has been done on the first order mesh.

7.1.2 Overlap Treatment

All the arguments of this section are illustrated on Figures 7.3, 7.4 and 7.5. Let us first give a look at Figure 7.3. We have two domains, one blue, one red, each one of them belonging to a different processor that will be called simply the blue and red processor respectively. The mesh is $\mathbb{P}^2$ and all the degrees of freedom lying on the splitting way belong to the blue processor. In order to update well their values, the blue processor has to know the values of all its direct neighbours. In particular, it has to know the values of the green DoFs (see Figure 7.4), that belongs actually to the red processor. The same thing on the red side, see Figure 7.5. To update correctly the values of the nodes situated at a distance of less than one element from the separating edges, the red processor has to know the good values of the nodes lying on the separating edges. Then the blue domain is extended by one element width and the red one is extended by the separating edges. However, the values of these green ghosts nodes are not updated at all in the associated

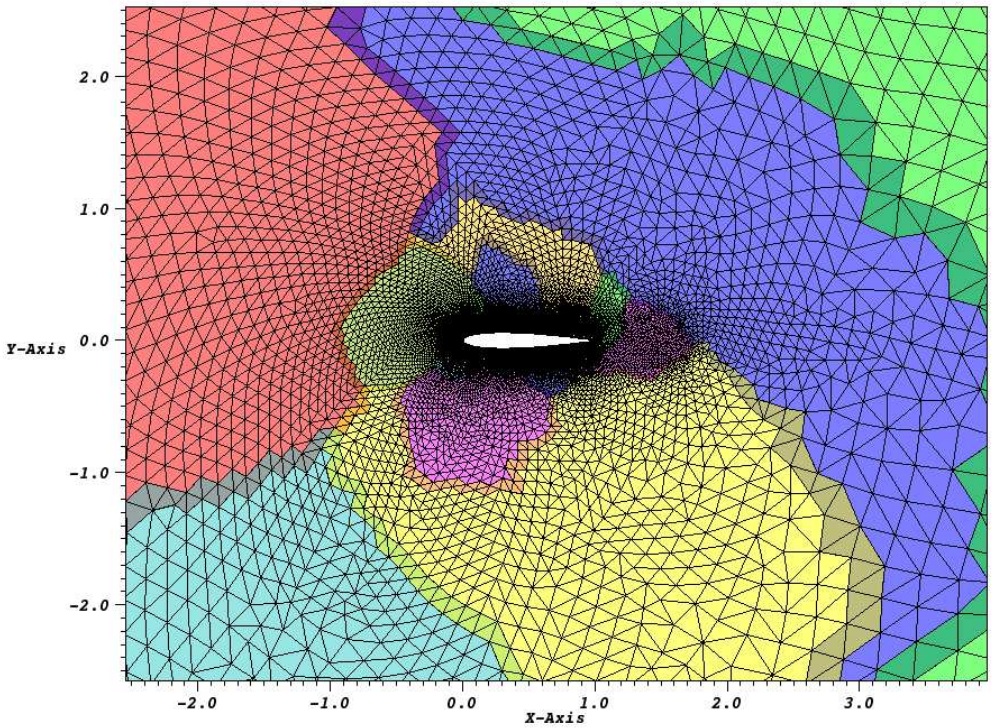


Figure 7.1: A example of a domain decomposition on 16 processors for a subsonic NACA012 mesh.

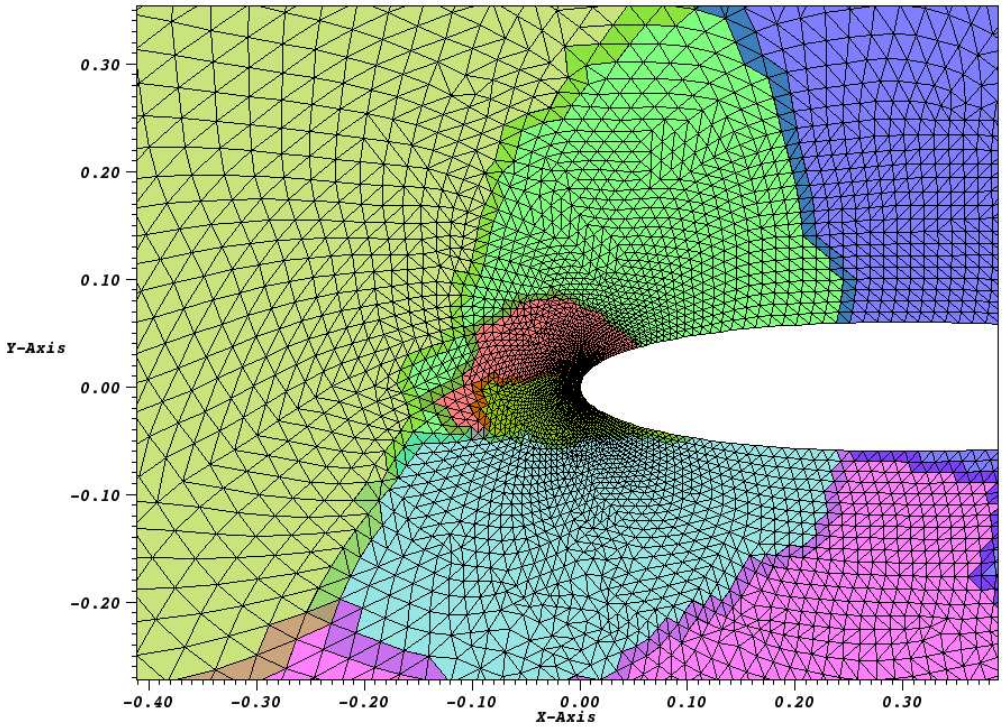


Figure 7.2: Detail around the stagnation point of the upper figure.

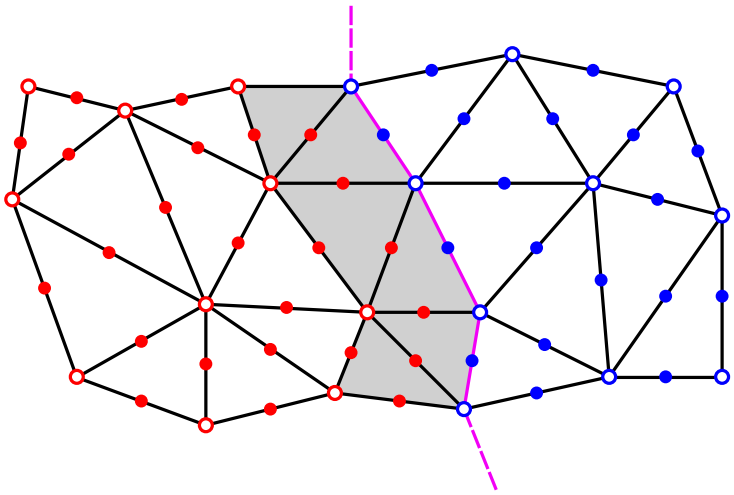


Figure 7.3: Detail of an overlap on a $\mathbb{P}^2$ mesh. The blue degrees of freedom belong to the blue processor while the red ones belong to the red processor. The gray area represents the overlap band. All the values within this band will have to be exchanged between the two neighbour processors.

domain and the processors have to exchange their values during each iterations, otherwise the computation would be wrong.

What is done in practice is that during the domain decomposition, the larger indices are given to the elements having at least one node in the *overlap*. Then, at the beginning of each iteration, the values on the ghosts DoFs have not been update in each domain, and they are thus wrong. But each processor starting by the elements having the smaller indices, these values are not needed at the beginning. Usually the number of *inner nodes* being very large compared to the number of nodes in the *overlap*, the processors have a sufficient amount of time to communicate to their neighbours processors the right values of their ghosts DoFs. This is possible because the calculation units of the CPUs can work separately from the communication units.

7.1.3 Speedup Analysis

They are two main advantages to the parallelization. The first one is to distribute the global computation load homogeneously between the available processors. The second one is to seriously accelerate the simulation. Ideally, if all the communications are executed behind the *inner nodes* computation, the simulation time should be divided by n, the number of processors. We have represented on Figure 7.6 the computational acceleration (also called *speedup*) brought by 2,4,8,16 and 32 processors. As one can see, the speedup curve is far from the ideal one and here follows the explanation.

First of all, as we have already said, the domain decomposition is not necessarily well load balanced, because the splitting have been done on the first order mesh. We represent on Table 7.1 the load unbalance measured by:

$$100\frac{n_{\max} - n_{\min}}{n_{\max}},$$

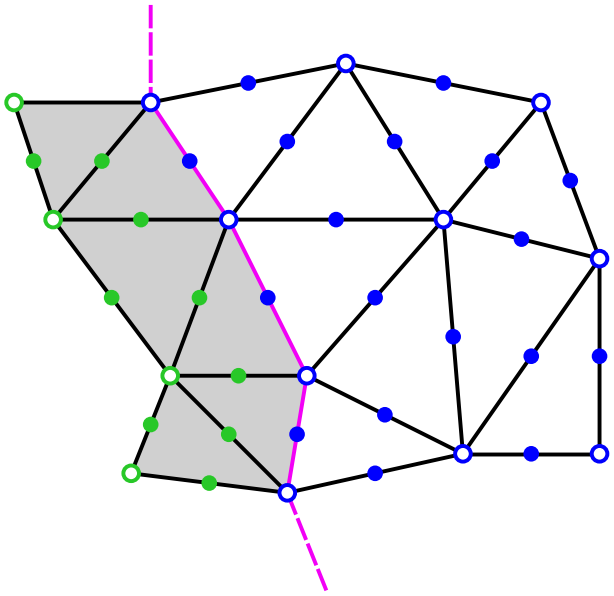


Figure 7.4: Blue processor computational domain. The blue degrees of freedom are the updated values. The green ones are the ghosts nodes needed to update the values of the blue points correctly.

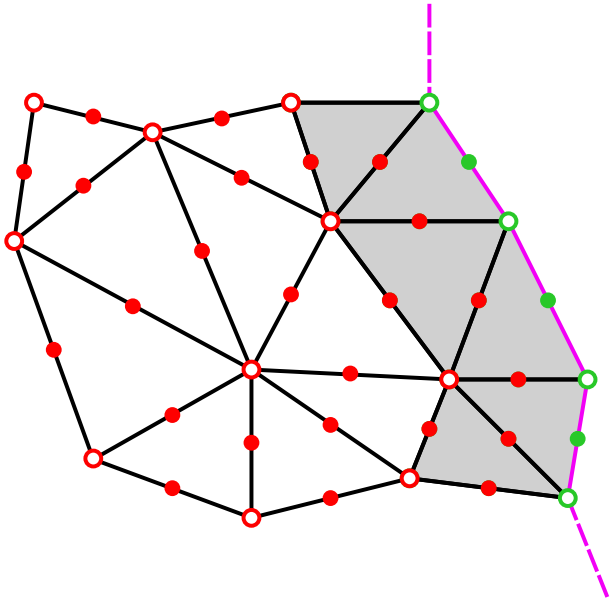


Figure 7.5: Red processor computational domain. The red degrees of freedom are the updated values. The green ones lying on the separating edges are the ghosts nodes needed to update the values of the red points correctly.

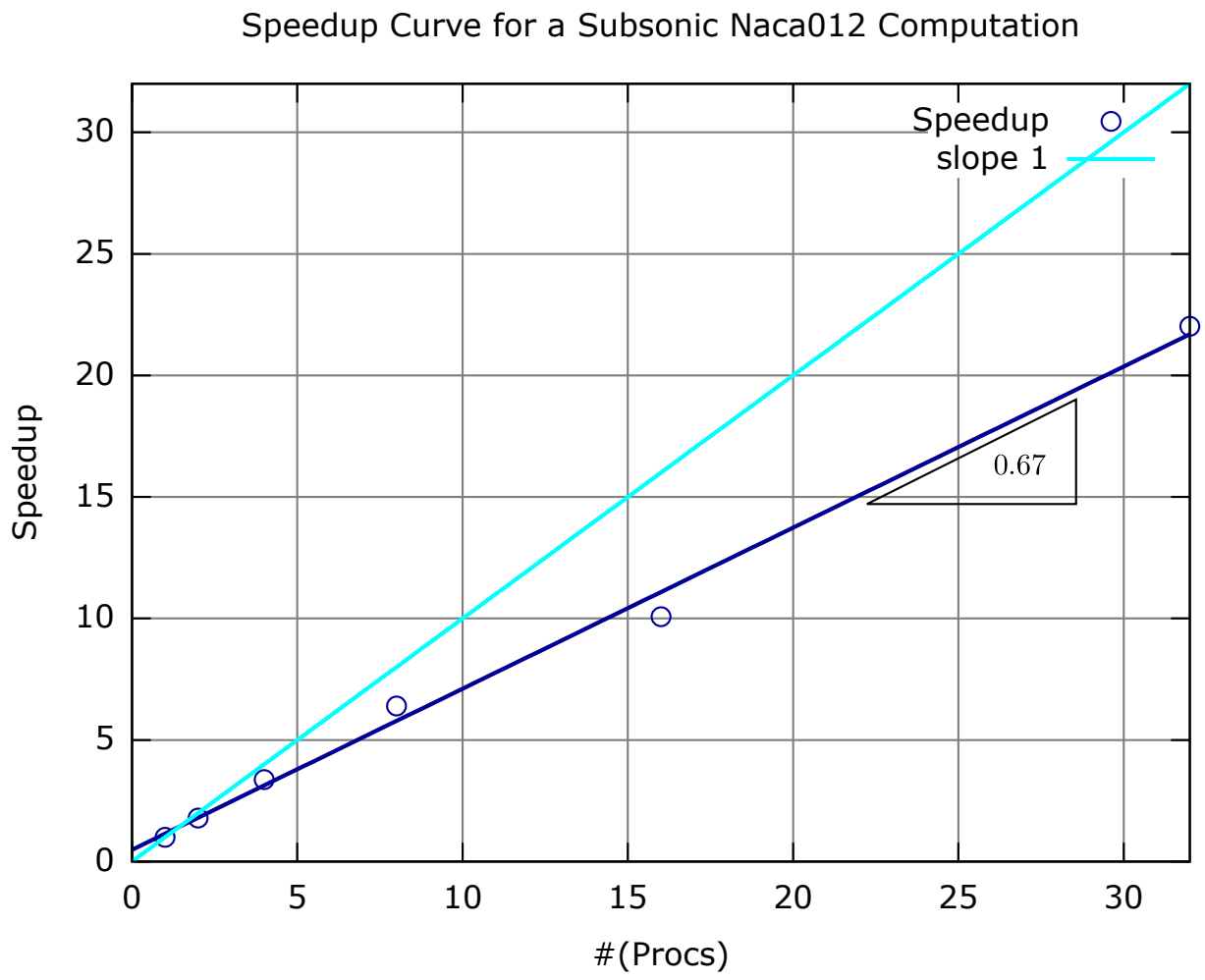


Figure 7.6: Speedup curve for 1,2,4,8,16 and 32 processors on a 0.5 fine $\mathbb{P}^2$ NACA012 simulation.

where $n_{\max}$ (resp. $n_{\min}$) is the maximal (resp. minimal) number of DoFs in a domain. As one

Procs	2	4	8	16	32
$\mathbb{P}^1$	1.7%	1.8%	8%	25%	42%
$\mathbb{P}^2$	3.8%	3.9%	9%	26%	47%

Table 7.1: Load unbalance for $\mathbb{P}^1$ and $\mathbb{P}^2$ simulations on a rather coarse mesh.

can see, the load balance for the $\mathbb{P}^2$ simulations is always worst than for the $\mathbb{P}^1$ ones. Plus, this mesh has only 11000 nodes. A 32 processors parallel simulation is not relevant at all: the load balance is bad and it is pretty much sure that the overlap communication time is not negligible anymore, compared to the inner domain computational time. This is the reason why the speedup curve bends when more processors are used. If one looks at the speedup curve for the $\mathbb{P}^1$, one can see that the speedup rate is very close to 1 for 2 and 4 processors, when the load balance is correct. As soon as the load unbalance exceeds 5%, we can see that on the curve. Even if we are doing *asynchronized parallelization*, we still have to wait for all the processors to finish the ongoing iteration prior to begin a new one. Then, the simulation is globally going at the speed of the most loaded processor.

We finish this section just by saying there is still much to do in this domain. For example, Discontinuous Galerkin methods which are also *maximum compact* and benefit parallelization since a couple more years than the $\mathcal{RDS}$, claim speedup rates oscillating between 0.98 and 0.99 on big enough problems.

7.2 3D Formulation

As we have already seen in the previous chapter, the 3D formulation has been developed so far only for tetrahedra. All the elements are thus tetrahedra, still denoted by T. In the system case, we have not tested polynomial approximation of higher order than 2. We first start this section by giving the numbering convention inside each tetrahedron for $\mathbb{P}^1$ and $\mathbb{P}^2$ formulation. On Figure 7.7 are given the numbering convention of the DoFs, the faces, the edges and, if needed, the sub-tetrahedra. Similarly to the 2D case, it will be useful to consider that for $i = 1, \ldots, 4$, $\vec{\mathbf{n}}_i$ is the normal to the face opposite to vertex i, pointing toward i and which length is scaled by the area of its associated face. For extra DoFs, we use the following convention: for $i = 5, \ldots, 10$, $\vec{\mathbf{n}}_i$ is the opposite of the sum of the two normals associated to the vertices which are not the extremity of the edge on which lies DoF i. If one look at Figure 7.7, we have:

$$\begin{aligned} \vec{\mathbf{n}}_5 &= -(\vec{\mathbf{n}}_3 + \vec{\mathbf{n}}_4),\ \vec{\mathbf{n}}_6 = -(\vec{\mathbf{n}}_1 + \vec{\mathbf{n}}_4),\ \vec{\mathbf{n}}_7 = -(\vec{\mathbf{n}}_2 + \vec{\mathbf{n}}_4), \\ \vec{\mathbf{n}}_8 &= -(\vec{\mathbf{n}}_2 + \vec{\mathbf{n}}_3),\ \vec{\mathbf{n}}_9 = -(\vec{\mathbf{n}}_1 + \vec{\mathbf{n}}_3),\ \vec{\mathbf{n}}_{10} = -(\vec{\mathbf{n}}_1 + \vec{\mathbf{n}}_2). \end{aligned}$$

There is a good explanation to such a convention. We need to compute the *global residual* of the tetrahedron, and as we did before, we compute it on the external envelop of the element. The 3D integral is split into four 2D ones:

$$\begin{aligned} \Phi^{\mathrm{T}} &= \int_{\mathrm{T}} \operatorname{div}\left(\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\mathbf{U})\right) d\mathbf{x} = \int_{\partial \mathrm{T}} \overrightarrow{\boldsymbol{\mathcal{F}}_h}(\mathbf{U}).\vec{\mathbf{n}} \\ &= \sum_{\text{face}} \left(\int_{\text{face}} \overrightarrow{\boldsymbol{\mathcal{F}}_h}(\mathbf{U}) \right).\vec{\mathbf{n}}_{\text{face}} \end{aligned}$$

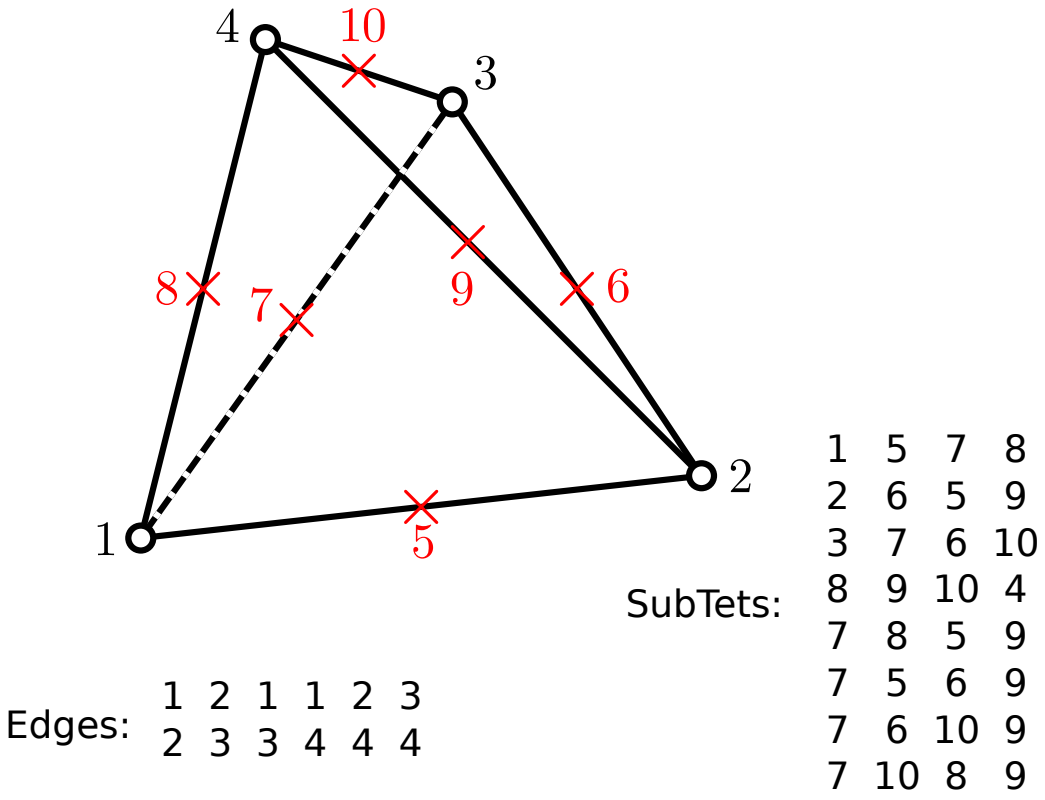


Figure 7.7: Numbering convention for $\mathbb{P}^1$ and $\mathbb{P}^2$ tetrahedra. When splitting the tetrahedron into sub-tetrahedra, the inside rhombohedron is split by its $7-9$ diagonal.

$\overrightarrow{\boldsymbol{\mathcal{F}}_h}$ being a $\mathbb{P}^k$ function, this last integral is just a linear combination of the fluxes on the DoFs sharing the face, the coefficients being the integral of the 3D Lagrangian basis function over the considered faces. The *global residual* is computed in practice as:

- In $\mathbb{P}^1$,
$$\Phi^{\mathrm{T}} = \sum_{i=1}^{4} \frac{\overrightarrow{\boldsymbol{\mathcal{F}}}_i.\vec{\mathbf{n}}_i}{3}.$$
- In $\mathbb{P}^2$,
$$\Phi^{\mathrm{T}} = \sum_{i=5}^{10} \frac{\overrightarrow{\boldsymbol{\mathcal{F}}}_i.\vec{\mathbf{n}}_i}{3}.$$

One can notice that in $\mathbb{P}^2$, the vertices of the tetrahedron do not interfere into the computation of the *global residual*. However, their values will still be used in the rest of the distribution process.

Otherwise, the rest of the scheme is almost straightforward. The Lax-Friedrichs first order residual is easily generalized to tetrahedra, the limitation is done following algorithm 1 page 93 and the stabilization term is computed using enough quadrature points in order the gradients are defined uniquely.

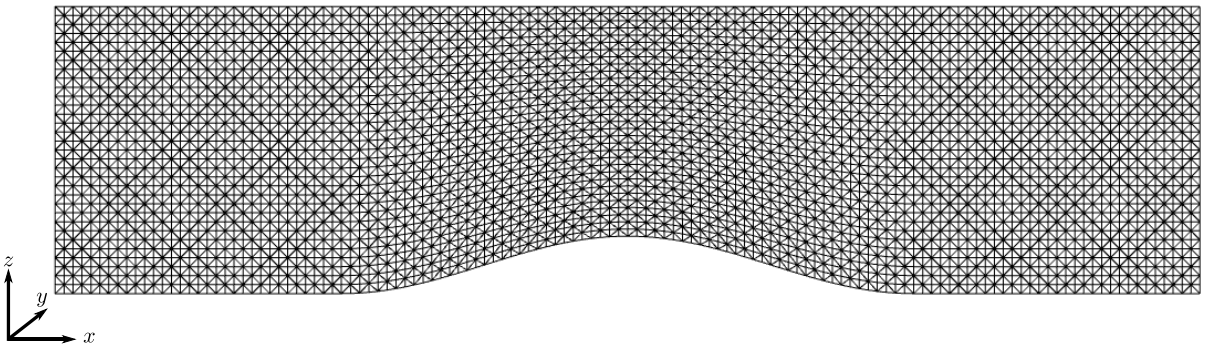


Figure 7.8: Two dimensional bump mesh. The three dimensional bump domain is obtained by a y shift of this 2D shape. This regular mesh is not representative of the one that has been used.

7.3 Numerical Results

7.3.1 3D Bump

The first test case presented in this section is a standard three dimensional bump. The main interest in this test case is to validate the code implementation, and to put the higher order better accuracy forward. We are also going to advance the implicit efficiency of the finite difference Jacobian compared to the first order Lax-Friedrichs Jacobian.

The domain is obtained by shifting along the y axis a two dimensional bump domain in Oxz shown on Figure 7.8. The flow enters the channel at section $x = x_{input}$ with a velocity having only an x component and leaves at section $x = x_{output}$. The Mach number on these sections is $Ma = 0.5$. All the sides of the channel (the bottom "bump" side plus the left, right and upper sides) are considered to be slip walls. The $\mathbb{P}^1$ mesh is made of 87349 tetrahedra and 17493 vertices. In the case of a $\mathbb{P}^2$ computation, the 87349 tetrahedra contain 128004 degrees of freedom. The problem starts to be quite big, as it requires at every iteration the resolution of a $5.10^5 \times 5.10^5$ sparse linear system. That is why the presented simulation have been split between 16 processors.

On Figures 7.9 and 7.10 we present the results obtained with second and third order 3D schemes. The cut are realized along the plane $z = 0.5$ which is the mid value of coordinate z in the mesh. On the first figure is represented in color the density component. Isolines represent the pressure. The black isolines are those of the second order solution, while the purple one come from the third order solution. First the purple isolines are globally smoother and more symmetric. This can be seen especially for the closest isolines to the top of the bump. Second, we see that they are some troubles at the beginning of the bump on both sides. It is likely this is due to the fact the mesh is rather coarse. And the third order solution does not improve very much the second order result in this region because the boundary are still represented linearly. On Figure 7.10, we have just represented the isolines of the horizontal velocity u. Second order solution is in black and third order in red. The red isolines are globally much smoother, especially on the entrance side. On this side it is clear that the third order formulation has improved the solution. But if we give a look at the output side, the difference between the two formulations is

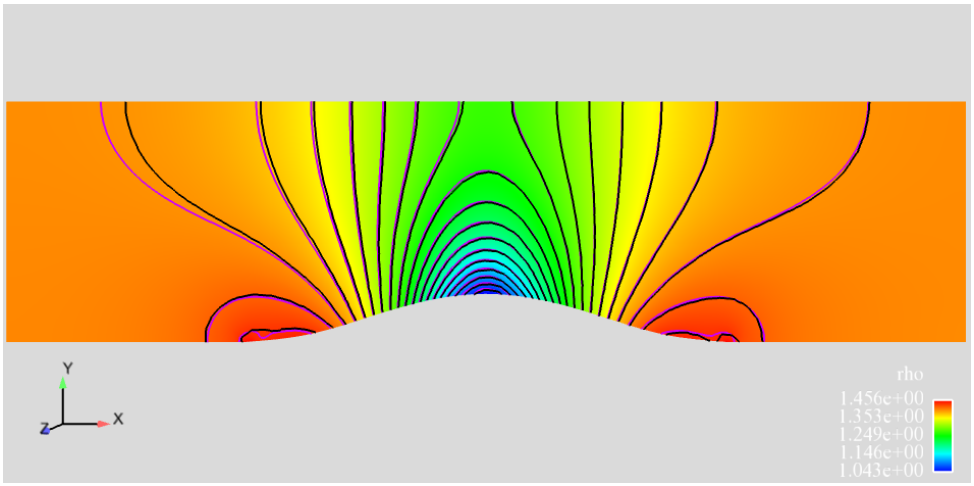


Figure 7.9: Solution of the 3D problem. In color is represented the density for the third order solution. Isolines are based on the pressure component of the solution. The second order solution is in black while the pink isolines denotes the third order solution.

more balanced. The red isolines are not varying with a monotone manner and the black isolines are not smooth at all. Globally, the solution is pretty good though and the 3D implementation of the code is validated.

Finally, we can give a look to the iterative convergence. To solve this problem, we have used two different types of Jacobian matrices of the residuals. we have used the first order Jacobian matrices, presented in Subsection 5.2.3, and the finite difference Jacobian matrices, see Subsection 5.2.4. We show here that the finite difference Jacobian matrices are more expensive in terms of calculations, but that it finally tremendously improves the scheme convergence. On Figure 7.11, we have represented the residual convergence with respect to the number of iterations. We see here that the scheme with finite difference Jacobian matrices converges to 10^{-6} within about a thousand iterations while it would have taken more than $20,000$ iterations for first order Jacobians scheme to reach the same level of convergence. What is hidden behind is that the computation of the finite difference Jacobian matrices is in fact much more expensive than for the first order Jacobians. We can see that on Figure 7.12 where we see that the scheme with finite difference Jacobian matrices is approximately 3 times faster than the one using first order Jacobian matrices in term of CPU time. A quick calculation show then that the computation of the finite difference Jacobian matrices is approximately 6 times more expensive that the first order Jacobian matrices. What is a bit disappointing is that the fastest scheme seems to saturate when reaching 10^{-7} residual error, whereas it should to converge toward at least 10^{-11}. It is a pity, because as one can see the convergence of the scheme with first order Jacobian matrices always changes slope around 10^{-5} and there usually seriously slows down. The advantage of the finite difference Jacobian matrices would then be increased when higher iterative convergence is needed.

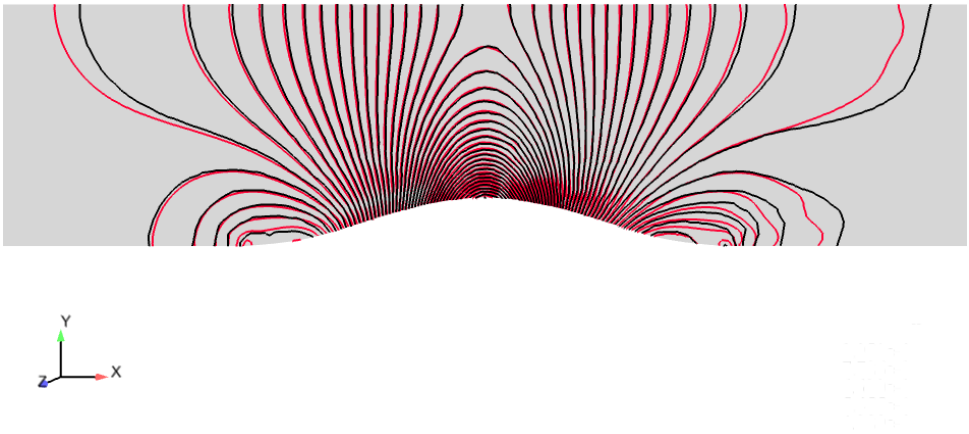


Figure 7.10: Comparison of the isolines of the horizontal velocity u of the second (black) and third order (red) solutions of the 3D bump problem.

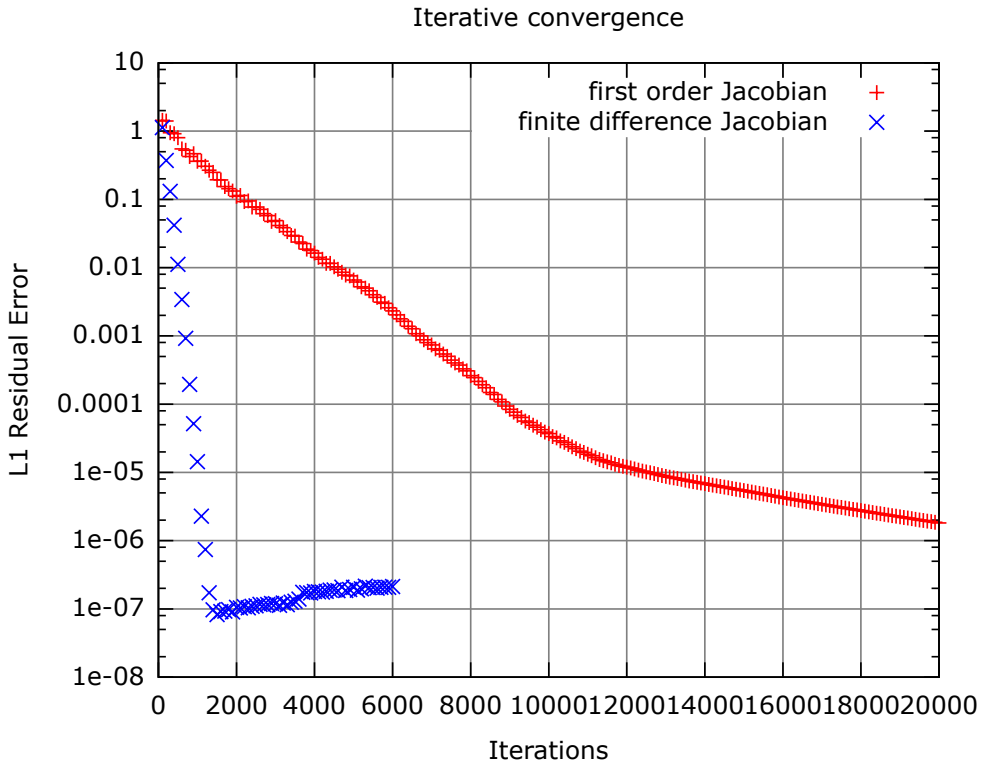


Figure 7.11: Residual L^1 norm convergence plotted with respect to the number of iterations for the schemes using finite difference and first order matrices.

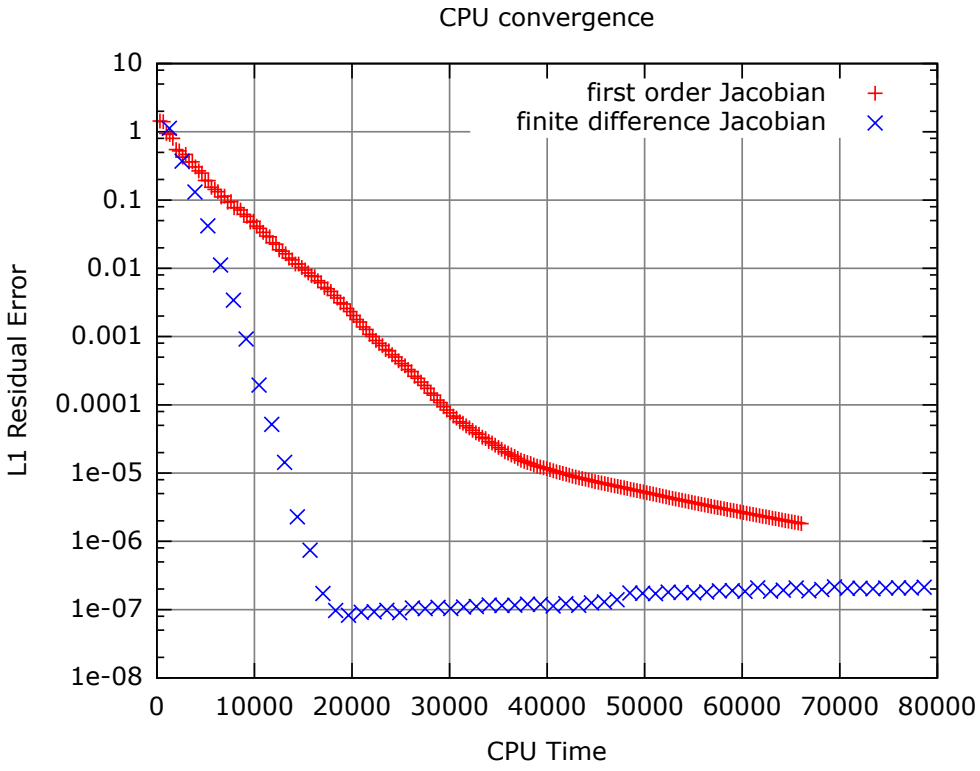


Figure 7.12: Residual L^1 norm convergence plotted with respect to the CPU time (in seconds) for the schemes using finite difference and first order matrices.

7.3.2 Subsonic Blunt Airfoil

The second test case is a "blunt" airfoil. The shape of the simulated aerodynamic object is similar to a cigar. We have run the simulations on four unstructured meshes, composed only of tetrahedra. These meshes have been generated by the VKI[6], in cooperation within the European ADIGMA project. The main characteristics of the meshes are detailed on Tabular 7.2. Second and third order simulations have been completed on these meshes, as well as second order simulations on the 3^{rd} order DoFs. The flow parameters are the following:

- Incidence: $\alpha = 5°$;
- Mach Number: Ma $= 0.5$.

Mesh	Vertices ($\times 10^3$)	$\mathbb{P}^2$ DoFs ($\times 10^6$)	Tetrahedra ($\times 10^6$)
1	33.8	0.26	0.19
2	45.6	0.35	0.25
3	80.5	0.62	0.44
4	245	1.87	1.31

Table 7.2: Number of $\mathbb{P}^1$ and $\mathbb{P}^2$ DoFs as well as the number of tetrahedra for the four meshes around the blunt airfoil. The domain is a sphere which diameter is 10 times larger that the airfoil cord.

On Figures 7.13 and 7.14 are represented the solutions for the three types of schemes: $\mathbb{P}^1$, $\mathbb{P}^2$ and $\mathbb{P}^1$ on the $\mathbb{P}^2$ DoFs. The color palette represents the entropy and the isolines are based on the density component. It is very clear on these images that the third order simulation has really improved the solution. And the graphical representation is not even quadratic. To represent the $\mathbb{P}^2$ solution, we have just given the vectorial values at each degree of freedom and ask the visualization software to show the solution linearly by sub-tetrahedra. Then the real third order solution looks even smoother. There are two important things to notice. First, the numerical entropy production at the stagnation point is much reduced when using a higher order scheme. Once more the problem is perfectly adiabatic and the entropy should be constant all over the mesh. Second, we observe that the isolines of entropy are wiggled along the blunt body. We do not have any precise explanation for this. We know that the airfoil is represented by faces: its surface is not smooth. We can see that above all when looking at the "nose" around the stagnation point. Furthermore, these wiggles appear in the region where the gradient of the variables is very small. Eventually, one can see that there is an extra numerical entropy production at the tail of the body. This is clear on the third order solution and on the solution of second order on the $\mathbb{P}^2$ mesh. This an extra argument to claim that the scheme is very sensitive to the boundary representation and to the boundary treatment in the code. Therefore, it would be very interesting to develop an isoparametrical representation of the boundaries in 3D, and to see if this helps this problem. However, this is not a simple task because the generalization of the two dimensional technique used in Subsection 6.2.4 gives a discontinuous representation of the boundary faces in 3D. The first thing is then to find a way of representing the boundaries a continuous quadratic manner.

[6] Von Karman Institute, Brussels, Belgium

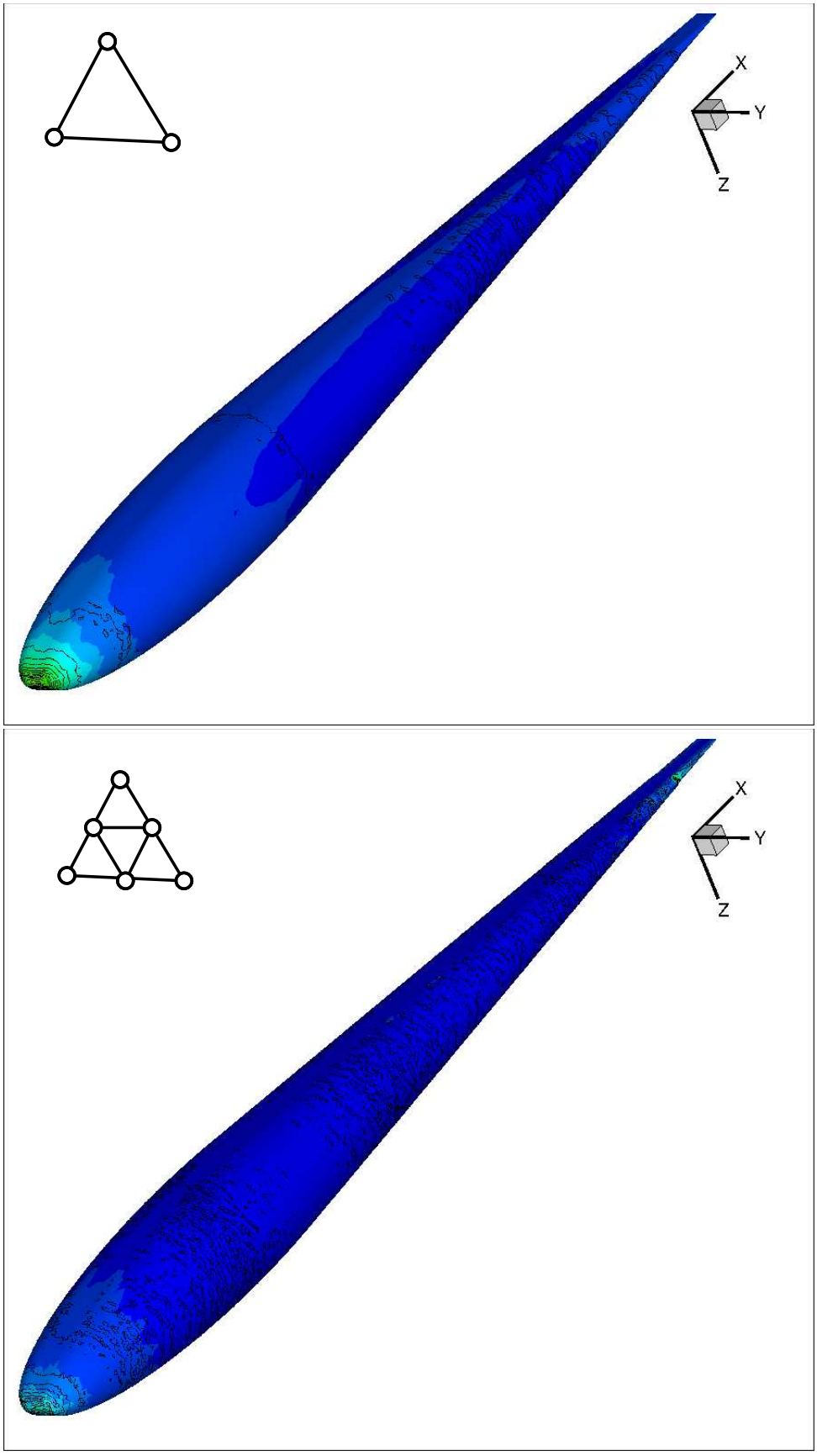


Figure 7.13: 2 solutions of the three dimensional Blunt Airfoil problem. The top one is the second order one, and the bottom one represents the solution obtained with a second order scheme on the subdivision of the third order mesh. Color palette represents the entropy while the isolines are based on the density component of the solution.

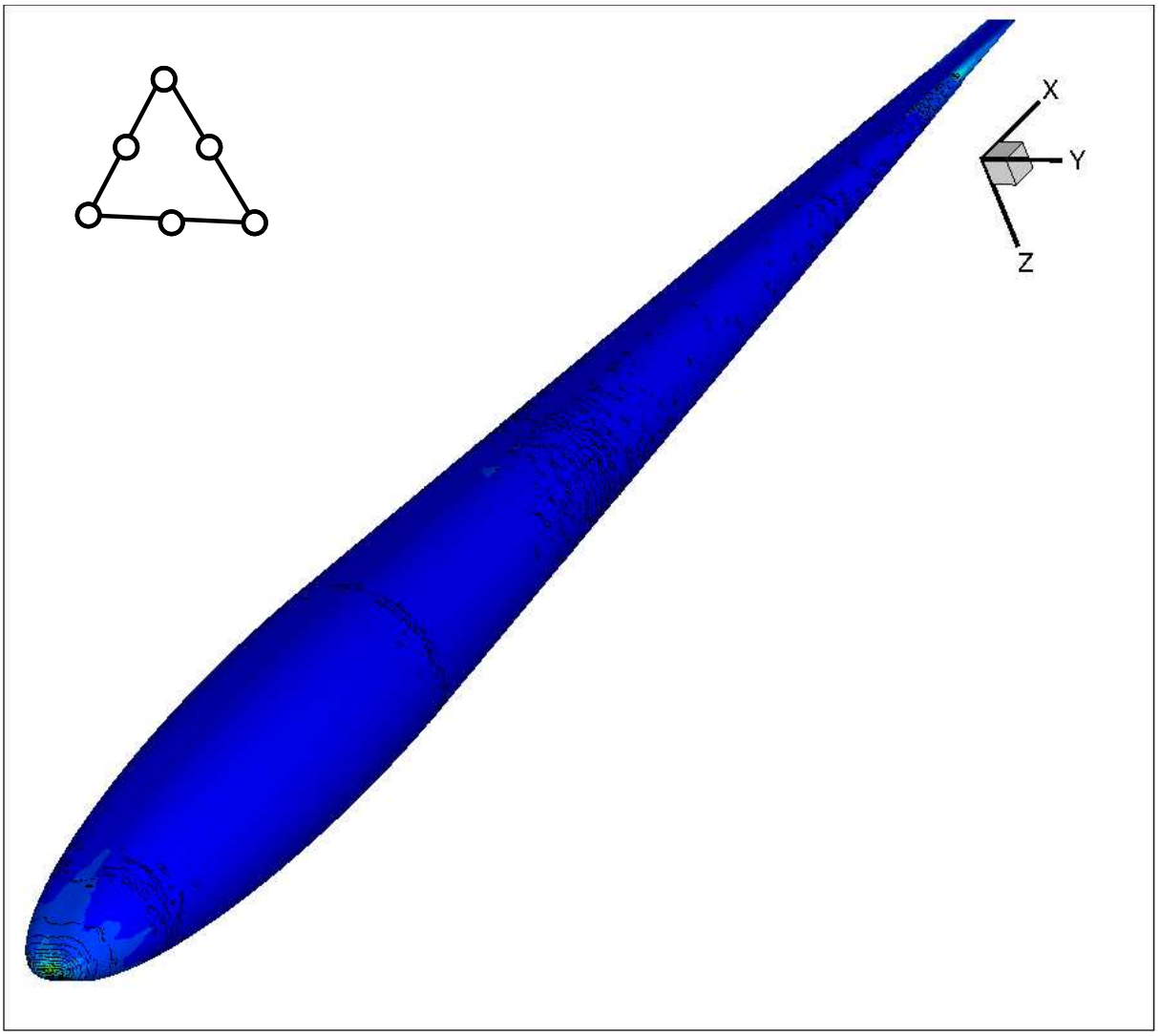


Figure 7.14: Third order solution for the Blunt Airfoil problem. As for Figure 7.13, the color palette represents the entropy while the isolines are based on the density component of the solution.

7.3.3 Transonic M6 Wing

We present here this test case because it is computed with transonic data and the solution is then discontinuous. It is interesting to analyze the comportment of the 3D scheme with shocks. The mesh has $265,000$ nodes and 1.64 million tetrahedra. The flow at infinity has the following characteristics:

- Incidence: $\alpha = 3^\circ$;
- Mach: $Ma = 0.84$

On Figure 7.15 is represented the top side of the body plus the solution over the plane $z = 0$. In color is represented the pressure and the isolines are based on the Mach number. This is only a $\mathbb{P}^1$ solution. Unfortunately, we have not been able to run a third order simulation. In that case, the computation starts to converge and then suddenly crashes. The reasons have not been discovered yet. It could come from the discontinuous character of the solution as from some default in the parallelization, or even from a bad implementation of the code for higher order 3D. But we more likely believe that this comes from the mesh and the boundary representation. As we can see on Figure 7.16, the end of the wing is represented very coarsely and there are even holes near the trailing edge. In $\mathbb{P}^2$, this could lead to the appearance of some unphysical phenomena that would make the computation crash. The second order solution of Figure 7.15 is nevertheless very good, we can notice that the shocks are well resolved and that the expected *lambda shock* can be seen between the main shock and the leading edge.

We also represent on Figure 7.17 the profile of the pressure component around the wing at $z = 0$. Of course, due to the fact the incident flow comes bellow the wing, the upper part of the curves correspond to the lower part of the wing and vice versa. At $x = 0$ is the stagnation point with the maximal pressure value of the whole domain. On the upper side, the pressure goes down to a local minimum which looks like to a small shock. It is the root of the lambda shock we observe then along the wing. At approximately $x = 0.6$, we observe the main shock which is pretty sharp an does not show any spurious oscillation. Finally, it is interesting to look at the trailing edge where we seem to have an unphysical value in the last layer of the mesh. The pressure suddenly drops down. We cannot explain that at that moment.

7.3.4 A Complete 3D Aircraft

We finally end this chapter with an Euler simulation on a complete aircraft. The name of the model is SSBJ, and it is a private supersonic jet (SuperSonic Business Jet) that has been designed by Dassault. The meshes have been designed by F. Alauzet at INRIA Rocquencourt.

The mesh has 203 kNodes and 1.15 million tetrahedra. The $\mathbb{P}^1$ solution presented here has been computed on 32 processors. As for the M6 Wing, no $\mathbb{P}^2$ results are available at that moment but we are keeping fare hope to publish them in a couple of months. The caracteristics of the simulation are the following:

- Incidence: $\alpha = 5^\circ$;
- Mach: $Ma = 2.0$

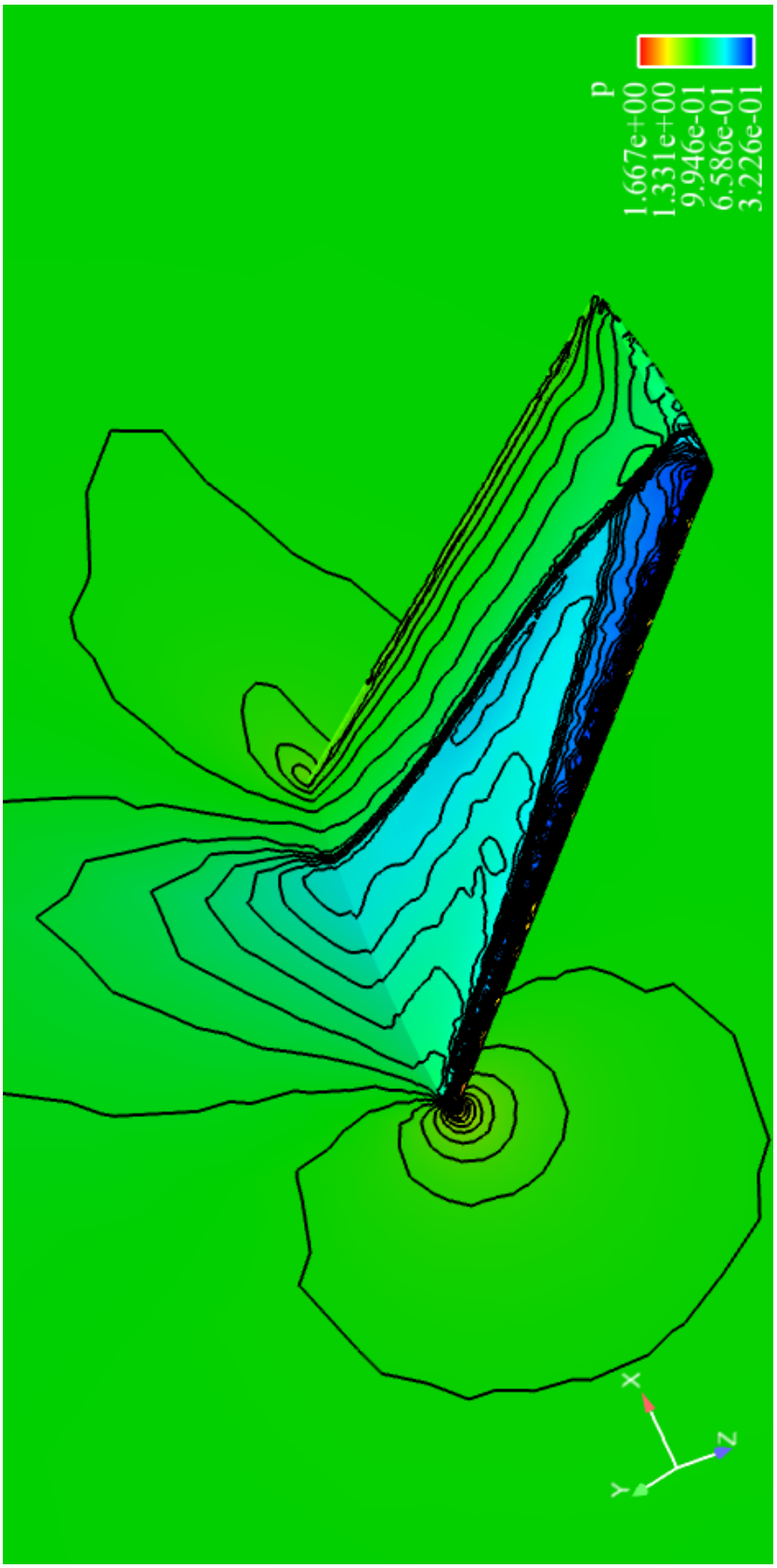


Figure 7.15: Top side view of the M6 Wing. Background is the solution over $z = 0$ plane. In color is represented the pressure and the isolines show the Mach number. The solution is only $\mathbb{P}^1$.

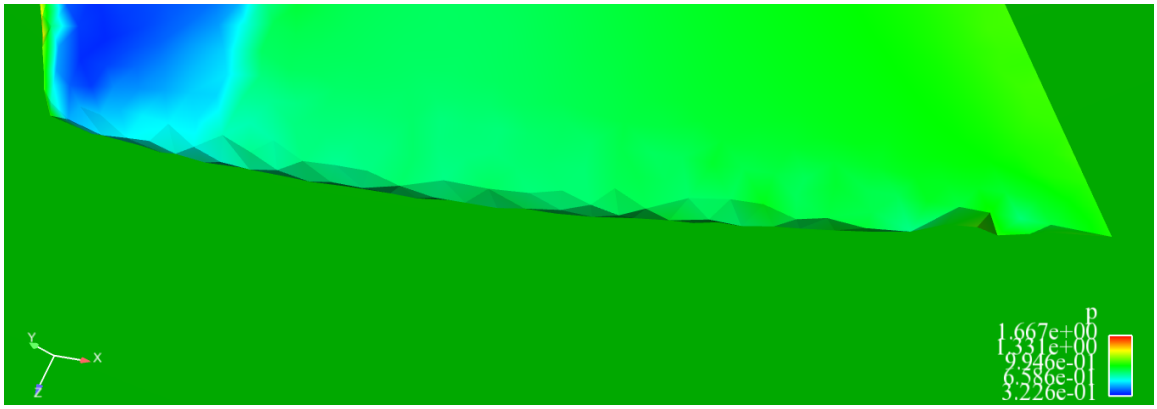


Figure 7.16: Zoom on the mesh at the end of the wing. We can see the representation of the body is very poor, there are even holes near the trailing edge. This could possibly explain why the third order simulation crash suddenly after a small convergence.

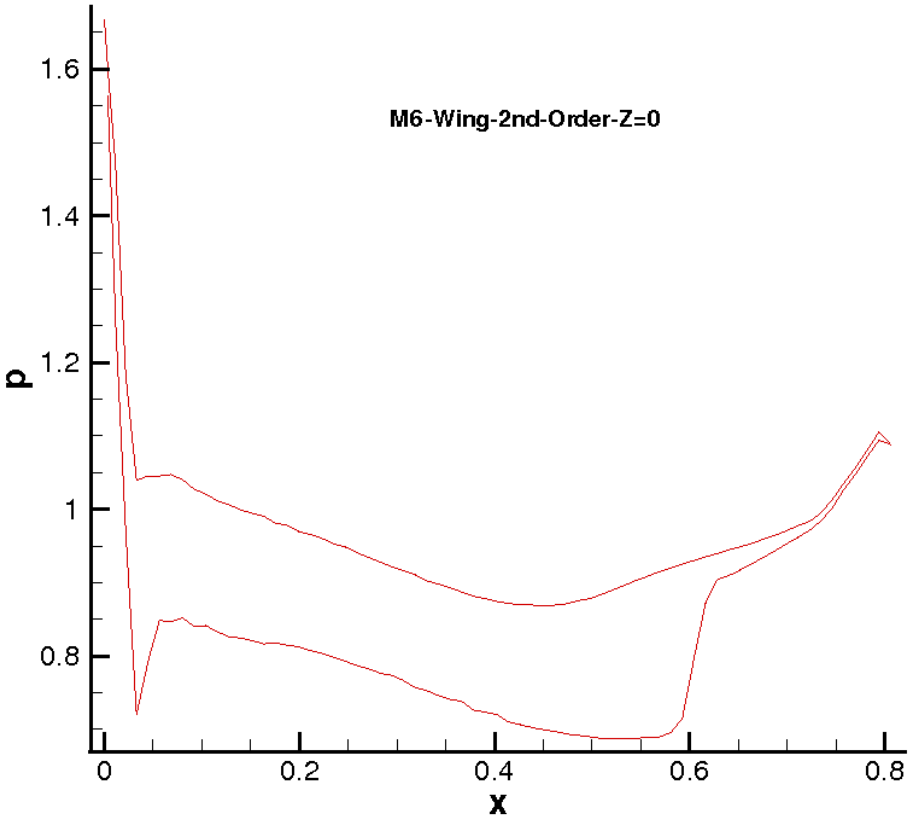


Figure 7.17: Profile of pressure around the wing at $z = 0$.

On Figure 7.18, we have represented all the shock surfaces around the aircraft. The body of the jet is colored by the density component. On Figure 7.19, we have represented the isolines of the density component over three different clipping planes situated at coordinates $x = 1$, $x = 3$, $x = 4$. The skin of the aircraft is colored by the x-velocity component.

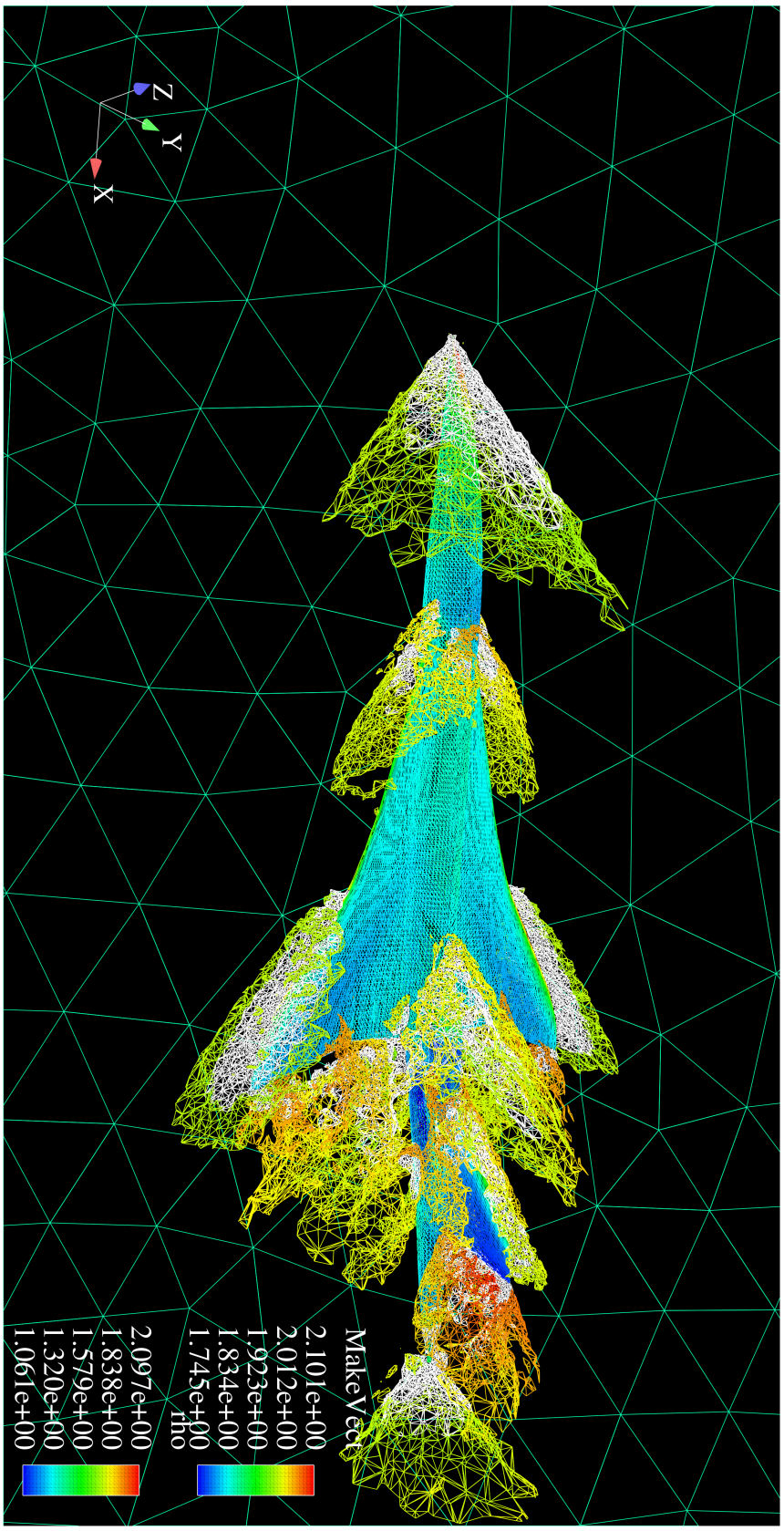


Figure 7.18: Shock surfaces of the simulation of a supersonic business jet at Mach=2.0. The color on the body represents the density component.

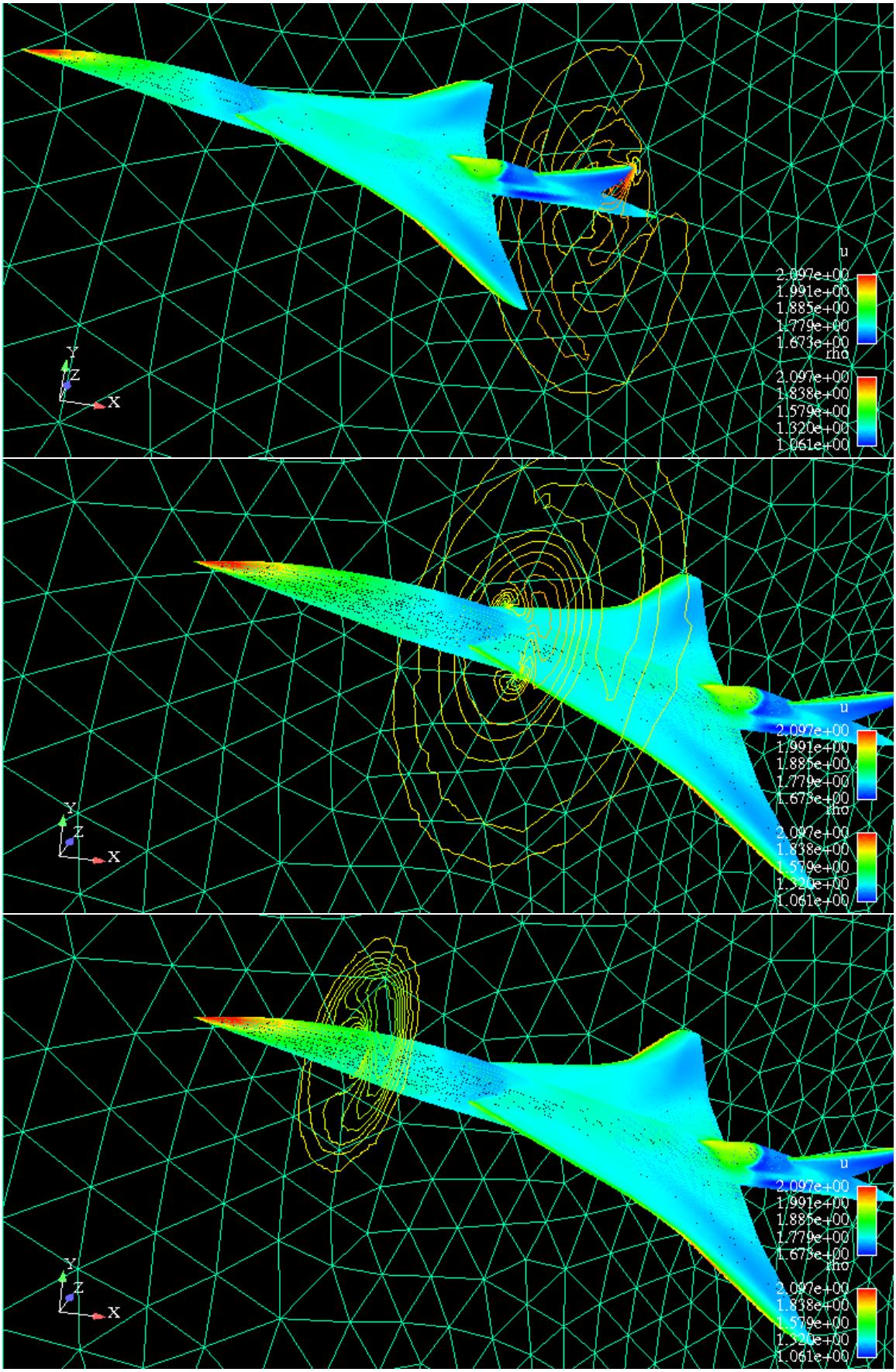


Figure 7.19: Isolines of the density component over 3 different clipping planes. The body color is the u component.

Chapter 8

Navier-Stokes Simulations

The physical system of the viscous Navier-Stokes equations have been presented in Subsection 2.2.10. The main difference with the Euler equations is the right hand side viscous term that not only involves the conservative variables, but also their spatial derivatives. This is still a big problem for the residual formulation of the Navier-Stokes equations. The ground idea of the $\mathcal{RDS}$ is that the solution is represented continuously, so that there is no need of any numerical flux to represent the interactions between the elements. One could of course store also the gradients of the conservative variables and rewrite the formulation in terms of the density, the momentum, the energy and their spatial derivatives, but this would be costly as this would multiply the number of unknown by the number of spatial dimensions. One could also consider the gradient functions inside each element and choose an associated smooth approximation: for example an $\mathcal{L}^2$ projection of the discontinuous gradients on the space of continuous functions. This method has two main drawbacks. First this method comes with a non negligible extra cost. At each time step, one has to reconstruct the chosen approximation. Second, the $\mathcal{L}^2$ projection cannot be implemented in a compact way, and this destroys the maximal compactness of the scheme. Therefore, the parallel efficiency of the scheme is going to be much reduced.

We have then chosen to discretize this viscous term by a Finite Element Galerkin formulation. The reasons are it handles well the discontinuous character of the gradients of the variables due to the compact support of the basis functions, and it keeps a maximum compact expression. The first section is going to describe the practical numerical formulation of the viscous term. Second section aims at explaining the theoretical consistency between the residual formulation of the inviscid flux and the Galerkin formulation of the viscous part. We are there going to see the problem is well-posed, but unfortunately not high order anymore. In the two next sections, we are going to present some results obtained with this formulation. The two dimensional Blasius boundary layer is our first test case showing that the formulation is working reasonably well. We compare the obtained results with the nondimensional exact solution. The second test case is a viscous NACA012 test case on which we are going to study the convergence rate of our formulation. This chapter will finally end with a review of other formulation that are today used or in development for the discretization of the viscous terms in the $\mathcal{RDS}$ framework.

8.1 Finite Element Galerkin Formulation

We first recall that the steady Navier-Stokes equation can be put into the form

$$\mathrm{div}\left(\overrightarrow{\boldsymbol{\mathcal{F}}}^E(\mathbf{U})\right) = \mathrm{div}\left(\overrightarrow{\boldsymbol{\mathcal{F}}}^V(\mathbf{U})\right), \tag{8.1}$$

where $\overrightarrow{\boldsymbol{\mathcal{F}}}^E$ and $\overrightarrow{\boldsymbol{\mathcal{F}}}^V$ stand for the Euler and the viscous flux respectively. Then, the Galerkin contribution of the right hand side to node i of the mesh is given by:

$$V_i = -\int_\Omega \overrightarrow{\nabla\varphi_i}\cdot\overrightarrow{\boldsymbol{\mathcal{F}}}^V(\mathbf{U}_h)dx, \tag{8.2}$$

where once more, φ_i is the Lagrange basis function associated to node i. This contribution is indeed split into the the sum of the integrals on the elements T where φ_i is not identically zero, which gives the element viscous contributions:

$$V_i^{\mathrm{T}} = -\int_{\mathrm{T}} \overrightarrow{\nabla\varphi_i}\cdot\overrightarrow{\boldsymbol{\mathcal{F}}}^V(\mathbf{U}_h)dx. \tag{8.3}$$

These integrals are computed thanks to a quadrature formula. For third order problem we usually use a 6 points Gaussian quadrature formula. In the implicit case, it is also useful to express the viscous flux in its quasi-linear form given in Subsection 2.2.10:

$$\overrightarrow{\boldsymbol{\mathcal{F}}}_i^V(u^h) = \sum_{j=1}^{dim} K_{ij}U_{,j},$$

where symbol $_{,j}$ stands for the derivative with respect to the j^{th} spatial variable. Tensor $\mathbb{K} = (K_{ij})_{i,j\in[\![1,dim]\!]}$ is fully described page 38. Then the viscous contribution to node i writes

$$\begin{aligned} V_i^{\mathrm{T}} &= |\mathrm{T}|\sum_{q=1}^{6}\omega_q\overrightarrow{\nabla\varphi_i}(x_q)\cdot\overrightarrow{\boldsymbol{\mathcal{F}}}^V(\mathbf{U}^h(x_q)) \\ &= |\mathrm{T}|\sum_{q=1}^{6}\sum_{j\in\mathrm{T}}\left\{\omega_q\overrightarrow{\nabla\varphi_i}(x_q)\cdot\left(\mathbb{K}(x_q)\overrightarrow{\nabla\varphi_j}(x_q)\right)\right\}\mathbf{U}_j \\ &= \sum_{j\in\mathrm{T}}\mathcal{M}_{ij}^T\mathbf{U}_j. \end{aligned}$$

This last equation reveals why the quasi-linear form of the viscous flux is so appealing. For the implicit formulation, we have just to assemble the matrices $\mathcal{M}_{ij}$, and the i^{th} line of the iteration linear system writes:

$$\begin{aligned} \left(\frac{\mathbf{I}}{\omega_i}+\sum_{\mathrm{T}\in\mathcal{D}_i}\left(\frac{\partial\Phi_i^{\mathrm{T}}}{\partial\mathbf{U}_i}+\mathcal{M}_{ii}^{\mathrm{T}}\right)\right)\Delta\mathbf{U}_i+\sum_{\substack{j\in\mathcal{D}_i\\ j\neq i}}\left(\sum_{\mathrm{T}\in\mathcal{D}_i\cap\mathcal{D}_j}\left(\frac{\partial\Phi_i^{\mathrm{T}}}{\partial\mathbf{U}_j}+\mathcal{M}_{ij}^{\mathrm{T}}\right)\right)\Delta\mathbf{U}_j \\ = -\sum_{\mathrm{T}\in\mathcal{D}_i}\left(\beta_i^*\Phi^{\mathrm{T}}(\mathbf{U})+\sum_{j\in\mathrm{T}}\mathcal{M}_{ij}^T\mathbf{U}_j\right) \end{aligned} \tag{8.4}$$

8.2 Consistency of the Viscous Term Treatment

We consider that all the boundary conditions are treated weakly and whether we impose a flux or the solution on the boundary, we always enforce a numerical flux that is going to be denoted by $\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\mathbf{U}_{\text{bound}})$. For all boundary types, the contribution of boundary condition to node i in element T is B_i^{T}. It is zero when i is strictly inside the domain. Now, the Navier-Stokes scheme writes:

$$\sum_{\text{T}\in\mathcal{D}_i}\left(\Phi_i^{\text{T}}+V_i^{\text{T}}\right)+\sum_{\partial\text{T}\cap\partial\Omega}B_i^{\text{T}}=0 \tag{8.5}$$

We recall, following what has been said in Section 5.4, that weak boundary conditions are of the form:

$$B_i^{\text{T}}=\int_{\partial\text{T}\cap\partial\Omega}\varphi_i^k\left(\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\mathbf{U}_{\text{bound}})-\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\mathbf{U}_h)\right).\vec{\mathbf{n}}\,dx. \tag{8.6}$$

We define Θ_i^{T} as the Galerkin Navier-Stokes residual, meaning

$$\Theta_i^{\text{T}}(\mathbf{U}_h) \;=\; \Psi_i^{\text{T}}+V_i^{\text{T}} \tag{8.7}$$

where Ψ_i^{T} has been defined in (4.18) as the Galerkin Euler residual. Then for any $\Upsilon\in\mathcal{C}^1(\mathbb{R}^2)$, if Υ_i is its value at node i, one has:

$$\sum_{i\in\mathcal{M}_h}\Upsilon_i\sum_{\text{T}\in\mathcal{D}_i}\left(\Phi_i^{\text{T}}+V_i^{\text{T}}\right)+\sum_{i\in\mathcal{M}_h}\Upsilon_i\sum_{\partial\text{T}\cap\partial\Omega}B_i^{\text{T}}=0 \tag{8.8}$$

$$\Rightarrow\quad\underbrace{\sum_{\text{T}\in\mathcal{M}_h}\sum_{i,j\in\text{T}}\left(\Phi_i^{\text{T}}(u_h)-\Psi_i^{\text{T}}(u_h)\right)(\Upsilon_i-\Upsilon_j)}_{\mathbf{I}}+\underbrace{\frac{1}{q}\sum_{\text{T}\in\mathcal{M}_h}\sum_{i\in\text{T}}\Upsilon_i\Theta_i^{\text{T}}(\mathbf{U}_h)}_{\mathbf{II}}$$

$$+\underbrace{\sum_{\text{T}\in\partial\Omega}\sum_{i\in\text{T}}\Upsilon_iB_i^{\text{T}}}_{\mathbf{III}}=0 \tag{8.9}$$

Now the proof is really similar to the one of Theorem 4.4 page 68. Let us begin by term **III**. Following what has been done for term **II** in the proof of Theorem 4.4, we have:

$$\mathbf{III} \;=\; \int_{\partial\Omega}\left(\pi_h^k\Upsilon\right)(\mathbf{x})\left(\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\mathbf{U}_{\text{bound}})-\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\mathbf{U}_h)\right).\vec{\mathbf{n}}dx \tag{8.10a}$$

$$\;=\;\underbrace{\int_{\partial\Omega}\Upsilon\overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U}_{\text{bound}}).\vec{\mathbf{n}}\,dx}_{\mathbf{i}}-\underbrace{\int_{\partial\Omega}\left(\pi_h^k\Upsilon\right)(\mathbf{x})\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\mathbf{U}_h).\vec{\mathbf{n}}\,dx}_{\mathbf{ii}}$$

$$+o_h(1) \tag{8.10b}$$

Now if we give a look to term **II**, we get easily that

$$\mathbf{II} \;=\; \int_\Omega\left(\pi_h^k\Upsilon\right)(\mathbf{x})\overrightarrow{\nabla}.\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\mathbf{U}_h)d\mathbf{x}+\int_\Omega\overrightarrow{\nabla(\pi_h^k\Upsilon)}.\overrightarrow{\boldsymbol{\mathcal{F}}_h}^V(\mathbf{U}_h)d\mathbf{x} \tag{8.11}$$

and if we use term **ii** of equation (8.10b), we can apply the Green formula and obtain

$$\mathbf{II} \;=\; -\int_\Omega\overrightarrow{\nabla(\pi_h^k\Upsilon)}.\left(\overrightarrow{\boldsymbol{\mathcal{F}}_h}(\mathbf{U}_h)-\overrightarrow{\boldsymbol{\mathcal{F}}_h}^V(\mathbf{U}_h)\right)d\mathbf{x}. \tag{8.12}$$

Then starting the similar reasoning as at equation (4.20c), we get:

$$\mathbf{II} \;=\; -\int_\Omega \overrightarrow{\nabla\Upsilon}.\left(\overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U}) - \overrightarrow{\boldsymbol{\mathcal{F}}}^V(\mathbf{U})\right) d\mathbf{x} + o_h(1). \tag{8.13}$$

Finally, term **I** is the same as in equation (4.19) and Lemma 4.5 proves it is bounded by h. We can now conclude, because if $(u_h)_h$ is a sequence of numerical solution of (8.5) verifying assumptions of Theorem 4.4 and $u \in \mathcal{L}^2(\mathbb{R}^2)$ is a function such that

$$\lim_{h\to 0} \|u - u_h\|_{\mathcal{L}^2_{loc}(\mathbb{R}^2)} = 0,$$

then

$$\int_\Omega \overrightarrow{\nabla\Upsilon}.\left(\overrightarrow{\boldsymbol{\mathcal{F}}}(u) - \overrightarrow{\boldsymbol{\mathcal{F}}}^V(\mathbf{U})\right) d\mathbf{x} - \int_{\partial\Omega} \Upsilon \overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U}_{\text{bound}}).\vec{\mathbf{n}}\, dx = o_h(1)$$

and u is a weak solution of the Navier-Stokes equations with the $\mathbf{U}_{\text{bound}}$ boundary conditions.

8.3 Accuracy Discussion

In this section, we wanted to prove that the problem mixing the residual formulation of the advective term and the Galerkin formulation of the viscous term is well-posed but that we can not expect to reach the $(k+1)^{\text{th}}$ order maximal accuracy. The demonstration has not been completed so far and that is why we only give here a intuition of what is going on and some routes to begin with the complete proof.

We first start by recalling the following theorems, for which proves can be found in [1].

Theorem 8.1 (*Neças*)

Let V and W two Hilbert spaces and $V_h \subset V$ and $W_h \subset W$ two approximations of these spaces that have the same space dimension. Let $a \in \mathcal{L}(V \times W, \mathbb{R})$ and $f \in \mathbf{V}'$. Then, the following problem

$$\begin{cases} \text{Find } u_h \in W_h \text{ such that} \\ a(u_h, v_h) = f(v_h), \forall v_h \in V_h \end{cases} \tag{8.14}$$

is well-posed if and only if there exist a constant $\alpha_h > 0$ such that

$$\inf_{w_h \in W_h} \sup_{v_h \in V_h} \frac{a(w_h, v_h)}{\|w_h\|_W \|v_h\|_V} \geqslant \alpha_h. \tag{8.15}$$

In that case, we have the following estimation

$$\forall f \in V', \|u_h\|_W \leqslant \frac{1}{\alpha_h}\|f\|_{V'}. \tag{8.16}$$

Lemma 8.2 (*Céa*)

Under the previous hypothesis, if u is the unique solution of problem

$$\begin{cases} \text{Find } u \in W \text{ such that} \\ a(u, v) = f(v), \forall v \in V \end{cases} \tag{8.17}$$

we have

$$\|u-u_h\|_W \leqslant \left(1+\frac{\|a\|}{\alpha_h}\right) \inf_{w_h\in W_h} \|u-w_h\|_W. \tag{8.18}$$

Next, we consider the following 1D advection-diffusion problem

$$\begin{cases} a\overrightarrow{\nabla u} = \varepsilon \Delta u, \quad x\in\Omega=[0;1] \\ u(0)=u(1)=0 \end{cases} \tag{8.19}$$

We have taken the homogeneous Dirichlet boundary condition to get rid of the boundary treatment. This will greatly simplify the explanation. Let us proceed to the variational formulation. Let φ be a test function, we have

$$a\int_0^1 \varphi\overrightarrow{\nabla u}dx + \varepsilon\int_0^1 \overrightarrow{\nabla\varphi}\overrightarrow{\nabla u}dx = 0$$

This has a sense when u and φ belong to $\mathrm{H}_0^1([0;1])$. Then we set

$$X=\mathrm{H}_0^1(\Omega)$$

and the problem becomes in its weak formulation:

$$\begin{cases} \text{Find } u\in X \text{ such that} \\ a\int_0^1 v\overrightarrow{\nabla u}dx + \varepsilon\int_0^1 \overrightarrow{\nabla v}\overrightarrow{\nabla u}dx = 0, \forall v\in X \end{cases} \tag{8.20}$$

This reasoning is very classical. Now, we divide $[0;1]$ into N regular intervals, and we set: $h=\frac{1}{N}$ and $\forall i\in[\![0;N]\!], x_i=ih$. T_i denotes interval $[x_i;x_{i+1}]$. On this 1D mesh, we define basis functions. For node i and interval T, the basis function writes:

$$\varphi_i^{\mathrm{T}} = \lambda_i^{\mathrm{T}} + \alpha_i^{\mathrm{T}}\gamma^{\mathrm{T}}, \tag{8.21}$$

where λ_i^{T} is the $\mathbb{P}^1$ Lagrange basis function in i, and γ^{T} is a piecewise linear continuous function that is null outside of T and that takes value 1 at the mid point of T. α_i^{T} is a coefficient that will actually depend on the solution but its value stays bounded. This basis function can thus be seen as a non linear perturbation of the Lagrange basis functions. Because γ^{T} is zero at all the nodes of the mesh, the basis functions can be joined continuously and it allows us to define the functional space approximation:

$$X_h = \mathrm{Span}_{i\in[\![1;N-1]\!]}\{\varphi_i\} \subset \mathrm{H}_0^1.$$

Then for $u\in X$, if u belongs moreover to H^2, because the α_i^{T} are bounded, we have the estimation

$$\|\pi_{X_h}(u)-u\|_{1,\Omega} \leqslant Ch, \tag{8.22}$$

constant C depending on $\|u\|_{2,\Omega}$.

The scheme writes then: find $u_h\in X_h$, such that $\forall i\in[\![1;N-1]\!]$,

$$0 = \sum_{k=i-1}^{i}\left(a\int_0^1 \varphi_k^{\mathrm{T}_k}\overrightarrow{\nabla u}dx + \varepsilon\int_0^1 \overrightarrow{\nabla\varphi_k^{\mathrm{T}_k}}\overrightarrow{\nabla u}dx\right) \tag{8.23}$$

$$= \sum_{k=i-1}^{i}\left(\frac{1}{2}(1+\alpha_k^{\mathrm{T}_k})\Phi^{\mathrm{T}} + \varepsilon\int_0^1 \overrightarrow{\nabla\lambda_k^{\mathrm{T}_k}}\overrightarrow{\nabla u}dx\right) \tag{8.24}$$

which is the 1D formulation of scheme (8.5) for problem (8.19), if

$$\beta_i^{\mathrm{T}} = \frac{1}{2}(1 + \alpha_k^{\mathrm{T}_k}) \quad \Longrightarrow \quad \alpha_k^{\mathrm{T}_k} = 2\beta_i^{\mathrm{T}} - 1.$$

The coefficient α_i^{T} is then bounded under the $\mathcal{LP}$ condition

Problem (8.19) is well posed if we can find a positive coefficient α_h such that (8.15) is met for the bilinear form

$$a(u,v) = a\int_0^1 v\overrightarrow{\nabla u}dx + \varepsilon\int_0^1 \overrightarrow{\nabla v}\overrightarrow{\nabla u}dx.$$

In fact, what we are expecting to find is that there exists two constants C_1 and C_2 such that

$$C_1 h \leqslant \alpha_h \leqslant C_2 h,$$

which means that the problem is indeed well posed, but also by Lemma 8.2 that the error estimation loses one order of accuracy when h becomes smaller. By (8.22),

$$\inf_{w_h \in W_h} \|u - w_h\|_{1,\Omega} \leqslant \|u - \pi_{X_h}(u)\|_{1,\Omega} \leqslant Ch,$$

then

$$\|u - u_h\|_{1,\Omega} \leqslant h + \frac{\|a\|}{C_2}$$

and the scheme is first order accurate while $h >> \frac{\|a\|}{C_2}$ and it looses its accuracy for smaller values of h.

8.4 Two Dimensional Blasius Layer

We start this sequence of viscous test cases by the Blasius Layer because it is one of the only test cases for which we have an exact solution, meaning that we know the equation governing the boundary layer. The problem is the following: the domain is the upper right quarter of the plane. The line $y = 0$ is a planar non-slip wall. The flow enters the domain along axis $x = 0$ and is homogeneous and parallel to the wall, with velocity u_∞. This problem have been solved by P.R.H Blasius in 1907 [22, 107]. The main ideas are presented here. The fluid is considered incompressible and the main assumption is to consider that the thickness δ of the boundary layer is very small compared to the size L of the non-slip wall. If we now look at the dimensional order of the derivatives in the complete Navier-Stokes equation, we can neglect some terms and obtain the *incompressible boundary layer equations*

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial x} = 0 \tag{8.25}$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\frac{\partial^2 u}{\partial y^2} \tag{8.26}$$

$$\frac{\partial p}{\partial y} = 0 \tag{8.27}$$

For simplicity, we have used the kinematic viscosity $\nu = \frac{\mu}{\rho}$. Furthermore, by this dimensional study, we reveal a characteristic thickness size

$$\delta(x) = \sqrt{\frac{\nu x}{u_\infty}}, \tag{8.28}$$

and because the solution must be self-similar due to the infinite size of the wall, the speed profile is given by

$$u = u_\infty\, g(s), \quad s = y/\delta(x). \tag{8.29}$$

Because of equation (8.27), the *pressure* is just a function of x, and the non-slip wall does not perturb the flow far away from the boundary layer. The pressure is there equal to p_e, the external pressure for the non perturbed problem and $\dfrac{\partial p}{\partial x} = 0$ everywhere.

Now equation (8.25) tells us that $(u\,dx - v\,dy)$ is a closed differential form and following Poincaré's lemma, there exist ψ such that

$$u = \frac{\partial \psi}{\partial y} \quad \text{and} \quad v = -\frac{\partial \psi}{\partial x}.$$

Then

$$\psi = \int_0^y u\,\mathrm{d}\,y = \sqrt{U_\infty \nu x} f(s),$$

where f is an antiderivative function of g. It is now easy to compute u, v and all their derivatives as a function of f, and by replacing all the terms in (8.26), we obtain the *Blasius boundary layer equation*

$$f f'' + 2f^{(3)} = 0, \tag{8.30}$$

coming with boundary conditions

$$f(0) = f'(0) = 0$$
$$\lim_{s \to +\infty} f'(s) = 1$$

because $u = v = 0$ along the wall and $\lim\limits_{y \to +\infty} = u_\infty$. We have been solving this equation numerically and this is how we obtain our reference solution.

Let us come to the numerical test case. The domain is $[0;20]\times[0;10]$, the segment $\{y = 0 \wedge 0 \leqslant x < 13\}$ is a slip wall boundary while the rest of the line, $\{y = 0 \wedge 13 \leqslant x \leqslant 20\}$ is a non slip boundary. The incoming flow has Reynolds number 450 and Mach 0.1. At the upper boundary, we use a Steger-Warming boundary condition in order to reproduce the *far-field* state. What we do on the output edge is a bit complex. The flow not being homogeneous along the output line, we cannot enforce a global *far-field* state. But we know we have 3 outgoing characteristics out of 4. Then we just need to impose one variable on the ingoing characteristic, for example the pressure. The boundary condition consists finally into enforcing a right state, equal to the inner state at left but with a fixed pressure. The first order mesh has 15213 vertices and 30430 triangles. We have represented a zoom around the boundary layer of the computed solution on Figure 8.1. On this figure the x-velocity is represented in color, while the black isolines represent the Mach number contours. The quality of the solution is very good. But if we look at Figure 8.2, which represents the isolines of the density component, we see there are small problems. First, on the slip boundary, the isolines make a small hook in the last layer of elements. In fact, we observe this phenomenon in many other computations using the slip boundary conditions (Naca, Blunt airfoil,...) and we haven't find any strong explanation to this at that moment. It is definitely relied to the slip boundary condition formulation and we are convinced this is due to a wrong implementation of the boundary flux. Second, we see there are small problems along the output boundary. The isolines are not completely straight in the vicinity of the output and the isoline

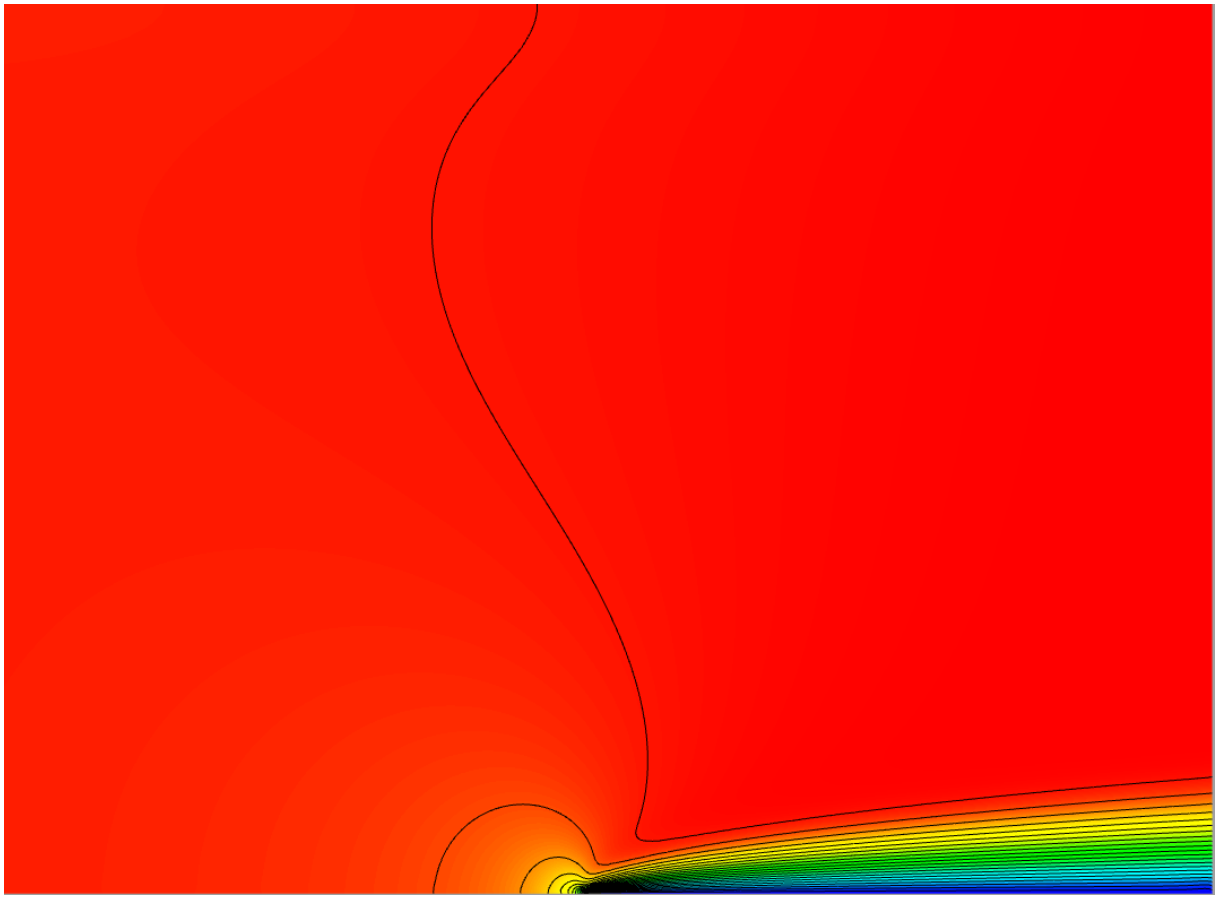


Figure 8.1: Third order solution of the Blasius problem. In color is represented the values of u the x-velocity. Isolines represent Mach number contours.

the nearer to the non-slip boundary reattach to the wall when it should not. This is also linked with the choice of the output boundary condition. The chosen formulation might not reproduce numerically the case of an infinite long non-slip wall and the boundary layer is modified in the vicinity of the output boundary.

We subsequently tried to compare the solution obtained with the computed exact solution inside the boundary layer. We have extracted the values of the second and third order solutions along the line $x = 17$ and plotted the dimensionless u profile with the expected exact profile. The result can be seen on Figure 8.4. The agreement is globally very good, especially inside the boundary layer, and the third order solution is a little bit better than the second order one in this region. However, we can observe an *overshoot* on both second and third order solutions, compared to the exact one. The assumptions leading to the Blasius equation (8.30) included in particular the fact that the domain were infinite in the y direction, which is of course not the case in our numerical simulation. Then, boundary condition $\lim_{y \to +\infty} = u_\infty$ is no more valid due to mass conservation. This could explain partly were the overshoot could come from. In order to assess this hypothesis, we have tried other third order computations for different Reynolds numbers. $Re = 450$ is the solution presented on Figure 8.4. We have represented the different profiles on Figure 8.5. First the axes have been changed. Instead of plotting the profile as a function of the

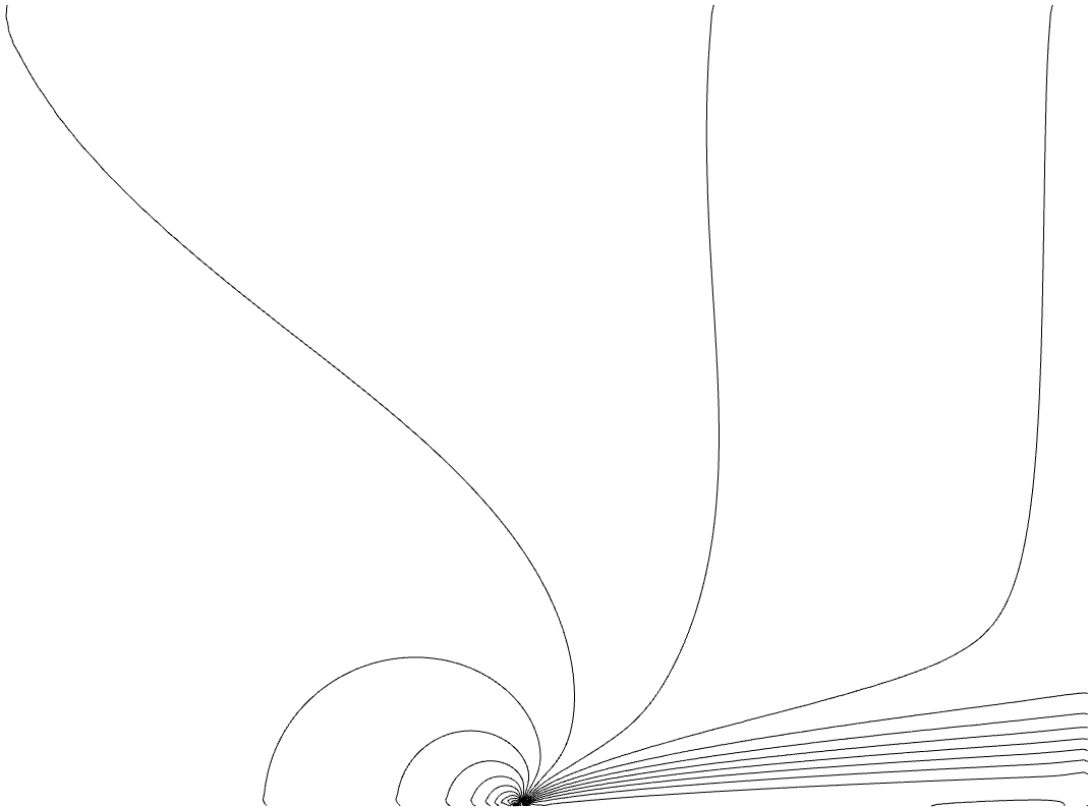


Figure 8.2: Third order solution of the Blasius problem. Density Isolines.

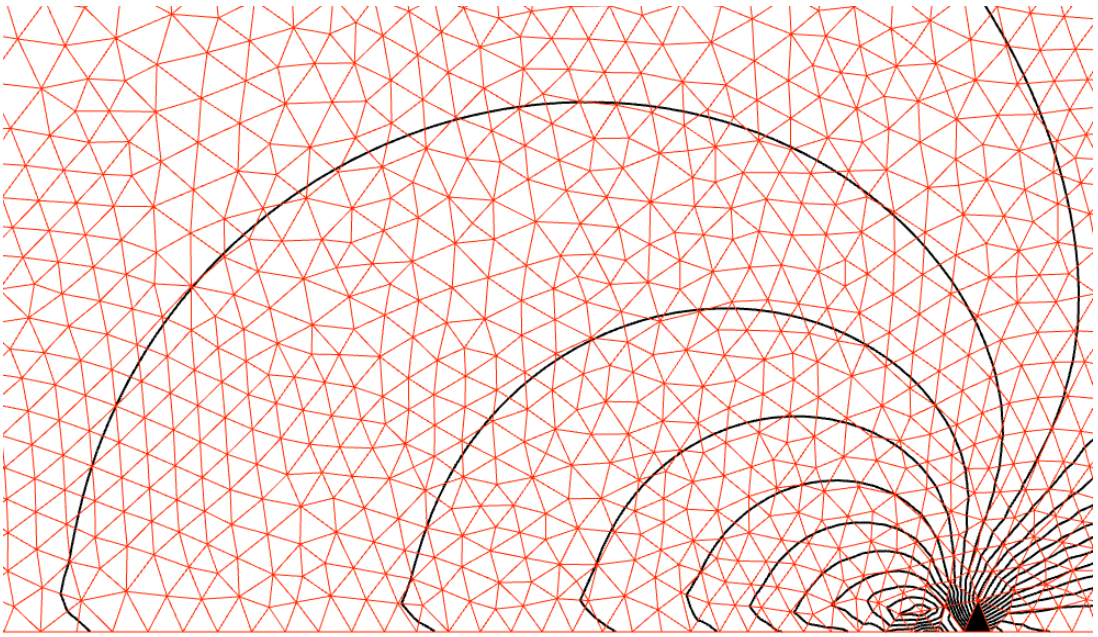


Figure 8.3: Zoom along the slip wall of the previous figure.

vertical coordinate y, we have used for abscissa the dimensionless distance to the wall $s = \frac{y}{\delta(x)}$. And instead of nondimentionalizing the x-velocity by the external velocity $u_e = u(17, 10)$, we have divided it by u_∞, the input velocity. We know that the lower the Reynolds is, the thicker the boundary layer is, the smaller s is at upper boundary $y = 10$ and the nearer to the boundary layer the boundary condition is applied. This is clear on graphic 8.5: by mass conservation the external velocity u_e must be greater than the input velocity u_∞, and it is even greater for smaller Reynolds numbers. Due to the upper boundary condition, the non-slip wall modifies the flow globally and this explain partly the obtained *overshoot*.

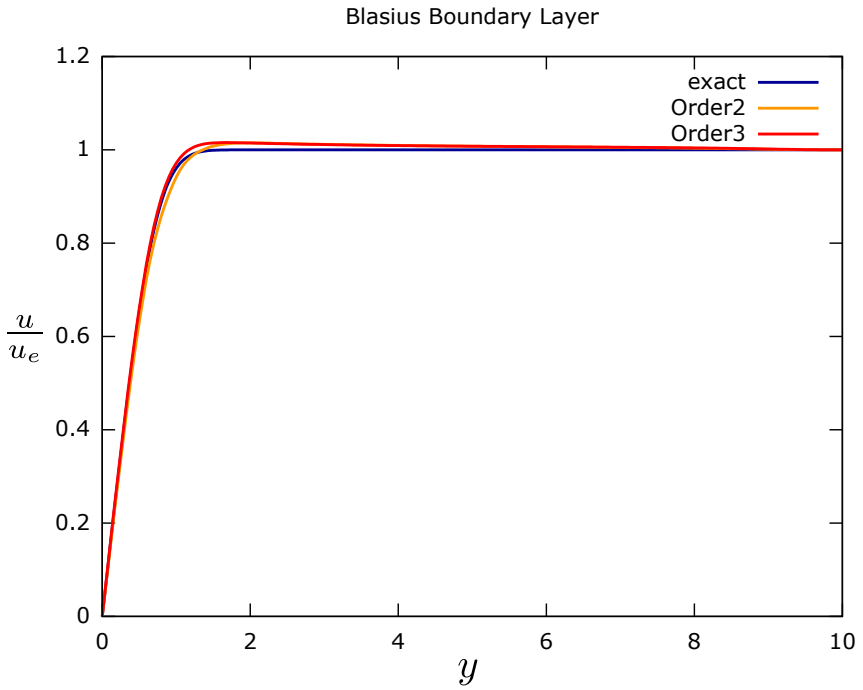


Figure 8.4: Second order, third order and exact x-velocity profile along the line $x = 17$ for the Blasius Problem.

8.5 Viscous NACA012 Test Case

We yet consider a viscous flow around a NACA012 airfoil. The flow parameters are the following:

- Incidence: 0° of incidence;
- Mach: $Ma = 0.5$;
- Reynolds: $Re = 500$.

This test case is known to be steady. We have run second and third order computations on 8 different meshes containing between 609 and 230×10^3 vertices. On Figure 8.7 are represented the horizontal velocity in color and the density isolines at third order for the finest mesh. We

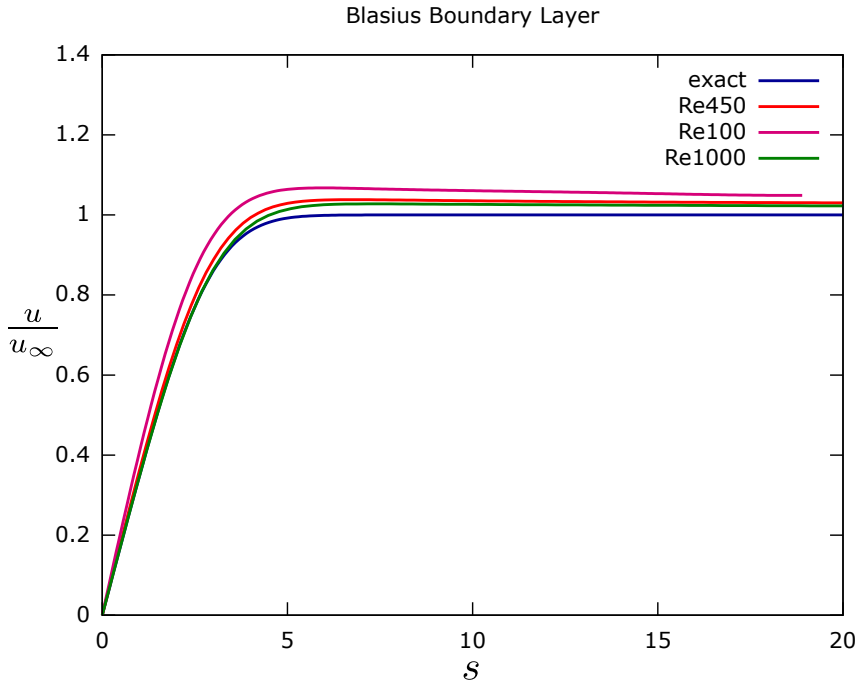


Figure 8.5: Third order and exact x-velocity profile along the line $x = 17$ for the Blasius Problem for different Reynolds number.

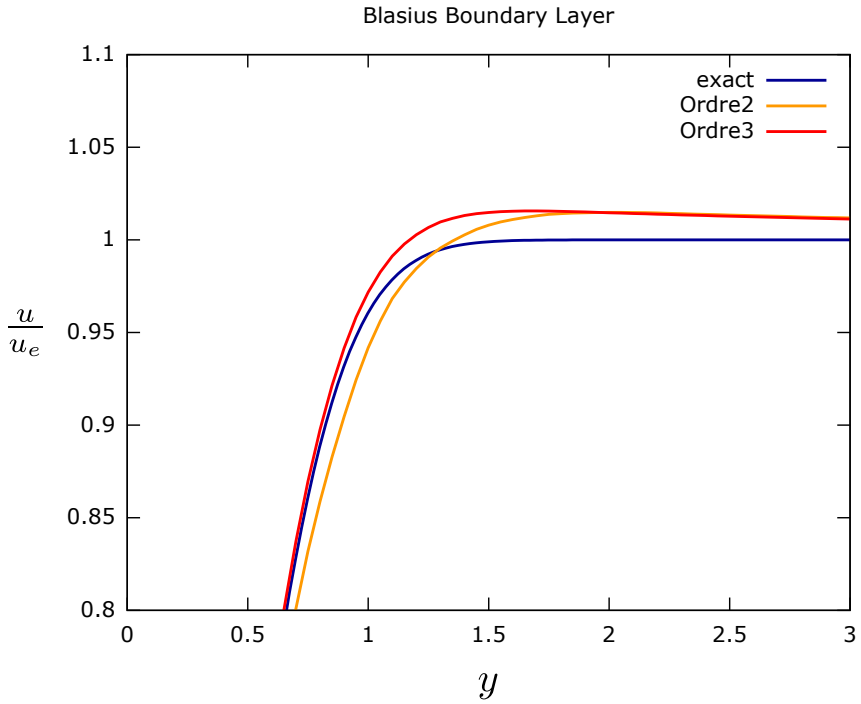


Figure 8.6: Detail of Figure 8.4.

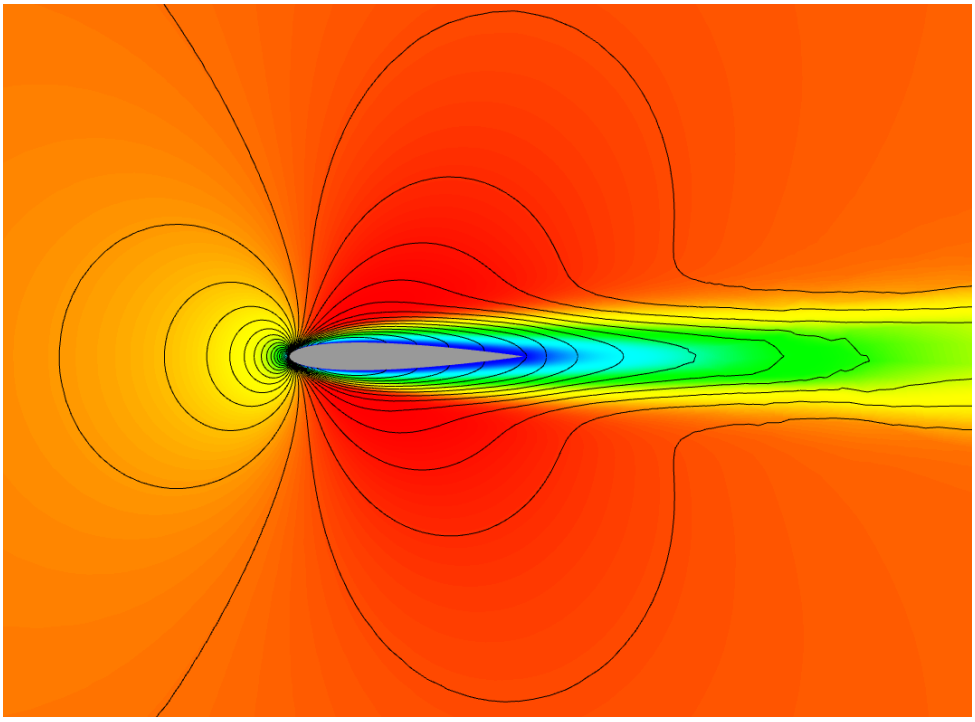


Figure 8.7: Third order solution on the finest mesh for the steady viscous NACA012 test case. x-velocity in color and isolines of the density component.

see that the global shape of the solution is the one expected, with the boundary layer around the airfoil and its wake. Now, because the incidence is null, the lift coefficient should be zero, but because the mesh is not symmetric, the numerical value of the lift coefficient is non zero. And it should converge to zero with the right order of convergence when the mesh gets finer. We have represented the value of the computed lift coefficients at steady state with respect to $h = \sqrt{\#\{\text{DoF}s\}}$ on Figure 8.8. Except for one strange value at second order for the 6^{th} mesh, all the second order estimated lift coefficients are larger in absolute value than their associated third order lift coefficients. Furthermore, the slope of the least square line is larger for the 3^{rd} order simulation than for the 2^{nd} order one. This means the third order scheme is doing a better job for viscous simulation. But on the other hand the slope is not the one expected. If 1.7 is a good result for the expected slope 2, 2.1 is a bit far from the slope 3 expected and it is clear that the convergence is not regular at all. The mix between the residual formulation of the advective term and the Galerkin treatment of the second order diffusive term does not seem to provide the right convergence rate. This might be explained by looking at the variational formulation of the problem. However, the final result is not so bad, because the mesh convergence is still acceptable and the solution is pretty nice.

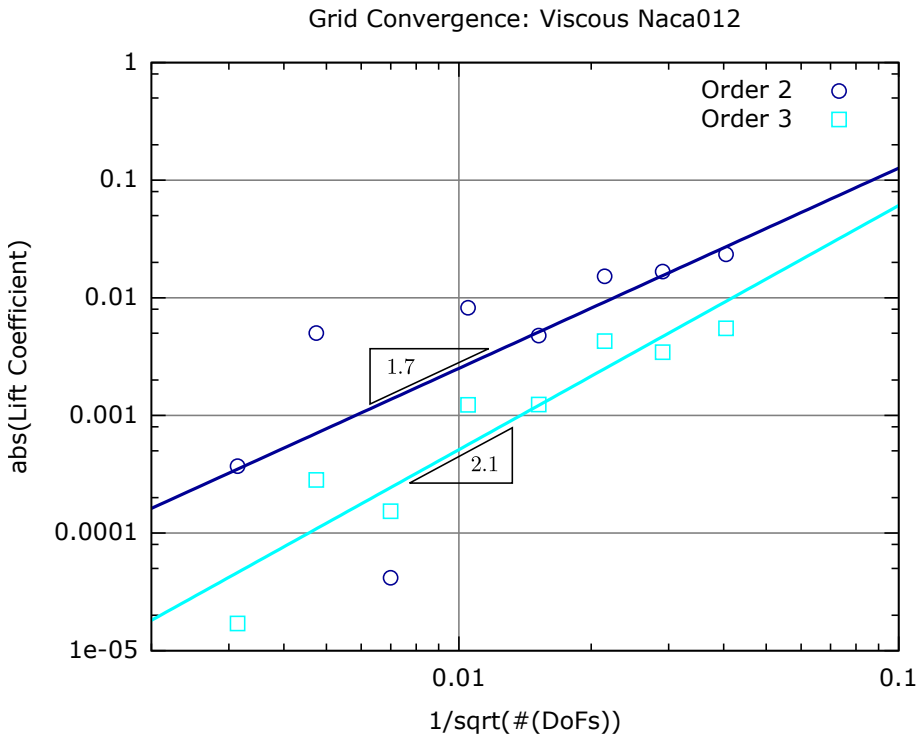


Figure 8.8: Convergence of the lift coefficient with respect to the mesh characteristic size $h = \sqrt{\#\{\text{DoF}s\}}$ for 2^{nd} and 3^{rd} order simulation of the viscous NACA012 problem.

Chapter 9

Conclusion and Perspectives

This thesis work has been devoted to the development and analysis of very high order non-oscillatory compact residual distribution schemes ($\mathcal{RDS}$) for the solutions of non linear conservation laws ($\mathcal{CL}$s) on unstructured hybrid meshes. The design methodology is the following:

1. Build a compact conservative linear monotone and stable $\mathcal{RDS}$ for scalar problems;
2. Define a geometrical discretization of the spatial domain that enables to build a k^{th} order continuous piecewise polynomial approximation of the variables in the domain. This discretization must handle with unstructured hybrid grids;
3. Design a *monotonicity preserving* procedure called "*limitation*" that recasts the first order linear scheme into a $(k+1)^{\text{th}}$ order one. This limitation technique also has to preserve the conservative character of the first order scheme;
4. Generalize the method to multidimensional non linear $\mathcal{CL}$s, as Euler or Navier-Stokes equation in our case.

The theoretical aspects of these design steps have been presented in a as rigorous manner as possible along all this manuscript. The theory has been then supported by numerous test cases. These numerical illustrations are first used to validate the scheme design and second allow to justify the straightforward generalization to non linear multidimensional $\mathcal{CL}$s, for which almost no theoretical results are available. All the numerical results given in this manuscript match well the expected behaviour and almost all of them show that the higher order discretization has greatly improved the solution for a limited calculation extra cost. Furthermore, the compact nearest-neighbours stencil of the scheme allowed us to parallelize the code quickly and successfully, so that huge problems such as a complete aircraft simulation, only take a couple of hours now on the computational cluster.

This conclusive chapter is divided as follows: in a first section, we briefly summarize the content of this manuscript and underline the main achievements of this thesis work. In a second section, we put the success of the high order $\mathcal{RD}$ schemes into perspective, discuss about their main weaknesses, and compare them with the other classical high order schemes for $\mathcal{CL}$s. Finally, we end this manuscript with a global overview of the work that has still to be done and of the possible future applications of such a class of numerical schemes.

9.1 Content Summary

9.1.1 Conservation Laws

This manuscript has started with a global recall of the basic theoretical results for Conservation Laws. Conservation Laws are the mathematical formulation of the first principle of thermodynamic for a given problem. It just claims "*nothing disappears, nothing suddenly appears, everything is transformed*".

We have begun by showing the classical approach of the PDEs is not sufficient because some very regular simple problems admit discontinuous solutions. Most of $\mathcal{CL}$s problems do not admit any solution in $\mathcal{C}^1$. Then, we have defined a larger class of solutions verifying a weaker form of the problem, that includes the class of the classical solutions. Unfortunately, the mathematical problem in its weaker form may admit an infinite number of solutions and is thus not well-posed. But if we look at the same problem perturbed by a small dissipative term, it is well-posed and moreover brings an extra condition enforcing the solution to respect some *entropy* constraint. The unique solution of the weak problem has been therefore defined as the limit of the perturbed problem for a decreasing dissipation coefficient and we have shown that this solution can in fact be sorted out from the infinity of weak solutions of the $\mathcal{CL}$ by an *entropy* criteria. This is exactly the second principle of thermodynamic.

We have been next interested in the class of $\mathcal{CL}$s that are diagonalizable and that we call *hyperbolic*. For this type of problems, we have seen that in every direction of the space the information propagates along propagating waves, and that we can define in space-time characteristic curves that rule the travel of the data. The information moves everywhere at finite speed and the value at one point of the space-time only depends on the values situated inside a *dependence cone*. This characteristic vision of the problem allowed us to give a theoretical look at the boundary conditions. The information crosses the boundary at the relative speeds of the propagating waves in the direction normal to the boundary. Applying boundary conditions for hyperbolic $\mathcal{CL}$s consists then in diagonalizing the problem in the direction normal to the boundary and in enforcing the solution only on the component of the entering waves.

Finally, in a last part, we have described the construction of the two systems of conservation laws for fluid mechanics: the Euler and Navier-Stokes equations. These systems are built by applying the conservation principle to the conserved variables: the mass, the momenta and the total energy. The very general conservative system has been reformulated for the case of a fluid by applying some restrictive constraints on the nature of the strain tensor and by using an equation of state of the gas dynamic, in order to close the problem. The Euler and Navier-Stokes equations differ just by the fact they do or not consider the viscous effects and the heat transfers.

9.1.2 High Order Discretization

In the second chapter, we have studied the solution approximation and the spatial discretization of the domain. It is a fact that, even for very simple problems, the exact continuous solution may not be known and has to be approximated. We do this today by defining a finite dimensional functional space that approximates the continuous functional space in the sense the projection of an element of the continuous functional space onto the finite dimensional space converges toward the considered element when the number of dimensions goes to infinity. We next show

that the components in a certain basis of the projection of the unknown continuous solution on the finite dimensional functional space are the unique solution of a non linear equation linked with the weak formulation of the $\mathcal{CL}$. This finite dimensional equation is now solved giving an approximation of the continuous solution which accuracy can be directly related to the nature of the finite dimensional functional space and its dimension.

The finite dimensional functional space is usually defined as the space of piecewise polynomial functions over a spatial discretization, the *mesh*. We have first started by the case when the approximated solution is piecewise linear, over triangles only, which is easy and classical because by 3 points of a 3D space passes a unique plane. At each *degree of freedom* of the mesh is associated a piecewise linear basis function and the finite dimensional functional space is spanned by these functions. We have then generalized this construction and defined basis functions over the triangular mesh that are piecewise polynomial of order k. This is what we have called the k^{th} order discretization. In order to handle with *hybrid* meshes, we have eventually detailed the construction of a k^{th} order polynomial discretization over quadrangle, $\mathbb{Q}^k$, that is compatible with the one defined over triangles.

In a last paragraph, we have discussed about the main advantages of the high order discretization. We have above all shown that for a given accuracy of the approximated solution, there always exists an optimal order of representation of the data such that the number of *degrees of freedom* in the mesh – and therefore the size of the associated algebraic problem – is minimized.

9.1.3 High Order Distribution Schemes

In Chapters 4 and 5, we have described in details the construction of the high order Residual Distribution Schemes and often linked it to its practical implementation.

After having briefly and very generally introduced the $\mathcal{RDS}$, we have first presented their main theoretical properties and explained the way they are ensured. In particular, we have started with the study of the consistency of the scheme with the continuous weak formulation of the *Conservation Law*. This is given in Theorem 4.4, page 68. We have then recalled the need of a *monotonicity preserving* formulation for stability of the numerical scheme and explained the way this property is enforced. Finally, we have studied the conditions under which the k^{th} order accuracy of the approximated solution is reached. It can be summarized as follows:

- The approximation of the data has to be of $(k+1)^{\text{th}}$ order and continuous;
- The distribution coefficients β_i^{T} must be all bounded by a constant ($\mathcal{LP}$ property);
- The scheme has to be non linear (Godunov).

Knowing the properties of the $\mathcal{RDS}$, we have given a non exhaustive list of the main $\mathcal{RDS}$ and compared them in term of theoretical behaviour. The N scheme is linear (thus first order), monotonicity preserving, conservative and upwind. By limiting its distribution coefficients, we obtain the PSI scheme that is now $\mathcal{LP}$. Unfortunately, the generalization of this almost perfect scheme to more than second order polynomial discretization does not seem to be possible. The LDA scheme which is $\mathcal{LP}$, conservative, upwind, but not monotonicity preserving has a possible generalization to high order discretization which is not simple. Looking at this, we have then

decided that for the study of high order $\mathcal{RDS}$, the simplest would be to use the *centered* Lax-Friedrichs scheme which is linear, monotonicity preserving and conservative. The main advantage of this scheme is that its formulation is very flexible and can be adapted to any polynomial order of approximation and any polyhedral spatial discretization. Furthermore, by using the limitation procedure that turns the N scheme into the PSI scheme, we get a $(k+1)^{\text{th}}$ order accurate scheme.

Considering exclusively the LxF version of the $\mathcal{RDS}$, we have then detailed the scheme implementation step by step, for scalar or multidimensional problems. The scheme starts by computing for each element the *global residual* (5.2). This quantity represents the amount of information that is leaving the element. This *global residual* is sent to the nodes of the element through the *nodal residuals* (5.8) which allow to define the *distribution coefficients* (5.11). The distribution coefficients are then limited in order to get the $\mathcal{LP}$ condition and the scheme is $\mathcal{LP}$, monotonicity preserving and conservative. The boundary conditions have been presented in two different classes. The most usual one and also the most correct is to define the weak formulation of the boundary condition and to impose some boundary flux along the input edges. But it is also practical to enforce sometimes the boundary conditions strongly. In that case, the values of the unknowns on the boundary are imposed at the beginning of the calculation and maintained all along the simulation. Finally, at the end of every iteration, the problem is solved by using either an *explicit* or an *implicit* method.

Unfortunately, the LxF scheme is a centered scheme and does not respect some physical *upwind* constraints. It is absolutely not sure that every downstream node is going to receive some signal from the distribution. After a short convergence, it happens that some unknowns can take any values in a given interval. The solution not uniquely defined and some spurious modes may appear even if the scheme is monotonicity preserving. We have overcome this problem by adding an extra dissipative term that has some upwind properties and that cures the ill-posedness. The origin of the problem and the computation of the cure term have been deeply studied throughout this thesis.

9.1.4 New Achievements

This thesis work has been looking at many different aspects of the resolution of Conservation Laws with High Order $\mathcal{RDS}$ and often brought some original contributions. At the beginning of this thesis, the theory for scalar problems was the same as the one presented here, but no simulation beyond 3^{rd} order of approximation could be realized here in Bordeaux and the computation could be done only on *triangulations*. For systems, only the $\mathbb{P}^1$ explicit and implicit with 1^{st} order Jacobians formulations for one and two dimensional Euler problems were available in *FluidBox*.

We have started this work by developing a new code that is today able to solve any scalar conservation law on hybrid meshes with an accuracy up to 4^{th} order. What has been shown with this code is that 2^{nd}, 3^{rd} and 4^{th} order can be reached by using their respective polynomial approximations. Moreover, quadrangular grids are very interesting because they contain up to twice less elements than their equivalent *triangulations* and the obtained solution is usually more accurate because $\mathbb{Q}^k$ functions are using cross terms that increase the accuracy of the approximation. This code has also permitted to look for new limitation techniques as the one illustrated on Figure 5.1, page 94. This limitation technique gives very good result and allows to get rid of the stabilization term, but it can be unfortunately applied only in the $\mathbb{P}^1$ scalar case.

After having validated the high order formulation on simple scalar problems, we could go further and try to generalize it to multidimensional problems. High order residual formulation of the Euler equations has been implemented into the INRIA Fortran platform for fluid simulations, *FluidBox*. In two dimensions, the code is today able to deal with hybrid meshes and piecewise quadratic representation. For most of the test cases, the advantage of the third order scheme is observed as it usually greatly improves the accuracy of the result for an equivalent computational effort. The numerical entropy creation is always much reduced with $\mathbb{P}^2$ approximation. However, the results are not perfect and sometimes far from the expected solution. The problems seeming to take their origin at the boundary, we have focused on the enforcement of the boundary conditions. Many formulations have been tried with relative success and we have presented here a higher order representation of the boundary edges by *isoparametrical elements*. It is half a success because the formulation is working and the numerical entropy is even more reduced, but the order of convergence is still not reached (see for example the *sphere problem* in Subsection 6.2.3). Finally, aiming at always accelerating the scheme convergence, we have been looking at improving the degree of approximation of the residual Jacobians. We have here shown that the Jacobians computed by finite differences are on one hand more expensive to build than the simple linearization of the LxF residual, but on the other hand so much improving the iterative convergence that their global advantage is clear. We have also tried to build the exact Jacobians which should theoretically even better improve the iterative convergence, but no test case has ever been positive and this experiment is nowadays a complete failure.

Three dimensional problems have been next considered. In 3D, the number of degrees of freedom is quickly very big and our sequential formulation (treated by a single processor) was not sufficient. The first challenge was then to *parallelize* the code. Thanks to the compactness of the $\mathcal{RDS}$, this has been achieved quite quickly, for a satisfying – even if not perfect – parallel efficiency. The $\mathbb{P}^1$ and $\mathbb{P}^2$ formulations over tetrahedrons have been then tested. The second order scheme is working on all kind of test cases and gives quite good results. Solution of a complete supersonic aircraft could be also computed. Unfortunately, the $\mathbb{P}^2$ formulation has still some difficulties and the comparison with higher order formulation has not been completed.

In the last chapter of this manuscript, we have presented our recent results for the simulation of viscous test cases. In our scheme, the viscous terms of the Navier-Stokes equations have been discretized by a Galerkin formulation. In a first part of Chapter 8, we have proved the consistency of this formulation with the weaker form of the Navier-Stokes equations. Then the scheme is converging toward an approximation of the solution of the continuous viscous problem. However, it is generally agreed that the obtained accuracy is not maximal. It is likely the global scheme (Residual formulation of the advective terms plus Galerkin formulation of the diffusive terms) looses an order of accuracy when dealing with finer grids. But as we have seen for the two dimensional Blasius boundary layer, page 170, and for the viscous NACA test case page 174, the higher order formulation is still improving the global result and the mesh convergence slope. At the end we can say that even if not expected, this formulation gives good results as it is and seems to be promising for the future.

9.2 Weaknesses of the High Order $\mathcal{RDS}$

All along this manuscript, even if some drawbacks of the High Order Residual Distribution schemes have been revealed, we have much underlined the advantages and further possibilities

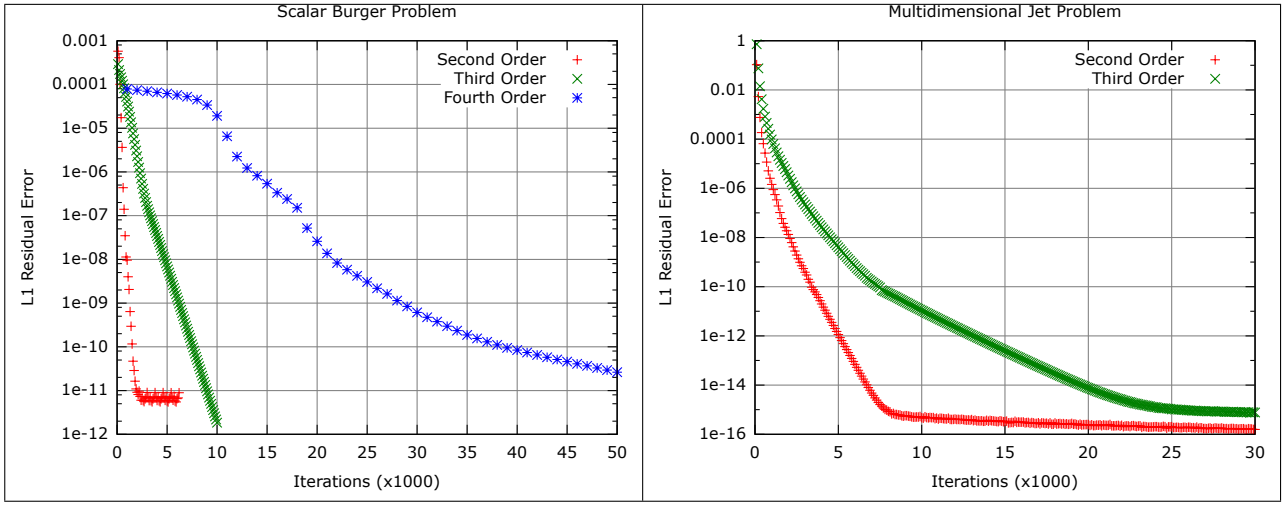


Figure 9.1: Iterative convergences for two similar problems. The left one is related to a scalar Burger problem, while the right one concerns a vectorial jet problem. Both solutions present shocks.

of such schemes and not talked so much about its weaknesses. We devote this whole section to a fair critics of the High Order $\mathcal{RDS}$ and more particularly of the Lax-Friedrichs version we have been developing since chapter 5.

9.2.1 Iterative Convergence

It is a fact that higher order schemes enable to reach higher accuracy with a reduced number of elements and DoFs. We have been proving this all along the manuscript. But on the other hand, High Order $\mathcal{RDS}$ suffer from much slower iterative convergence and they seem to have more difficulties to reach machine zero as the polynomial order of representation of the data increases. Then for a given sufficient high accuracy, all the time that would have been gained by using a very high order scheme needing much less degrees of freedom than a second order scheme and thus much less time per iteration is lost through the increased number of iterations needed to get full convergence. On Figure 9.1, we have represented on the left the explicit iterative convergence of the scalar version of the LxF scheme for second, third and fourth order. On the right is plotted the convergence history for a $\mathbb{P}^1$ and $\mathbb{P}^2$ scheme applied on a vectorial problem and using an implicit solving procedure with first order linear Jacobians. It is here obvious that the higher the polynomial order of representation is, the slower the scheme converges toward machine zero. And the convergence slope losses approximately a factor two for any increase of one unit of the order of representation of the data. That is a pretty big problem. There is nowadays no solution to remedy to this problem, but we can however give some ideas. We have been speaking of the finite difference Jacobians for the implicit matrix computation.

9.2.2 Boundary Conditions

Another weakness of the $\mathcal{RDS}$ that have already been underlined during this manuscript is the treatment of the boundary conditions. We have already seen that for a high order scheme,

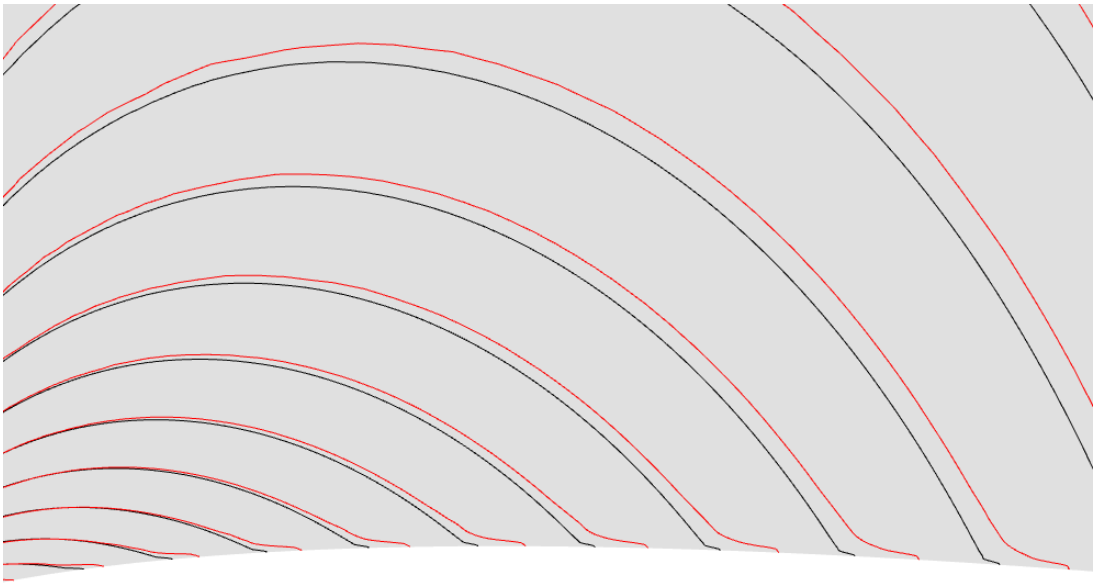


Figure 9.2: Isolines of density of the second (red) and third (black) order solution of Mach 0.5 inviscid NACA012 test case. Detail of the slip wall boundary.

the order of representation of the boundary edges is not sufficient and seems to reduce the mesh convergence order, especially when considering quantities that are strongly bounded to the boundaries, as the lift, drag and momentum coefficients or the entropy deviation. However, in Subsection 6.2.4 we have developed a way of representing the boundary edges with high order accuracy in 2D. Unfortunately, this treatment does not provide the expected results: it is indeed improving the global solution, in particular in term of entropy production, but on the other hand, the mesh convergence slope is not very much improved. Then, if the wrong mesh convergence is almost independent of the boundaries representation, we can suppose that it is related directly to the way we enforce the boundary values. Copying what is done for the Galerkin Finite Element methods as presented in Section 5.4 might not be very well suited for $\mathcal{RD}$ schemes. On Figure 9.2, we have represented a close zoom to the upper boundary edge of a NACA012 airfoil and the isolines of the $\mathbb{P}^1$ and $\mathbb{P}^2$ solution of a Mach 0.5 inviscid flow. The isolines should be everywhere orthogonal to the boundary edge which is not the case here. The isolines bend strongly in the last level of elements as if there was a boundary layer there which is obviously not the case because it is a Euler solution. It is likely that the boundary treatment provides an extra entropy production along the profile whatever the order of representation of the edges is. It is moreover a phenomena that is observed only for $\mathcal{RDS}$ and not for $\mathcal{DG}$ or $\mathcal{FV}$ for example.

In order to go further into the resolution of aeronautical problems by High Order $\mathcal{RDS}$, it seems obvious to find a global residual formulation of the projected weak problem, so that the boundary conditions stay consistent with the inner scheme and the right mesh convergence is observed for all the possible parameters of problem.

9.2.3 Stabilization Term

In the case of the Lax-Friedrichs $\mathcal{RDS}$, we have seen that the high order formulation is ill-posed. This is due to the centered character of the first order scheme: some values are not uniquely defined. At that time, we overpass this difficulty by adding an extra *stabilization term* that enables the scheme to fully converge. Even if many efforts have been given throughout this thesis to improve our understanding of this term and to compute it well, this solution is still not satisfying. The main reasons are:

- This term destroys the monotonicity preserving property of the LLxF scheme. Using this term as it is leads to solution with over/undershoots in the vicinity of discontinuities. In order to limit this effect, we have to use an additional shock capturing term that is usually defined intuitively and that has often to be fitted to each test case. This goes exactly the opposite way of a completely parameter-free scheme. Furthermore, even when using the shock capturing term, monotonicity is still not ensured and nothing prevents the solution of some tough test cases to blow up suddenly;
- The implementation of this extra term is rather complex and its computation may be costly especially in the case of quadrangles where we have to use all the degrees of freedom of the element and to reconstruct the residual Jacobians at these nodes;
- This term does not take place into a global formulation for the approximation of the *Conservation Laws*. It is added to the residual formulation in order to fix the ill-posedness of the simple LLxF scheme. It can be quite rightly considered as a simple patch to counterbalance the default of the LLxF scheme.

Many tries have been made to get rid of this *stabilization term*, each time without success. We have been looking for new limitations that would give some upwind property to the scheme but when a solution is found for one situation, its generalization to other situations is always impossible. A global residual distribution formulation is now needed to find a scheme that would combine all the advantages of the schemes presented in Section 4.4.

9.2.4 Navier-Stokes Global Formulation

As we have seen in Chapter 8, we are today able to perform Navier-Stokes simulations, but we are in fact discretizing the viscous term with a Galerkin residual. Even if the global formulation using the $\mathcal{RDS}$ framework for the Euler part of the equation and a Galerkin formulation of the viscous terms stays consistent with the initial conservation law, it seems that the optimal $(k+1)^{\text{th}}$ order is lost for the finest meshes. A global residual formulation of the Navier-Stokes equations is needed.

9.3 Perspectives

In order to conclude this manuscript and to see the problem beyond the drawbacks of the method that have just been described, we present in this section a non exhaustive list of the possible perspectives for the Residual Distribution Schemes. The very high order discretization

of the conservation laws used in this thesis has shown a potential that certainly justifies its further development. There are still several fronts that greatly need to be developed. We detail them in the following paragraphs.

Unsteady Case

Even if a little bit of the unsteady terms treatment has been discussed in Chapter 3, no unsteady results have been presented in this manuscript. It is however an important part of the work that has been started at INRIA Bordeaux Sud-Ouest. Unfortunately, scalar results are at that time not very satisfying and that is why the unsteady case has been somehow forgotten in this manuscript. There are today two main ways to treat the unsteady test cases. First, the unsteady *Conservation Law* (2.1), page 16, can be seen as a steady equation in space-time. By using a $\mathcal{RD}$ framework within the prismatic elements described in Subsection 3.2.3, page 55, we hope to obtain a scheme that is both $(k+1)^{\text{th}}$ order accurate in space and $(\ell+1)^{\text{th}}$ in time. Second, the unsteady terms can be first discretized by finite differences (as a Runge-Kutta method or anything else giving the desired accuracy in time) and the $\mathcal{RD}$ formalism is next applied. We then obtain a formulation that is very similar to the steady case, just adding a time dependent source term in the right hand side. The main advantage of this second formulation is a big reduction of unknowns with respect to the first one. Only the space is meshed when the a space-time domain is discretized in the first case.

Viscous Term Treatment Improvement

As we have just said in the previous section, the formulation we have presented in this manuscript suffers from a lack of consistency between the $\mathcal{RD}$ formulation of the Euler terms and the Galerkin approximation of the viscous terms. This leads to the loss of an order of accuracy when looking at the mesh convergence for rather fine meshes. One wishes then to find an approximation of the Navier-Stokes equations that would be globally more consistent. A smart solution developed by Nishikawa [75, 74] is to add the gradients of the solution as extra unknowns. Thus, the $\mathcal{RD}$ formalism can be applied directly on the Navier-Stokes equations and the $(k+1)^{\text{th}}$ order of accuracy is expected. The main inconvenient of this method is the doubling of the amount of unknowns per degree of freedom. But on the other hand, the method has the maximal order of accuracy and is still maximal compact, so that it can be easily very efficiently parallelized. The numerical mathematics being today very much interested in the development of computations on very large clusters, this method is very promising.

Turbulent Cases

Residual Distribution Methods will have a real future when their applications to industrial problems will be possible. Among the many gaps that still need to be filled, the simulation of turbulent test cases is mandatory. To do so, the correct $\mathcal{RDS}$ discretization of some turbulent models has to be assessed and included in the code.

Low Mach / Incompressible Flows

Speaking about the industrial applications of the $\mathcal{RDS}$, it would be also very interesting to look at the behaviour of the scheme for Low Mach or even Incompressible flows. It is well known that the conditioning of the algebraic problem becomes worse with a decreasing Mach number. Oscillations appear in the numerical solution when the Mach number lowers under some threshold. Many articles in the literature give then recipes to overcome this problem and it seems possible to apply some of them to the $\mathcal{RD}$ framework. Preliminary results on the design of wind power plants are already available and give surprisingly good results.

Other Applications

At INRIA Bordeaux Sud-Ouest, the $\mathcal{RD}$ framework is also applied to problems that are not fluid mechanics problems. For example, the method can be applied on Magnetohydrodynamics problems (MHD), such as reentry problems, or Aeroacoustics problems (noise generation), or even Geophysics problems (sismic waves propagation), *etc...* The development of such a method on different problems that imply different outlooks is necessary for the global improvement and the global assessment of the method.

Bibliography

[1] J.-L. Guermond A. Ern. Eléments finis : théorie, application, mise en œuvre. Springer, 2002. Written in French, English Translation available under title *Theory and practice of finite elements*.

[2] R. Abgrall. Toward the ultimate conservative scheme : Following the quest. J. Comput. Phys, 167(2):277–315, 2001.

[3] R. Abgrall. Essentially non oscillatory residual distribution schemes for hyperbolic problems. J. Comput. Phys, 214(2):773–808, 2006.

[4] R. Abgrall. Méthodes variationnelles pour les problèmes hyperboliques et paraboliques, application à la mécanique des fluides. École Matméca, Bordeaux, France, 2006.

[5] R. Abgrall. Residual distribution schemes: Current status and future trends. Computers and Fluids, 35(7):641–669, 2006.

[6] R. Abgrall and F. Marpeau. Residual distribution schemes on quadrilateral meshes. Journal of Scientific Computing, 30(1):131–175, 2007.

[7] R. Abgrall and M. Mezine. Construction of second order accurate monotone and stable residual distribution schemes for unsteady flow problems. Journal of Computational Physics, 188:16–55, 2003.

[8] R. Abgrall and M. Mezine. Construction of second-order accurate monotone and stable residual distribution schemes for steady flow problems. Journal of Computational Physics, 195:474–504, 2004.

[9] R. Abgrall and P.L. Roe. High order fluctuation schemes on triangular meshes. Journal of Scientific Computing, 2003.

[10] Ricchiuto M. Abgrall R., Larat A. Construction of High Order Residual Distribution Schemes. CFD Review, 2009. Accepted.

[11] G. S. Almasi and A. Gottlieb. Highly parallel computing. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1989.

[12] A. Athanasiadis. Three-dimensional hybrid grid generation and application to Navier-Stokes computations. PhD thesis, Université Libre de Bruxelles, 2004.

[13] C.-W. Shu B. Cockburn. The runge-kutta discontinuous galerkin method for conservation laws v: Multidimensional systems. J. Comput. Phys., 141(2):199–224, 1998.

[14] S.-Y. Lina B. Cockburn and C.-W. Shu. Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws iii: One-dimensional systems. J. Comput. Phys., 84(1):90–113, 1989.

[15] T.J. Barth. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations. in VKI LS 1994-05, Computational Fluid Dynamics Course, von Karman Institute for Fluid Dynamics, 1994.

[16] T.J. Barth. Numerical methods for gasdynamic systems on unstructured meshes. In Kröner, Ohlberger, and Rohde, editors, An Introduction to Recent Developments in Theory and Numerics for Conservation Laws, volume 5 of Lecture Notes in Computational Science and Engineering, pages 195–285. Springer-Verlag, Heidelberg, 1998.

[17] T.J. Barth and H. Deconinck, editors. High-Order ENO and WENO schemes for computational fluid dynamics, volume 9 of Lecture Notes in Computational Science and Engineering. Springer-Verlag, Heidelberg, 1999.

[18] T.J. Barth and H. Deconinck, editors. Error estimation and adaptive discretization methods in CFD, volume 25 of Lecture Notes in Computational Science and Engineering. Springer-Verlag, Heidelberg, 2003.

[19] T.J. Barth and P.O. Frederickson. High-order solution of the Euler equations on unstructured grids using quadratic reconstruction. AIAA paper 90-0013, January 1990. 28th AIAA Aerospace Sciences Meeting, Reno, Nevada (USA).

[20] T.J. Barth and D.C Jespersen. The design and application of upwind schemes on unstructured meshes. AIAA paper 89-0355, January 1989. 27th AIAA Aerospace Sciences Meeting, Reno, Nevada (USA).

[21] T.J. Barth and M. Ohlberger. Finite volume methods: foundation and analysis. In E. Stein, R. de Borst, and T.J.R. Hughes, editors, Encyclopedia of Computational Mechanics. John Wiley & Sons, Ltd., 2004.

[22] P.R.H Blasius. Grenzschichten in flüssigkeiten mit kleiner reibung. Z. Math u. Phys., 56:1, 1908. English Translation NACA TM 1256.

[23] J. Botsis and M. Deville. Mćanique des milieux continus : une introduction. Presses Polytechniques et Universitaires Romandes (PPUR), 2006.

[24] P. Brenner. The cauchy problem for symmetric hyperbolic systems in $\mathcal{L}^p$. Math. Scan, 19:27–37, 1966.

[25] Johnson C. and Nävert U. An analysis of some finite element methods for advection-diffusion problems. Analytical and numerical approaches to asymptotic problems in analysis (Proc. Conf., Univ. Nijmegen, Nijmegen, 1980), 47:99–116, 1981.

[26] S. Candel and M. Barrere. Mécanique des fluides : Cours. Dunod, Paris, 2001.

[27] P.J. Capon. Adaptive Stable Finite Element Methods for the Compressible Navier-Stokes Equations. PhD thesis, University of Leeds, 1995.

[28] J.-C. Carette, H. Deconinck, H. Paillère, and P.L. Roe. Multidimensional upwinding – its relation to finite elements. Int. J. Numer. Meth. Fl., (8/9):935–955, 1995.

[29] INRIA CEA, CNRS. Contrat de licence de logiciel libre cecill-c, 2006. http://www.cecill.info/licences/Licence_CeCILL-C_V1-fr.html.

[30] Alain Lerat Christophe Corre. High-order residual-based compact schemes for advection-diffusion problems. Computers and Fluids, 37(5):505–519, 2008.

[31] Alain Lerat Christophe Corre, Fabrice Falissard. High-order residual-based compact schemes for compressible inviscid flows. Computers and Fluids, 36(10):1567–1582, 2007.

[32] Xi Du Christophe Corre. A residual-based scheme for computing compressible flows on unstructured grids. Computers and Fluids, 38(7):1338–1347, 2009.

[33] Jean-François Remacle (Catholic University of Louvain) Christophe Geuzaine (University of Liège). Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, 2009. http://www.geuz.org/gmsh/.

[34] Jean-François Remacle (Catholic University of Louvain) Christophe Geuzaine (University of Liège). Gmsh documentation, 2009. http://www.geuz.org/gmsh/doc/texinfo/gmsh.pdf.

[35] P. Cinella. Roe-type schemes for dense gas flow computations. Computers and Fluids, 35:1264–1281, 2006.

[36] P. Cinella and E.A. Luke. Numerical simulations of mixtures of fluids using upwind algorithms. Computers and Fluids, 36:1547–1566, 2007.

[37] Grossman B. Cinnella P. Computational methods for chemically reacting flows. Handbook of fluid dynamics and fluid machineries, pages 1541–1590, 1996.

[38] Á. Csík, H. Deconinck, and S. Poedts. Monotone residual distribution schemes for the ideal magnetohydrodynamics equations on unstructured grids. AIAA Journal, 39(8):1532–1541, 2001.

[39] P. De Palma, G. Pascazio, G. Rossiello, and M. Napolitano. A second-order-accurate monotone implicit fluctuation splitting scheme for unsteady problems. J. Comput. Phys., 208(1):1–33, 2005.

[40] H. Deconinck, K. Sermeus, and R. Abgrall. Status of multidimensional upwind residual distribution schemes and applications in aeronautics. In AIAA CFD Conference, pages 2000–2328, 2000.

[41] Jiří Dobeš and Herman Deconinck. Second order blended multidimensional upwind residual distribution scheme for steady and unsteady computations. J. Comput. Appl. Math., 215(2):378–389, 2008.

[42] F. Dubois and P. Le Floch. Boundary conditions for nonlinear hyperbolic conservation laws. In Second International Conference on Hyperbolic Problems, Aachen, Germany, 1988.

[43] G. Alonso E. Valero, J. de Vicente. The application of compact residual distribution schemes to two-phase flow problems. Computers and Fluids. In Press, Corrected Proof, Available online 12 June 2009.

[44] S. Rebay F. Bassi. A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier-stokes equations. J. Comput. Phys., 131(2):267–279, 1997.

[45] S. Rebay F. Bassi. High-order accurate discontinuous finite element solution of the 2d euler equations. J. Comput. Phys., 138(2):251–285, 1997.

[46] A.C. Galeão and E.G Dutra do Carmo. A consistent approximate upwind petrov-galerkin method for convection dominated problems. Comp. Meth. Appl. Mech. Engrg., 68, 1989.

[47] M. Gisclon. Étude des conditions aux limites pour des systèmes strictement hyperboliques, via l'approximation parabolique. PhD thesis, École Normale Supérieure de Lyon, France, 1994.

[48] E. Godlewski and P.A. Raviart. Hyperbolic systems of conservation laws. Ellipses, Paris, 1991.

[49] E. Godlewski and P.A. Raviart. Numerical approximation of hyperbolic systems of conservation laws, Applied Mathematical Sciences. Springer, New York, 1996.

[50] S. K. Godunov. A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. Mat. Sb., 47, 1959.

[51] J. Goodman and Z. Xin. Viscous limits for piecewise smooth solutions to systems of conservation laws. Arch. Rational Mech. Anal., 121(3):235–265, 1992.

[52] M. Ricchiuto H. Deconinck. Residual distribution schemes: foundations and analysis. In E. de Borst E. Stein and T.J. Hughes Ed.s, editors, Encyclopedia of Computational Mechanics, Volume 3: Fluids. John Wiley and Sons, Ltd, 2007.

[53] R. Abgrall H. Guillard. Modélisation numérique des fluides compressibles. Elsevier, Paris, France, 2001.

[54] A. Harten. On the symmetric form of systems of conservation laws with entropy. J. Comput. Phys., 49:151–164, 1983.

[55] M. Hubbard and A.L. Laird. High order fluctuation splitting schemes for time-dependent advection on unstructured grids. Computers and Fluids, 34(4/5):443–459, 2005.

[56] M. E. Hubbard. A framework for discontinuous fluctuation distribution. International Journal for Numerical Methods in Fluids, 56:1305–1311, March 2008.

[57] M. E. Hubbard, M. J. Baines, and P. K. Jimack. Consistent dirichlet boundary conditions for numerical solution of moving boundary problems. Appl. Numer. Math., 59(6):1337–1353, 2009.

[58] Matthew Hubbard. Non-oscillatory third order fluctuation splitting schemes for steady scalar conservation laws. J. Comput. Phys., 222(2):740–768, 2007.

[59] Matthew Hubbard. Discontinuous fluctuation distribution. J. Comput. Phys., 227(24):10125–10147, 2008.

[60] T.J.R. Hughes, L.P. Franca, and M. Mallet. A new finite element formulation for CFD I: symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics. Comp. Meth. Appl. Mech. Engrg., 54:223–234, 1986.

[61] T.J.R. Hughes and M. Mallet. A new finite element formulation for CFD IV: a discontinuity-capturing operator for multidimensional advective-diffusive systems. Comp. Meth. Appl. Mech. Engrg., 58:329–336, 1986.

[62] Équipe TROPICS INRIA Sophia Antipolis Méditérranée. Tapenade, on-line automatic differentiation engine, 2009. `http://tapenade.inria.fr:8080/tapenade/index.jsp`.

[63] M. Ricchiuto J. Dobes and H. Deconinck. Implicit space-time residual distribution method for unsteady laminar viscous flows. Comp. & Fluids, 34(4-5):593–615, 2005.

[64] C. Johnson. Numerical Solution of Partial Differential equations by the Finite Element method. Cambridge University Press, 1987.

[65] J.O. Langseth and R.J. LeVeque. Wave propagation method for three-dimensional hyperbolic conservation laws. J. Comput. Phys., 165:126–166, 2000.

[66] Randall J. LeVeque. Finite Volume Methods for Hyperbolic Problems. Cambridge Texts in Applied Mathematics, 2002.

[67] R.J. LeVeque. Wave propagation algorithms for multi-dimensional hyperbolic systems. J. Comput. Phys., 131:327–353, 1997.

[68] M. Mezine. Conception de schḿas distributifs pour l'aérodynamique stationnaire et instationnaire. PhD thesis, Université de Bordeaux I, 2002.

[69] M.S. Mock. Systems of conservation laws of mixed type. J. Diff. Eqns., 37:70–88, 1980.

[70] G.E Moore. Cramming more components onto integrated circuits. Electronics, 38(8), 1965.

[71] J.R. Munkres. Elements of algebraic topology. Addison-Wesley, 1984.

[72] Brooks A. N. and Hughes T. J. R. A multidimensional upwind scheme with no crosswind diffusion. Finite element methods for convection dominated flows (Papers, Winter Ann. Meeting Amer. Soc. Mech. Engrs., New York, 1979), 34:19–35, 1979.

[73] Brooks A. N. and Hughes T. J. R. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. Computer Methods appl. Mech. Engin., 32:199–259, 1982.

[74] Hiroaki Nishikawa. A first-order system approach for diffusion equation. ii: Unification of advection-diffusion. J. Comput. Phys. Submitted.

[75] Hiroaki Nishikawa. A first-order system approach for diffusion equation. i: Second-order residual-distribution schemes. J. Comput. Phys., 227(1):315–352, 2007.

[76] H. Paillère. Multidimensional upwind residual distribution schemes for the Euler and Navier-Stokes equations on unstructured grids. PhD thesis, Université libre de Bruxelles, 1995.

[77] F. Pellegrini. Scotch official repository, 2006. `http://gforge.inria.fr/projects/scotch/`.

[78] F. Pellegrini. Scotch and libscotch 5.1 user's guide, 2008. `http://gforge.inria.fr/docman/?group_id=248`.

[79] F. Pellegrini. Scotch web page, 2008. `http://www.labri.fr/perso/pelegrin/scotch/scotch_en.html`.

[80] Ciarlet P.G. and Raviart P.-A. Interpolation theory over curved elements, with applications to finite element methods. Computer Methods appl. Mech. Engin., 1:217–249, 1972.

[81] C. Tavé. R. Abgrall. Construction of very high order residual distribution schemes. In Springer Verlag H. Deconinck, E. Dick, editor, Proceeding of the 6th International Conference on CFD, 2006.

[82] C. Tavé R. Abgrall, M. Ricchiuto and A. Larat. Non-oscillatory very high order residual distribution schemes for steady hyperbolic conservation laws : preliminary results. 14th International Conference on Finite Elements in Flow Problems, March 2007.

[83] M. Ricchiuto R. Abgrall and A. Larat. Construction of very high order residual distribution schemes for compressible flow problems. ISIS18, july 2008.

[84] M. Ricchiuto R. Abgrall and A. Larat. Non-oscillatory very high order residual distribution schemes for steady hyperbolic conservation laws. ECCOMAS 2008, july 2008.

[85] M. Ricchiuto R. Abgrall and A. Larat. Very high order residual distribution schemes for steady flow problems. ICCFD5, july 2008.

[86] R. Huart R. Abgrall. Simulation of mhd compressible flows by a residual distribution scheme. Numerical Flow Models for Controlled Fusion, April 2007.

[87] P.A. Raviart and Thomas. Introduction à l'analyse numérique des équations aux dérivées partielles. Dunod, 2004.

[88] Adam Larat Rémi Abgrall and Mario Ricchiuto. Construction of very high order residual distribution schemes for compressible flow problems. In ICOSAHOM'09, Trondheim, Norway, june 2009.

[89] M. Ricchiuto. Construction and Analysis of Compact Residual Discretizations for Consevation Laws on Unstructured Meshes. PhD thesis, Von Karman Institute for Fluid Dynamics, 2005. Available at `http://www.math.u-bordeaux.fr/~ricchiuto/Thesis_mario.pdf.gz`.

[90] M. Ricchiuto and R. Abgrall. Stable and convergent residual distribution for time-dependent consarlation laws. ICCFD4, 4th International conference on computational fluid dynamics, July 2006.

[91] M. Ricchiuto, R. Abgrall, and H. Deconinck. Application of conservative residual distribution schemes to the solution of the shallow water equations on unstructured meshes. J. Comput. Phys., 222(1):287–331, 2007.

[92] M. Ricchiuto and A. Bollermann. Accuracy of stabilized residual distribution for shallow water flows including dry beds. In HYP08, 12th international conference on hyperbolic problems, Maryland (USA), June 2008.

[93] M. Ricchiuto, N. Villedieu, R. Abgrall, and H. Deconinck. On uniformly high-order accurate residual distribution schemes for advection-diffusion. J. Comput. Appl. Math., 215(2):547–556, 2008.

[94] Mario Ricchiuto and Andreas Bollermann. Stabilized residual distribution for shallow water simulations. J. Comput. Phys., 228(4):1071–1115, 2009.

[95] Mario Ricchiuto, Árpád Csík, and Herman Deconinck. Residual distribution for general time-dependent conservation laws. J. Comput. Phys., 209(1):249–289, 2005.

[96] R. Abgrall A. Lerat M. Ricchiuto and C. Tavé. A simple construction of very high order non-oscillatory compact schemes on unstructured meshes. Computers and Fluids, 38(7):1314–1323, 2009.

[97] P. L. Roe. Linear advection schemes on triangular meshes. Technical Report CoA 8720, Cranfield Institue of Technology, 1987.

[98] P.L. Roe. Approximate riemann solvers, parameter vectors, and difference schemes. J. Comput. Physics, 43:357–372, 1981.

[99] P.L. Roe. Fluctuations and signals - a framework for numerical evolution problems. In K.W. Morton and M.J. Baines, editors, Numerical Methods for Fluids Dynamics, pages 219–257. Academic Press, 1982.

[100] P.L. Roe. Fluctuations and signals, a framework for numerical evolutions problems. In Numerical Methods for Fluid Dynamics, 1984.

[101] G. Rossiello, P. De Palma, G. Pascazio, and M. Napolitano. Third-order-accurate fluctuation splitting schemes for unsteady hyperbolic problems. J. Comput. Phys., 222(1):332–352, 2007.

[102] G. Rossiello, P. De Palma, G. Pascazio, and M. Napolitano. A second-order-accurate explicit fluctuation splitting scheme for unsteady problems. Computers and Fluids, 38(7):1384–1393, 2008.

[103] J.A. Rossmanith. High-order residual distribution schemes for steady 1d relativistic hydrodynamics. Hyperbolic Problems: Theory, Numerics, and Applications II, pages 259–266, 2006.

[104] James A. Rossmanith. A class of residual distribution schemes and their relation to relaxation systems. J. Comput. Phys., 227(22):9527–9553, 2008.

[105] P. Cinella R.W. Walters and D.C. Slack. Characteristic-based algorithms for flows in thermochemical nonequilibrium. AIAA Journal, 30(5):1304–1313, 1992.

[106] D. Serre S. Benzoni-Gavage. Multi-dimensional Hyperbolic Partial Differential Equations: First-order Systems and Applications. Oxford Science Publications, Univ. Lyon I, France, 2007 A Verifier.

[107] H. Schlichting. Boundary Layer Theory. McGraw-Hill, New York, 1979.

[108] K. Sermeus and H. Deconinck. An entropy fix for multidimensional upwind residual distribution schemes. Computers and Fluids, 34(4):617–640, 2005.

[109] D. Serre. Systèmes de lois de conservation. Tome 1, Hyperbolicité, entropies, ondes de choc. Diderot Editeur Arts Sciences, École Normale Supérieure de Lyon, France, 1996.

[110] C.-W. Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In A. Quarteroni, editor, Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, volume 1697 of Lecture Notes in Mathematics, pages 325–432. Springer-Verlag, Heidelberg, 1998.

[111] C.-W. Shu. High-order methods for computational physics. In T.J. Barth and H. Deconinck, editors, High-Order ENO and WENO schemes for computational fluid dynamics, volume 9 of Lecture Notes in Computational Science and Engineering, pages 439–582. Springer-Verlag, Heidelberg, 1999.

[112] E.H. Spanier. Algebraic topology. McGraw-Hill, 1966.

[113] R. Struijs. A multi-dimensional upwind discretization method for the Euler equations on unstructured grids. PhD thesis, TU Delft, 1994.

[114] C. Tavé. Construction simple de schémas distribuant le résidu non-oscillants et d'ordre élevé pour la simulation d'écoulements stationnaires sur maillages triangulaires et hybrides. PhD thesis, INRIA and University of Bordeaux I, 2007.

[115] R. Abgrall A. Larat M. Ricchiuto C. Tavé. A simple construction of very high order non oscillatory compact schemes on unstructured meshes. Computers and Fluids, 38(7):1314–1323, 2008.

[116] R. Abgrall M. Ricchiuto N. Villedieu C. Tavé and H. Deconinck. Very high order residual distribution on triangular grids. ECCOMAS CFD 2006, European Conference on Computational Fluid Dynamics, September 2006.

[117] E. F. Toro. Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction. Springer, 1999. Second Edition.

[118] R.S. Varga. Matrix Iterative Analysis. Prentice Publishing, 1962.

[119] J.D. Van Der Waals. On the Continuity of the Gaseous and Liquid States. PhD thesis, Universiteit Leiden, The Netherlands, 1873.

[120] R.F. Probstein W.D. Hayes. Hypersonic Inviscid Flow. Lavoisier, 2004. 2nd edition.

Appendix A

3D Diffusive Matrix

$\mathbb{K}$ is a $d \times d$ diffusive matrix of $m \times m$ ($m = d + 2$) matrices $K_{ij}, \quad i, j = 1, \ldots, d$, such that the 3D Navier-Stokes equations write:

$$\frac{\partial \mathbf{U}}{\partial t} + \operatorname{div}\left(\overrightarrow{\boldsymbol{\mathcal{F}}}(\mathbf{U})\right) = \operatorname{div}\left(\mathbb{K}.\overrightarrow{\nabla \mathbf{U}}\right)$$

$$K_{11} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -\frac{4u}{3} & \frac{4}{3} & 0 & 0 & 0 \\ -v & 0 & 1 & 0 & 0 \\ -w & 0 & 0 & 1 & 0 \\ -\left((1-\frac{\gamma}{\mathrm{Pr}})|\vec{\mathbf{u}}|^2 + \frac{u^2}{3} + \frac{\gamma E}{\mathrm{Pr}}\right) & u\left(\frac{4}{3} - \frac{\gamma}{\mathrm{Pr}}\right) & v\left(1 - \frac{\gamma}{\mathrm{Pr}}\right) & w\left(1 - \frac{\gamma}{\mathrm{Pr}}\right) & \frac{\gamma}{\mathrm{Pr}} \end{pmatrix},$$

$$K_{12} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{2v}{3} & 0 & -\frac{2}{3} & 0 & 0 \\ -u & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\frac{uv}{3} & v & -\frac{2u}{3} & 0 & 0 \end{pmatrix}, \; K_{13} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{2w}{3} & 0 & 0 & -\frac{2}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -u & 1 & 0 & 0 & 0 \\ -\frac{uw}{3} & w & 0 & -\frac{2u}{3} & 0 \end{pmatrix},$$

and

$$K_{22} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -u & 1 & 0 & 0 & 0 \\ -\frac{4v}{3} & 0 & \frac{4}{3} & 0 & 0 \\ -w & 0 & 0 & 1 & 0 \\ -\left((1-\frac{\gamma}{\mathrm{Pr}})|\vec{\mathbf{u}}|^2 + \frac{v^2}{3} + \frac{\gamma E}{\mathrm{Pr}}\right) & u\left(1 - \frac{\gamma}{\mathrm{Pr}}\right) & v\left(\frac{4}{3} - \frac{\gamma}{\mathrm{Pr}}\right) & w\left(1 - \frac{\gamma}{\mathrm{Pr}}\right) & \frac{\gamma}{\mathrm{Pr}} \end{pmatrix},$$

$$K_{21} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -v & 0 & 1 & 0 & 0 \\ \frac{2u}{3} & -\frac{2}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\frac{uv}{3} & -\frac{2v}{3} & u & 0 & 0 \end{pmatrix}, \; K_{23} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{2w}{3} & 0 & 0 & -\frac{2}{3} & 0 \\ -v & 0 & 1 & 0 & 0 \\ -\frac{vw}{3} & 0 & w & -\frac{2v}{3} & 0 \end{pmatrix},$$

and

$$K_{33} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -u & 1 & 0 & 0 & 0 \\ -v & 0 & 1 & 0 & 0 \\ -\frac{4w}{3} & 0 & 0 & \frac{4}{3} & 0 \\ -\left((1-\frac{\gamma}{\mathrm{Pr}})|\vec{\mathbf{u}}|^2 + \frac{w^2}{3} + \frac{\gamma E}{\mathrm{Pr}}\right) & u\left(1-\frac{\gamma}{\mathrm{Pr}}\right) & v\left(1-\frac{\gamma}{\mathrm{Pr}}\right) & w\left(\frac{4}{3}-\frac{\gamma}{\mathrm{Pr}}\right) & \frac{\gamma}{\mathrm{Pr}} \end{pmatrix},$$

$$K_{31} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -w & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{2u}{3} & -\frac{2}{3} & 0 & 0 & 0 \\ -\frac{uw}{3} & -\frac{2w}{3} & 0 & u & 0 \end{pmatrix},\ K_{32} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -w & 0 & 0 & 1 & 0 \\ \frac{2v}{3} & 0 & -\frac{2}{3} & 0 & 0 \\ -\frac{vw}{3} & 0 & -\frac{2w}{3} & v & 0 \end{pmatrix}.$$

Appendix B

3D Jacobians

We detail here the three dimensional Jacobians of the advective flux $\mathrm{A} = \dfrac{\partial \mathbf{F}_1}{\partial \mathbf{U}}$, $\mathrm{B} = \dfrac{\partial \mathbf{F}_2}{\partial \mathbf{U}}$ and $\mathrm{C} = \dfrac{\partial \mathbf{F}_3}{\partial \mathbf{U}}$.

$$\mathrm{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ (\gamma-1)\mathscr{E}_c - u^2 & (3-\gamma)u & (1-\gamma)v & (1-\gamma)w & (\gamma-1) \\ -uv & v & u & 0 & 0 \\ -uw & w & 0 & u & 0 \\ u\left((\gamma-1)\mathscr{E}_c - \mathcal{H}\right) & \mathcal{H} + (1-\gamma)u^2 & (1-\gamma)uv & (1-\gamma)uw & \gamma u \end{pmatrix}$$

$$\mathrm{B} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ -uv & v & u & 0 & 0 \\ (\gamma-1)\mathscr{E}_c - v^2 & (1-\gamma)u & (3-\gamma)v & (1-\gamma)w & (\gamma-1) \\ -vw & 0 & w & v & 0 \\ v\left((\gamma-1)\mathscr{E}_c - \mathcal{H}\right) & (1-\gamma)uv & \mathcal{H} + (1-\gamma)v^2 & (1-\gamma)vw & \gamma v \end{pmatrix}$$

$$\mathrm{C} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ -uw & w & 0 & u & 0 \\ -vw & 0 & w & v & 0 \\ (\gamma-1)\mathscr{E}_c - w^2 & (1-\gamma)u & (1-\gamma)v & (3-\gamma)w & (\gamma-1) \\ w\left((\gamma-1)\mathscr{E}_c - \mathcal{H}\right) & (1-\gamma)uw & (1-\gamma)vw & \mathcal{H} + (1-\gamma)w^2 & \gamma w \end{pmatrix}$$

If $\vec{\boldsymbol{\lambda}} = (\mathrm{A}, \mathrm{B}, \mathrm{C})$ is the advection speed, then for any $\vec{\mathbf{n}} = (n_x, n_y, n_z) \in \mathrm{S}^2$, $\vec{\boldsymbol{\lambda}}.\vec{\mathbf{n}}$ is diagonalizable and one has $\vec{\boldsymbol{\lambda}}.\vec{\mathbf{n}} = \mathcal{R}\boldsymbol{\Lambda}\mathcal{L}$ with:

$$\boldsymbol{\Lambda} = \mathrm{diag}\left(\vec{\mathbf{u}}.\vec{\mathbf{n}} - c, \vec{\mathbf{u}}.\vec{\mathbf{n}}, \vec{\mathbf{u}}.\vec{\mathbf{n}}, \vec{\mathbf{u}}.\vec{\mathbf{n}}, \vec{\mathbf{u}}.\vec{\mathbf{n}} + c\right),$$

$$\mathcal{R} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ u - cn_x & u & n_y & n_z & u + cn_x \\ v - cn_y & v & -n_x & 0 & v + cn_y \\ w - cn_z & w & 0 & -n_x & w + cn_z \\ \mathcal{H} - \vec{\mathbf{u}}.\vec{\mathbf{n}}c & \mathscr{E}_c & un_y - vn_x & un_z - wn_x & \mathcal{H} + \vec{\mathbf{u}}.\vec{\mathbf{n}}c \end{pmatrix},$$

$$\mathcal{L} = \begin{pmatrix} \frac{1}{2c}\left(\frac{\gamma-1}{c}\mathscr{E}_c + \vec{\mathbf{u}}.\vec{\mathbf{n}}\right) & -\frac{1}{2c}\left(\frac{\gamma-1}{c}u + n_x\right) & -\frac{1}{2c}\left(\frac{\gamma-1}{c}v + n_y\right) & -\frac{1}{2c}\left(\frac{\gamma-1}{c}v + n_y\right) & \frac{\gamma-1}{2c^2} \\ 1 - \frac{(\gamma-1)\mathscr{E}_c}{c^2} & \frac{(\gamma-1)u}{c^2} & \frac{(\gamma-1)v}{c^2} & \frac{(\gamma-1)w}{c^2} & \frac{(1-\gamma)}{c^2} \\ vn_x - un_y & n_y & -n_x & 0 & 0 \\ wn_x - un_z & n_z & 0 & -n_x & 0 \\ \frac{1}{2c}\left(\frac{\gamma-1}{c}\mathscr{E}_c - \vec{\mathbf{u}}.\vec{\mathbf{n}}\right) & -\frac{1}{2c}\left(\frac{\gamma-1}{c}u - n_x\right) & -\frac{1}{2c}\left(\frac{\gamma-1}{c}v - n_y\right) & -\frac{1}{2c}\left(\frac{\gamma-1}{c}w - n_z\right) & \frac{\gamma-1}{2c^2} \end{pmatrix}.$$