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
Luc MIEUSSENS	Professeur, IMB	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

to e mn n u il p t pivo p k plno d l ch v ci
venuji tuto t zu m mu d dovi ouckovi

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 (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 (CLs). 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 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 CL considérée, qu'ils sont stables en norme L^8 et qu'ils ont la bonne erreur de troncature. Même si tous ces développements théoriques ne sont démontrés que dans le cas de CLs 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 RD d'ordre très élevé et soulève de nouvelles questions pour des améliorations futures. Ces améliorations devrait faire des schémas RD une alternative attractive aux discrétisations classiques 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 (RDS) with the most possible compact stencil.

The manuscript starts with a review of the mathematical theory of hyperbolic Conservation Laws (CLs). 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 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 CL, that they are stable in L^8 norm, and that they have the proper truncation error. Even though all the theoretical developments are carried out for scalar CLs, 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 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 RDS, and contributes substantially to bring forward a number of open issues for future improvement. These improvements should make RD discretizations a very appealing alternative to now classical high order and very high order 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;

E ciency: 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 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 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 (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 DG schemes relies either on the use of FV limiters, which can reduce dramatically their accuracy, or, as stabilized 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^8 stability of the solution, on the other hand does not fully guarantee its local monotonicity. More importantly, these shock-capturing (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 DG framework. It is clear the 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 DG scheme and by a RD one. This is done in Tabular 1.1. The Residual Distribution framework presents always much less unknowns than its 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 RD scheme needs approximately $k^2 n$ degrees of freedom when DG needs $\frac{1}{2}(k+1)(k+2)n$. The same in 3D, both scheme needing asymptotically $k^3 n$ degrees of freedom.

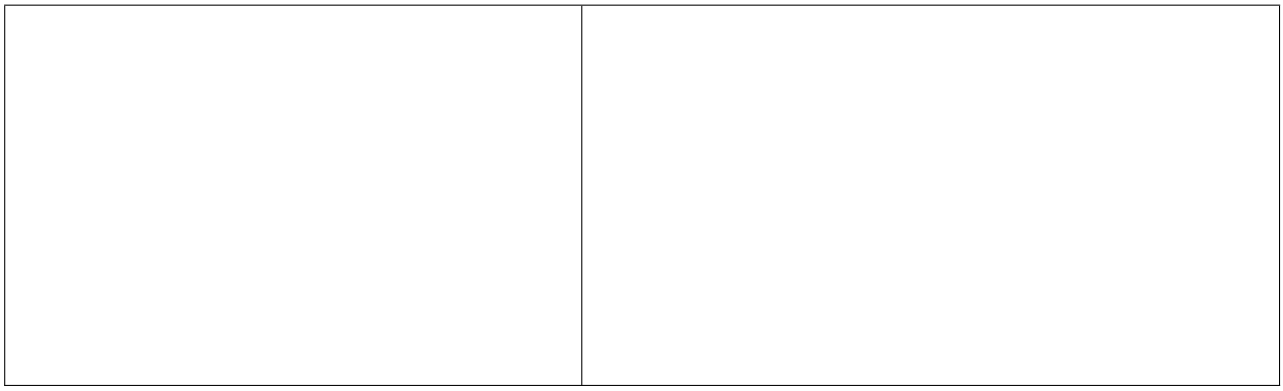


Figure 1.1: Third Order RD and DG meshes.

	2D		3D	
Order	DG	RD	DG	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 DG or a 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 (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.

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 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 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 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: RD schemes are not very well suited at that moment to deal with viscous problems. The main reason is that 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 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 RD schemes and to detail as much as possible the practical implementation of the 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 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 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 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 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 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 R^2, or a part of 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 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 (RDS). Indeed, 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 R^m, and U a vector of m variables $u_1;::::;u_m$. U is assumed to be a function from R^2 $r0;\ 8r$ into D. We call *system of* m *equations of conservation laws*, the system

$$\frac{BU}{Bt} \quad \frac{BF\ pUq}{Bx} \quad \frac{BG\ pUq}{By} \quad 0; \quad X \quad p\,x;y\,q\ PR^2; \quad t\ ¥\ 0 \tag{2.1}$$

where F and G are called the *flux-functions*. They are smooth functions from D into R^m. We also introduce the *flux-vector* $\tilde{F} \quad p\,F;G\,q$, which enables us to rewrite equation (2.1) into an equivalent form

$$\frac{BU}{Bt} \quad \tilde{r}:\tilde{F}\ pUq \quad 0; \quad X \quad p\,x;y\,q\ PR^2; \quad t\ ¥\ 0: \tag{2.1}$$

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

$$\frac{BU}{Bt} \quad \tilde{}:\tilde{r}U \quad 0; \quad X \quad p\,x;y\,q\ PR^2; \quad t\ ¥\ 0 \tag{2.2}$$

with $\tilde{} \quad \frac{BF}{BU};\frac{BG}{BU}$, the *flux Jacobians*.

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

$$\frac{d}{dt} \int_{\Omega} U\,dX \quad \int_{B\Omega} \tilde{F}\ pUq:\vec{n}\,ds \quad 0: \tag{2.3}$$

That means the time variation of $\int_{\Omega} U\,dX$ is equal to the mean flux $\tilde{F}\ pUq$ entering Ω. And because the flux entering Ω is the flux going out of $R^2z\Omega$, the quantities $u_1;::::;u_m$ 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.

$$\begin{cases} \frac{BU}{Bt} \quad A:\frac{BU}{Bx} \quad 0; \quad x\ PR; \ t\ ¡\ 0 \\ U\,px;0q \quad U_l; \quad x \quad 0 \\ U\,px;0q \quad U_r; \quad x\ ¡\ 0 \end{cases} \tag{2.4}$$

If we consider A as diagonalizable, there exists L and R, matrices of left and right eigenvectors respectively, such that $A \quad R \quad L$, with $\quad diagp\ _1;::::;\ _m q$. There is no restriction considering

Figure 2.1: Solution of the 1D linear Riemann problem for a4 dimensional unknown. The solution is represented in the eigenspace.

$_1; : : : ; _m$ sorted by increasing order. It is now straightforward that $V = p v_1; : : : ; v_m q$ $L U$ veri es the decoupled system:

$$\begin{cases} \dfrac{BV}{Bt} + \dfrac{BV}{Bx} = 0, & x \in R, t > 0 \\ V(x, 0) = L U_l = V_l, & x < 0 \\ V(x, 0) = L U_r = V_r, & x > 0 \end{cases} \tag{2.5}$$

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

$$v_i(x, t) = v_i(x - _i t, 0) = v_i(x, 0), \quad (x, t) \in R \times R^+, \quad i = 1 : : : m.$$

$U = \sum_i v_i r_i$ 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, $_1; : : : ; _m$. 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 $\dfrac{BF}{BU}$ and $\dfrac{BG}{BU}$ 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].

De nition 2.2 (Symmetrizability)

Operator D is symmetrizable if there exists a symmetric positive-de nite 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 $A(x,\xi) = P(x,\xi)^{-1} D(x,\xi) P(x,\xi)$ with $D(x,\xi)$ a real diagonal matrix, we have

$$\|\exp(i A(x,\xi)t)\| \leq \|P(x,\xi)\| \, \|P(x,\xi)^{-1}\| \, \|\exp(i D(x,\xi)t)\|$$

And condition (2.7) is ful lled when the conditioning $\|P(x,\xi)\| \, \|P(x,\xi)^{-1}\|$ of P is bounded independently of .

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

$$A(x,\xi) = R(x)^{-1} S_0 A(x,\xi) R(x) R(x)^{-1}$$

The matrix $RS_0A(x,\xi)R$ is symmetric and diagonalizable in an orthogonal basis and may be written as $Q(x,\xi)^T D(x,\xi) Q(x,\xi)$. We now have :

$$\|P(x,\xi)\| \, \|P(x,\xi)^{-1}\| \leq \|Q(x,\xi) R(x)^{-1}\| \, \|R Q(x,\xi)^T\| \leq \|R(x)^{-1}\| \, \|R\|,$$

a number independent of .

In the case of a constantly hyperbolic operator, the eigenspaces depend continuously on . Then for any $\xi_0 \in S^{d-1}$, there exists a neighborhood of ξ_0 on which a choice of $P(x,\xi)$ depends continuously on , and is thus bounded. And as the sphere S^{d-1} is compact, it is covered by a nite number of such neighborhoods. There now exists $C \in \mathbb{R}$ such that $\forall \xi \in S^{d-1}, \ \frac{1}{C} \leq \|P(x,\xi)\| \leq C$. 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 A. This result can be extended to any symmetrizable hyperbolic systems, as shown in [106].

Consider again equation (2.6) and use the notation, $\forall \xi \in S^1, \ A(\xi) = A_1 B_2$. If our system is symmetrizable, there exists a *s.p.d* constant matrix S_0 such that S_0A and S_0B are symmetric matrices. The system

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

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

$$Char = \{(x,\xi) \in S^1 \times \mathbb{R}; \ \det(S_0 A(x,\xi) - \lambda I_m) = 0\},$$

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

$$K(X,T) = \{(x,t) \in \mathbb{R}^2 \times \mathbb{R}^+; \ \forall (\xi,\tau) \in S^1, \ t - T \geq \xi(x - X) \Rightarrow \tau \leq 0\}.$$

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

Theorem 2.4 (Finite Propagation Speed)

If $V_{|t=0} = 0$ then $V(x,t) = 0$, $\forall (x,t) \in K(X,T)$

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

$$\frac{\partial}{\partial t}\|V\|^2_{2,m} + \frac{\partial}{\partial x}\langle A^1 V, V\rangle_m + \frac{\partial}{\partial y}\langle B^1 V, V\rangle_m \leq 0 \tag{2.9}$$

where the notation $\langle\cdot,\cdot\rangle_m$ is used for the canonical scalar product in $\mathbb{R}^m$ and matrices A^1 and B^1 are $A^1 = S_0^{1/2} A S_0^{1/2}$, $B^1 = S_0^{1/2} B S_0^{1/2}$.

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

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

and integrate relation (2.9) over $K(0,\varepsilon)$ (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 $(\nu_x, \nu_y) \in$ Char, being maximal in the direction ν. Thus we have:

$$\int_{\mathcal{K}(0,\varepsilon)} \tilde{r} \cdot \begin{pmatrix} \langle I_m V, V\rangle_m \\ \langle A^1 V, V\rangle_m \\ \langle B^1 V, V\rangle_m \end{pmatrix} dx\,dt \qquad \int_{\omega(t-\varepsilon)} \|V\|^2_{2,m}\,dx \leq \int_{\omega(0)} \|V\|^2_{2,m}\,dx$$

$$\int_0^{\cdot} \int_{\text{side}} \langle (A^1 \nu_1 + B^1 \nu_2 - I_m) V, V\rangle\,dx\,dt$$

But as for all ν, (ν_x, ν_y) is maximal in the direction ν, matrix $A^1 \nu_1 + B^1 \nu_2 - I_m$ 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 V is identically null on the bottom it is straightforward that it is null everywhere in $K(0,\varepsilon)$, ε being arbitrarily small, $V_{|K(X,T)} = 0$. ■

This result shows that in the case of constant coefficients matrices, for any (X,T) in the space-time domain, we can define a dependence cone, function of the eigenvalues of $A(\nu)$ in all space direction ν. We then know that the value of the solution at point (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 $A(\nu)$, 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 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 C^∞ - initial condition, there might exist a time T such that $\forall t \geq T$, the solution 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 (X, T). The propagation is anisotropic. $K(0; \nu)$ is the part of the cone between the two surfaces $\lambda(0)$ and $\lambda(t)$. $\nu = (\nu_t, \nu_x)$ is a normal to the side surface. It is an element of Char, with 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

$$u_0(x) = \begin{cases} 1, & x \leq 0; \\ 1 - x, & 0 \leq x \leq 1; \\ 0, & x \geq 1: \end{cases} \tag{2.11}$$

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 U_0 be a vector of m bounded function in $\mathbb{R}^2$. A function $U \in L^\infty(\mathbb{R}^2 \times \mathbb{R}_+; \mathbb{R}^m)$ is called *a weak solution* of problem (2.1) with initial condition U_0, if $U(x,t) \in D$ a.e. and satisfies for any C^1 function φ with compact support in $\mathbb{R}^2 \times [0, \infty)$

$$\int_0^\infty \int_{\mathbb{R}^2} U \cdot \frac{\partial \varphi}{\partial t} + F(U) \cdot \frac{\partial \varphi}{\partial x}\, dx\, dt + \int_{\mathbb{R}^2} U_0(x) \cdot \varphi(x, 0)\, dx = 0. \tag{2.12}$$

Remark 2.6

If U is a 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 $p1; 1q$ and the solution cannot be continuous there.

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

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

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 $u_{0\varepsilon} \in L^2$, 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) $\exists C \in \mathbb{R}$; $\|u_\varepsilon\|_\infty \leq C$; independently of ε;

b) $u_\varepsilon \xrightarrow[\varepsilon \to 0]{} u$ almost everywhere in $\mathbb{R}^2$ $r > 0$; $\forall r$;

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:

$$\frac{\partial}{\partial t} S(u) + \operatorname{div}(G(u)) \leq 0; \tag{2.16}$$

where

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

(ii) G is a vector of 2 scalar smooth functions such that

$$S'(u) f_j'(u) = G_j'(u), \quad j = 1; 2: \tag{2.17}$$

$(S; G)$ is called *a pair of Entropy-Flux*, S an *entropy function* and 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 $S'(U)$, quick calculation shows that U satisfies an additional conservation relation

$$\frac{\partial}{\partial t} S(U) + \sum_{r=1}^{2} \frac{\partial}{\partial x_r} G(U) \leq 0; \quad X = (x; y) \in \mathbb{R}^2; \quad t \geq 0: \tag{2.18}$$

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 $u_0 \in L^\infty(\Omega)$, verifying relation (2.16) for any pair of Entropy-Flux (S, G) is unique and called the entropy solution . Moreover this solution is bounded

$$\partial T > 0, \quad u \in L^\infty(\Omega \times [0, T])$$

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:

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

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 nite di erence scheme applied to equation (2.14) with initial boundary conditions u_l 1 and u_r 1.

Theorem 2.9 (Mock)

Let S : D ÝÑ R be a smooth convex function. A necessary and su cient condition forS to be an entropy for system (2.1) is that them m matrices $S^2pU qF^1pU q$ and $S^2pU qG^1pU q$ are symmetric.

Proof : Let rst assume S is a convex entropy for system (2.1). Then there exists a vector of smooth functions G, such that $S^1pU qF^1pU q$ $G_1^1pU q$ and $S^1pU qG^1pU q$ $G_2^1pU q$. Let consider only the rst relation and di erentiate its k^{th} -line with respect to u_j . We obtain :

$$\frac{B}{Bu_j}\sum_{i\ 1}^{m}\frac{BF_i}{Bu_k}\frac{BS}{Bu_i}\quad\frac{BG_1}{Bu_k}\qquad 0 \tag{2.20}$$

$$\hat{o}\quad\frac{B^2G_1}{Bu_k u_j}\quad\sum_{i\ 1}^{m}\frac{B^2F_i}{Bu_k u_j}\frac{BS}{Bu_i}\quad\sum_{i\ 1}^{m}\frac{BF_i}{Bu_k}\frac{B^2S}{Bu_i u_j}: \tag{2.21}$$

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

$$\sum_{i\ 1}^{m}\frac{BF_i}{Bu_k}\frac{B^2S}{Bu_i u_j}\quad\sum_{i\ 1}^{m}\frac{BF_i}{Bu_j}\frac{B^2S}{Bu_i u_k}: \tag{2.22}$$

This means exactly the matrix $S^2pU qF^1pU q$ is symmetric. And same argument holds for the second coordinate$G pU q$.

Conversely, assuming (2.22), we have

$$\frac{B}{Bu_j}\sum_{i\ 1}^{m}\frac{BF_i}{Bu_k}\frac{BS}{Bu_i}\qquad\sum_{i\ 1}^{m}\frac{BF_i}{Bu_k}\frac{B^2S}{Bu_i Bu_j}\quad\frac{B^2F_i}{Bu_k Bu_j}\frac{BS}{Bu_i} \tag{2.23}$$

$$\frac{B}{Bu_k}\sum_{i\ 1}^{m}\frac{BF_i}{Bu_j}\frac{BS}{Bu_i}\quad: \tag{2.24}$$

If our spatial domain is contractible (there is a homotopy that continuously deforms to a point), it follows from Poincaré's lemma that there exists a function G_1, such that

$$\frac{BG_1}{Bu_k}\quad\sum_{i\ 1}^{m}\frac{BF_i}{Bu_k}\frac{BS}{Bu_i};\quad @k\ P\ v1;mw$$

And because once more the same arguments hold forG_2, S is an entropy function associated with the entropy uxes G_1 and G_2. ■

Figure 2.5: E ect of the boundary and initial conditions on the i^{th} component of the unknown in both cases when $_i \quad 0$ and $_i ¡ 0$

We now come to a more complex problem, with space dimension m and non constant coef- cient 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 R^n. It is then well de ned by its unit normal . We moreover suppose that points inside the domain. If we further assume that our problem is symmetrizable, the Jacobian of the ux is diagonalizable in the direction of and we once more call $_1; : : : ; _m$ its eigenvalues in the direction , sorted by increasing order and $r_1; : : : ; r_m$ the associated eigenvec- tors. If p is the integer index such that $_p \quad 0 ¤ _{p\ 1}$, $r_1; : : : ; r_p$ are the direction of strictly outgoing information, $r_{p\ 1}; : : : ; r_m$ 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 rst start by the main mechanical conservation laws and apply some restrictions coming from uid mechanics. Some ner hypothesis on the uid 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.

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

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

$$\forall M \in \Omega,\ \forall \vec{n} \in R^2,\ \tilde{F}_s(M,\vec{n}) = \sigma(M)\vec{n}.$$

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 ($i = 1,2$)

$$\frac{\partial}{\partial t}(u_i) = \operatorname{div}(u_i\vec{u}) + (\tilde{f}_v) + \operatorname{div}(\sigma_i), \tag{2.28}$$

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

$$\frac{D}{Dt}\int_{\omega(t)} \overrightarrow{OM} \wedge \vec{u}\, dM = \int_{\omega(t)} \overrightarrow{OM} \wedge \tilde{f}_v\, dM + \int_{\partial\omega(t)} \overrightarrow{OM} \wedge \sigma(M)\vec{n}\, ds \tag{2.29}$$

In 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 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 $E = \frac{1}{2}\|\vec{u}\|^2 + e$ 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 $\vec{q}$ the heat flux inside Ω, this is translated for any time t as

$$\frac{D}{Dt}\int_{\omega(t)} E\, dx = \int_{\omega(t)} \tilde{f}_v \cdot \vec{u}\, dx + \int_{\partial\omega(t)} \tilde{F}_s(M,\vec{n}) \cdot \vec{u}(M)\, ds + \int_{\omega(t)} w\, dx - \int_{\partial\omega(t)} \vec{q}\cdot\vec{n}\, ds \tag{2.30}$$

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

$$\frac{\partial E}{\partial t} = \operatorname{div}(E\vec{u} - \sigma\cdot\vec{u} + \vec{q}) + \tilde{f}_v\cdot\vec{u} + w \tag{2.31}$$

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

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

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

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

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

$$\frac{D}{Dt}\int_{\omega(t)} s\,dx \geq \int_{\omega(t)} \frac{w}{T}\,dx - \int_{\partial\omega(t)} \frac{q\cdot n}{T}\,ds. \tag{2.33}$$

We then obtain the *local entropy inequality*:

$$\frac{\partial s}{\partial t} + \mathrm{div}(s\,u) + \frac{\mathrm{div}\,q}{T} \geq \frac{w}{T}. \tag{2.34}$$

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

$$w = \frac{De}{Dt} + \mathrm{div}(q) - \sigma : D,$$

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

$$T\frac{Ds}{Dt} - \frac{De}{Dt} + \frac{q\cdot\nabla T}{T} + \sigma : D \geq 0. \tag{2.35}$$

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:

$$\frac{De}{Dt} = \left(\frac{\partial e}{\partial s}\right)_v \frac{Ds}{Dt} + \left(\frac{\partial e}{\partial v}\right)_s \mathrm{div}(u)$$
$$\left(\frac{\partial e}{\partial s}\right)_v \frac{Ds}{Dt} + \left(\frac{\partial e}{\partial v}\right)_s \mathrm{Tr}(D),$$

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

$$T\left(\frac{\partial e}{\partial s}\right)_v \frac{Ds}{Dt} - p\left(\frac{\partial e}{\partial v}\right)_s \mathrm{Tr}(D) - \mathrm{div}(u)\lambda(\mathrm{div}(u))^2 - 2\mu D : D + \frac{q\cdot\nabla T}{T} \geq 0. \tag{2.36}$$

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

$$T \quad \frac{De}{Ds} \qquad \text{and} \qquad \frac{q \cdot \nabla T}{T} \geq 0:$$

If you consider the heat transfers follow the Fourier law $q \quad k\nabla T$, 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 $T \quad \frac{Be}{Bs}$,

Clausius-Duhem Inequality says

$$p \quad \frac{Be}{Bv}_s \quad Tr\ pDq \quad p\mathrm{div}pvq \quad 2\ D : D \quad \frac{q \cdot \nabla T}{T} \geq 0;$$

which is always satisfied if and only if:

$$p \quad \frac{Be}{Bv}_s \qquad \text{and} \qquad p\mathrm{div}pvq \quad 2\ D : D \quad \frac{q \cdot \nabla T}{T} \geq 0$$

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

$$3 \quad 2 \geq 0: \tag{2.37}$$

Eventually, we can physically define entropy functions $p \quad Sq$ 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

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

where N is the number of mole of gas contained in the volume v and $R \quad 8{:}3144\,\mathrm{J.K}^{-1}\mathrm{:mol}^{-1}$ is a universal constant.

Polytropic Gas: a *polytropic gas* is merely an ideal gas for which the heat capacity at constant volume is constant. $c_v = \left(\frac{\partial e}{\partial T}\right)_v = e = c_v T$. Then relation (2.38) is reformulated into

$$p = \rho(\gamma - 1)e \tag{2.39}$$

where γ is the ratio of the heat capacities $\left(\frac{c_p}{c_v} = \gamma = 1.4 \text{ for the air}\right)$.

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 : $\sigma' = 0$,
2. There is no body forces : $\vec{f}_v = \vec{0}$,
3. There is no heat transfer : $w = 0$; $q = 0$.

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

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \operatorname{div}(\rho \vec{u}) = 0 \\ \dfrac{\partial \rho u_i}{\partial t} + \operatorname{div}(\rho u_i \vec{u} + p\,\delta_i) = 0, \quad i = 1,2 \\ \dfrac{\partial \rho E}{\partial t} + \operatorname{div}(\rho E \vec{u} + p\vec{u}) = 0 \end{cases} \tag{2.40}$$

where δ_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

$$U = \begin{pmatrix} \rho \\ \rho \vec{u} \\ \rho E \end{pmatrix}, \quad \text{and} \quad \vec{F} = (F_1, F_2), \quad \text{with} \quad F_i = \begin{pmatrix} \rho u_i \\ \rho u_i \vec{u} + p\,\delta_i \\ \rho E u_i + p u_i \end{pmatrix} \tag{2.41}$$

system (2.40) is rewritten in the compact form

$$\frac{\partial U}{\partial t} + \operatorname{div}(\vec{F}(U)) = 0$$

and if we denote by A $\frac{\partial F_1}{\partial U}$, B $\frac{\partial F_2}{\partial U}$ and $\tilde{}$ p A; Bq the Jacobian of the fluxes, we obtain for a smooth enough solution the equivalent quasi-linear form

$$\frac{\partial U}{\partial t} \quad \tilde{} : r \acute{\text{Y}} \acute{\text{K}} U \quad 0:$$

2.2.9 Properties of the Euler Equations

The matrices A and B are the following

$$A \quad \begin{array}{ccccc} 0 & & 1 & 0 & 0 \\ p \quad 1 \text{o} E_c \quad u^2 & & p3 \quad qu & p1 \quad qv & p \quad 1q \\ uv & & v & u & 0 \\ u\,pp \quad 1 \text{o} E_c \quad Hq & H & p\,1 \quad qu^2 & p1 \quad quv & u \end{array}$$

$$B \quad \begin{array}{ccccc} 0 & & 0 & 1 & 0 \\ uv & & v & u & 0 \\ p \quad 1 \text{o} E_c \quad v^2 & & p1 \quad qu & p3 \quad qv & p \quad 1q \\ v\,pp \quad 1 \text{o} E_c \quad Hq & & p1 \quad quv & H \quad p\,1 \quad qv^2 & v \end{array}$$

where E_c p u^2 v^2q{2 and H e p{ denote the *kinetic energy* and the *enthalpy* per unit volume, respectively. Given a unit normal ƞ p n_x; n_yq PS^1, the matrix

$$\tilde{} : ƞ \quad \begin{array}{ccccc} 0 & & n_x & n_y & 0 \\ p \quad 1 \text{o} E_c n_x \quad u \text{ʊ} : ƞ & & \text{ʊ} : ƞ \quad p\,2 \quad qun_x & un_y \quad p\,1 \quad qvn_x & p \quad 1 \text{o} n_x \\ p \quad 1 \text{o} E_c n_y \quad v \text{ʊ} : ƞ & & vn_x \quad p\,1 \quad qun_y & \text{ʊ} : ƞ \quad p\,2 \quad qvn_y & p \quad 1 \text{o} n_y \\ \text{ʊ} : ƞ\,pp \quad 1 \text{o} E_c \quad Hq & & Hn_x \quad p\,1 \quad qu \text{ʊ} : ƞ & Hn_y \quad p\,1 \quad qv \text{ʊ} : ƞ & \text{ʊ} : ƞ \end{array}$$

is diagonalizable and one has $\tilde{} : ƞ$ R L with:

$$\text{ʊ} : ƞ \quad c \quad 0 \quad 0 \quad 0$$
$$0 \quad \text{ʊ} : ƞ \quad 0 \quad 0$$
$$0 \quad 0 \quad \text{ʊ} : ƞ \quad 0 \quad ;$$
$$0 \quad 0 \quad 0 \quad \text{ʊ} : ƞ \quad c$$

$$R \quad \begin{array}{cccc} 1 & 1 & 0 & 1 \\ u \quad cn_x & u & n_y & u \quad cn_x \\ v \quad cn_y & v & n_x & v \quad cn_y \\ H \quad \text{ʊ} : ƞc & E_c \quad \text{ʊ} : ƞ^K & H & \text{ʊ} : ƞc \end{array} ;$$

$$L \quad \begin{array}{cccc} \frac{1}{2c} \quad \frac{1}{c} E_c \quad \text{ʊ} : ƞ & \frac{1}{2c} \quad \frac{1}{c} u \quad n_x & \frac{1}{2c} \quad \frac{1}{c} v \quad n_y & \frac{1}{2c^2} \\ 1 \quad \frac{p \quad 1 \text{o} E_c}{c^2} & \frac{p \quad 1 \text{o} qu}{c^2} & \frac{p \quad 1 \text{o} qv}{c^2} & \frac{p1 \quad q}{c^2} \\ \text{ʊ} : ƞ^K & n_y & n_x & 0 \\ \frac{1}{2c} \quad \frac{1}{c} E_c \quad \text{ʊ} : ƞ & \frac{1}{2c} \quad \frac{1}{c} u \quad n_x & \frac{1}{2c} \quad \frac{1}{c} v \quad n_y & \frac{1}{2c^2} \end{array} :$$

We have introduced a new variable c b $\frac{}{p}$ which represents the speed of propagation of the acoustic phenomena. It is well known that for air c 330m.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*:

$$\sigma = p\,\frac{\partial u}{\partial x}\,\frac{\partial v}{\partial y}\,I + \frac{\partial u}{\partial y}\,\frac{\partial v}{\partial x};$$

   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*: $q = k\nabla T$,

3. There is no body forces : $f_v = 0$,

4. There is no heat production inside the domain : $w = 0$,

5. The fluid is a polytropic gas : $p = \rho\varepsilon$. 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 $\lambda + \frac{2}{3}\mu \geq 0$ 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

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \operatorname{div}(\rho u) = 0 \\ \dfrac{\partial \rho u_i}{\partial t} + \operatorname{div}(\rho u_i u) + \dfrac{\partial p}{\partial x_i} = \operatorname{div}\tau + \rho f_i, \quad i = 1,2 \\ \dfrac{\partial \rho E}{\partial t} + \operatorname{div}(\rho E u) + \operatorname{div}(p u) = \operatorname{div}(k\nabla T) + \operatorname{div}(\tau u) + \rho f \cdot u + \rho w \end{cases} \tag{2.42}$$

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

$$\tau = 2\mu D(u) + \lambda \operatorname{div}(u) I = \begin{pmatrix} 2\mu\dfrac{\partial u}{\partial x} + \lambda\operatorname{div}(u) & \mu\left(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}\right) \\ \mu\left(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}\right) & 2\mu\dfrac{\partial v}{\partial y} + \lambda\operatorname{div}(u) \end{pmatrix}$$

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

$$K_{11} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{4}{3}u & \frac{4}{3} & 0 & 0 \\ v & 0 & 1 & 0 \\ 2E_c\left(\frac{u^2}{3} + \frac{1}{Pr}pe\right) - E_c q\left(u\frac{4}{3} + \frac{v}{Pr}\right) & -E_c q\frac{1}{Pr} & -E_c q\frac{1}{Pr} & 0 \end{pmatrix};$$

$$K_{12} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{2}{3}v & 0 & \frac{2}{3} & 0 \\ u & 1 & 0 & 0 \\ \frac{uv}{3} & v & \frac{2}{3}u & 0 \end{pmatrix}; \quad K_{21} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ v & 0 & 1 & 0 \\ \frac{2}{3}u & \frac{2}{3} & 0 & 0 \\ \frac{uv}{3} & \frac{2}{3}v & u & 0 \end{pmatrix};$$

and

$$K_{22} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ u & 1 & 0 & 0 \\ \frac{4}{3}v & 0 & \frac{4}{3} & 0 \\ 2E_c\left(\frac{v^2}{3} + \frac{1}{Pr}pe\right) - E_c q\left(u + \frac{v}{Pr}\right) & -E_c q\frac{u}{Pr} & -E_c q\left(v\frac{4}{3} + \frac{1}{Pr}\right) & -E_c q\frac{1}{Pr} \end{pmatrix};$$

we can rewrite system (2.42) as

$$U_{,t} + \operatorname{div}\left(\tilde{F}(pU)\right) = \operatorname{div}\left(pK_{ij}U_{,j}q_{,i}\right) + \operatorname{div}\left(K : \widetilde{\nabla U}\right) \tag{2.43}$$

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 U_∞ is the state we want to impose on boundary Γ_∞, one has:

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

In practice, if n is the inward normal to Γ_∞, some characteristics in the direction 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

$$U(x) - A(n)(U(x) - U_\infty), \quad \forall x \in \Gamma_\infty.$$

where $A(n)$ denotes the positive part of the Jacobian operator in the direction n.

No-Slip Wall : when the fluid is considered viscous, it sticks to the walls. By continuity, the velocity v of the flow along the wall must be the same as the velocity of the wall v_{wall}

$$v(x) = v_{wall}, \quad \forall x \in \Gamma_{wall}. \tag{2.45}$$

In most of cases, the wall is still and $v_{wall} = 0$. 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}

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

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:

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

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

$$v(x) \cdot n = 0, \quad \forall x \in \Gamma_{wall}. \tag{2.48}$$

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 E a functional space with infinite dimension and $\Diamond$ a differential operator on E. We also denote by L a Hilbert functional space such that :

i) $E \in L$;

ii) $\Diamond : E \tilde{N} L$.

3.1.1 Finite Dimension Approximation

We want to solve the following problem:

$$\text{Find } u \in E; \text{ such that } \begin{cases} \Diamond u = f; & x \in \Omega \\ u = g; & x \in \Gamma \subset B\Omega: \end{cases} \tag{3.1}$$

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 $\forall v \in L^2(\Omega)$, $\langle \mathcal{L}u, v \rangle_{L^2} = \langle f, v \rangle_{L^2}$. We now denote by W_h a subspace of E with finite dimension n, and by $w_1^h, \ldots, w_n^h$ a basis of W_h. The subscript "h" is used in order to keep in mind that W_h depends on the geometry of Ω, and on a spatial discretization of Ω, 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 P_h as a projection from E to W_h, for example

$$P_h : \begin{cases} E \to W_h \\ u \mapsto \sum_{i=1}^{n} \left\langle u, w_i^h \right\rangle_{L^2} w_i^h \end{cases}$$

We will see next this is not the only way of defining a projection from E to 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

$$\text{Find } u_h \in W_h, \text{ such that } \begin{cases} \langle \mathcal{L}u_h, v_h \rangle_{L^2} = \langle f, v_h \rangle_{L^2} \; ; & \forall v_h \in W_h \\ u_h = P_h(g), & \forall x \in \Gamma \end{cases} \tag{3.2}$$

If $\mathcal{L}$ is a linear operator, this problem can be obviously put into the matricial form $A.U = B$ where $A_{ij} = \left\langle \mathcal{L}w_i^h, w_j^h \right\rangle_{L^2}$ and $B_i = \langle f, w_i^h \rangle_{L^2} - F_i^{bc}$. F_i^{bc} stands here for the contribution of some numerical fluxes on the boundary Γ, this ensuring the boundary condition $u_h = P_h(g)$.

In this case, problem (3.2) is well-posed if matrix A is invertible and admits then a unique solution $u_h \in W_h$. 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 L^2 and we can then write the global error of approximation as

$$\|u - u_h\|_{L^2} \leq \underbrace{\|u - P_h(u)\|}_{I} + \underbrace{\|P_h(u) - u_h\|}_{II}$$

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 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 W_h. That means in particular that $P_h(u)$ converges toward u as h goes to 0 for any regular enough $u \in E$, and that in a certain sense, W_h converges toward 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 L^∞-norm of the exact solution u. Thus, below we will speak of a $(k+1)^{th}$-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 E by 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 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 $\Omega \in \mathbb{R}^2$ 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 $\bigcup_{T \in \mathcal{T}_h} T = \Omega$ or at least such that the area belonging to $\bigcup_{T \in \mathcal{T}_h} T$ 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

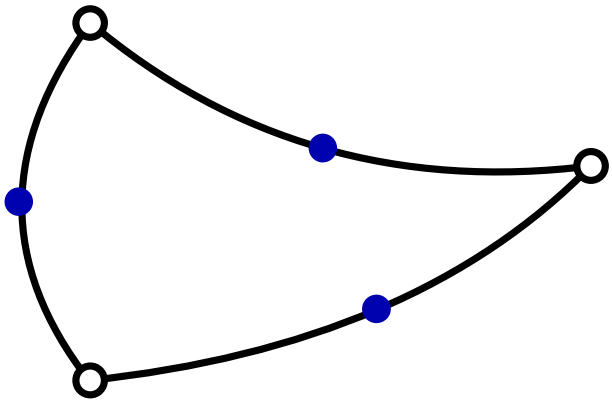


Figure 3.1: (Isoparametric Elements) The edges of these elements are represented by the same polynomial order k as the one used inside the element to approximate the solution. In that case, k 2 and the edges are quadratic, uniquely defined by the vertices and the middles of the edges. These elements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

C_2.

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

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 $|T|$ 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, D_i the subset of elements sharing i, its direct neighbours $j_1, \dots, j_6$ and the associated dual cell C_i. C_i is defined by joining the midpoints of the edges sharing i and the centroids of the triangles of D_i. This can be generalized to any polyhedral.

associate its dual cell, 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 D_i with the midpoints of the edges meeting at i. This notion is very important in the case of Finite Volume Schemes (FV), see Subsection 4.1.3 page 62. In the case of 2D schemes, we are mainly interested by the dual cell area

$$|C_i| \quad \frac{|D_i|}{3};$$

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 M_h be a unstructured hybrid meshing of a two dimensional simply connected domain and $F; E; V$ being respectively, the number of elements, edges and vertices of M_h. Then

$$F \quad E \quad V \quad 1 \tag{3.4}$$

Remark 3.2 (Euler Characteristic)

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

Figure 3.5: 7 points, 7 edges, 2 triangles and 2 connected components: 7 2 2.

any choosen order k. To do so, we have to add new degrees of freedom inside each element, in order to define what we are going to call 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: $\forall T \in \mathcal{D}_i,\ \varphi_i^T = 0$, we define the basis function associated to node i by

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

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 $\mathcal{B}^1 = \{\varphi_i^1; i \in \mathcal{M}_h\}$. 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 $\mathcal{B}^1$ is a basis of $W_h^1 = \text{Span}(\mathcal{B}^1)$, and 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 $P^1(\mathcal{M}_h)$ or simply P^1 when no confusion is possible. 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 W_h^1 taking value v_i at node i, in the basis $(\varphi_i^1)_{i\in[\![1;n]\!]}$.

Figure 3.8: Control cells C_i, C_j and C_k and sub-triangulation in P^2 formulation.

i 10

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

k=4: i 1::3

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

i 4::9, j is the vertex of T the nearest to i, k is the other tip of the edge

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

i 10::12, $j;k$ are the tips of the edge i is part of

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

i 13::15, j is the vertex of T the nearest to i

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

We still use the convention $\forall T \in D_i,\ \varphi_i^{T;k} = 0$ and thus define the k^{th}-order basis function associated to node i by :

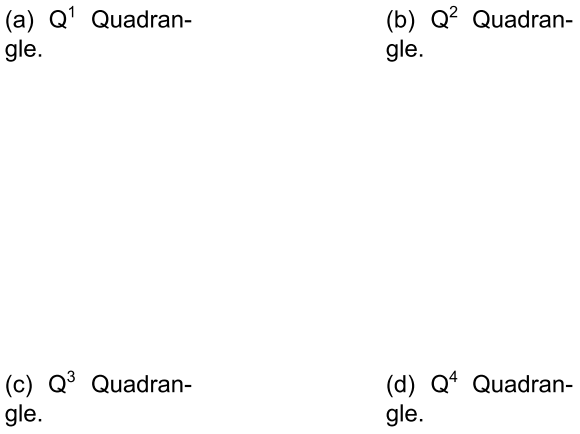
$$\varphi_i^k(x) = \varphi_i^{T;k}(x), \text{ when } x \in T. \tag{3.7}$$

Once more the finite subset $\left(\varphi_i^k\right)_{i\in[\![1;\#DoF]\!]}$ has linearly independant elements and is then a basis of 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 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 W_h^k, also denoted by $\Pi_h^k u$, by

$$\Pi_h^k u = u_h^k = \sum_{i\in M_h} u(x_i)\varphi_i^k. \tag{3.8}$$

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



(a) Q^1 Quadrangle.

(b) Q^2 Quadrangle.

(c) Q^3 Quadrangle.

(d) 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

$$\varphi_i^{k,\ell}(x,t) = \varphi_i^k(x)\,\psi_i^\ell(t) \tag{3.13}$$

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 $(k+1)^{th}$-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

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

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 $\varepsilon^{d/k} \xrightarrow[k\to\infty]{} 1$. 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 P^1 steady scalar non viscous case, we still aim at explaining the end of this manuscript the treatment of a 3D, 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

$$\begin{cases} \operatorname{div} \vec{F}(u) = 0 & (4.1) \\ \text{Boundary Conditions (Dirichlet, Neumann, strong or weak...)} \end{cases}$$

As we did before, the flux vector $\vec{F}$ 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 $u_{|B\Omega} = 0$ 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 (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 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 RDS, obviously, but also all the numerical formulations on conservative systems. In particular, it is very hard to show that a 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

$$\Phi^T = \int_T \operatorname{div} \vec{F}(u,q)\, dx = \int_{\partial T} \vec{F}(u,q)\cdot \vec{n}\, ds; \tag{4.2}$$

where T does not have to be a triangle and $\vec{n}$ is the outward unit normal. This quantity represents the global flux $\vec{F}$ 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

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

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

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

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

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

This can be straightforwardly rewritten in term of distribution coefficients:

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

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

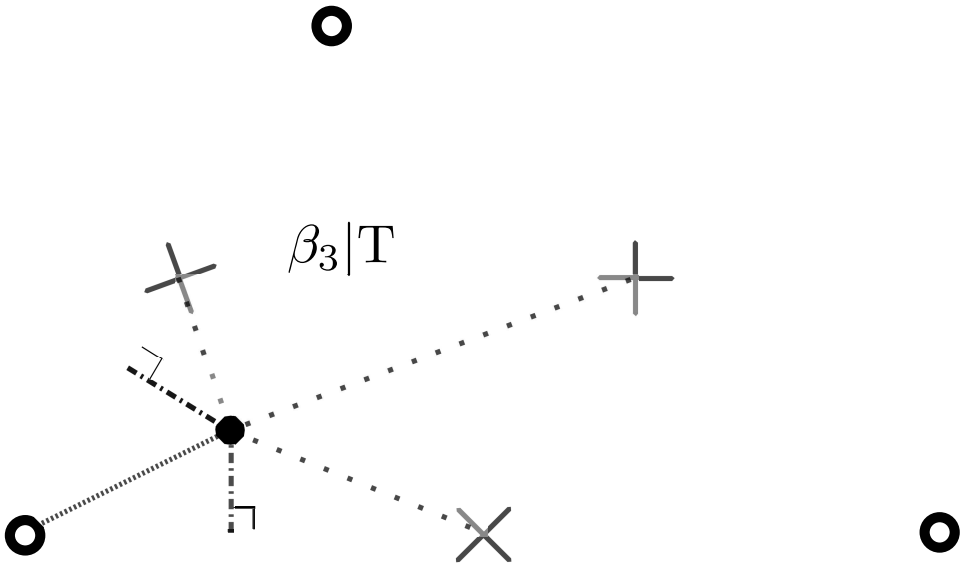


Figure 4.1: Find barycentric coordinates $(\lambda_1, \lambda_2, \lambda_3)$ of B such that quadrilaterals 14B6, 24B5 and 36B5 have areas $\alpha_1|T|$, $\alpha_2|T|$ and $\alpha_3|T|$ respectively. $|14B6| = \lambda_2 \frac{|T|}{2} = \alpha_1|T|$ and the same reasoning being true for the two other vertices, one gets $\lambda_i = 1 - 2\alpha_i$, $i = 1, \ldots, 3$

$$\sum_{T \in D_i} \Phi_i^T = 0, \quad \forall M_h. \tag{4.5}$$

4.1.2 Geometrical Interpretation in the P^1 Case

Let consider a P^1 mesh. Each triangle has three degrees of freedom. Because $\sum_{i \in T} \alpha_i^T = 1$, 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 $\alpha_i^T|T|$. This point has barycentric coordinates $(1 - 2\alpha_1, 1 - 2\alpha_2, 1 - 2\alpha_3)$, see figure 4.1. If we define the new control cell associated to node i with these quadrilaterals, and denote it by C_i, we obtain that the integral of equation (4.1) on each control cell gives the expression of the scheme (4.5)

$$\forall M_h, \sum_{T \in D_i} \Phi_i^T = \int_{C_i^\beta} \operatorname{div}(\vec{F}(u_h, \nabla u_h))\, dx = 0.$$

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 (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 α_i^T depends on the value of the solution u_h. Then the problem writes:

Figure 4.2: 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 $\tilde{r}:u_h$ 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 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 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 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 its associated median dual cell C_i, generated by the midpoint of the edges and the barycentric centers of the elements it 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:

$$u_h \in \{f : \Omega \to \mathbb{R},\ \forall C_i \in \mathcal{M}_h,\ f_{|C_i} \text{ is constant}\}.$$

FV formulation of continuous scalar equation (4.1) reads

$$\sum_{I_{ij} \in \partial C_i} \int_{I_{ij}} \tilde{\Phi}(u_i, u_j, n)\, dl = 0. \tag{4.6}$$

where $\tilde{\Phi}(u, v, n)$ stands for the FV numerical flux, I_{ij} is the portion of ∂C_i separating C_i from C_j (see Figure 4.2) and n is the outward unit normal.

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

$$\Phi_i^T = \sum_{j \in T, j \neq i} \left(\tilde{F}_h(u_j, u_j) - \tilde{F}_h(u_i, u_j) \right) \cdot n_{ij} \quad Q(u_i, u_j, n_{ij})(u_j - u_i),$$

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 FV numerical fluxes and not only to (4.7). Surprisingly, starting from the piecewise constant 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 (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 P^1 numerical resolution. We have in that case to solve the finite dimensional problem:

$$\int_\Omega \nabla \cdot \tilde{F}_h(u_h) \varphi_i \, dx = 0, \quad \forall \varphi_i \in V_h. \tag{4.9}$$

φ_i denotes the 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 $\tilde{F}$ is continuously approximated by its P^1 projection $\tilde{F}_h$, $\nabla \cdot \tilde{F}_h(u_h)$ is constant over every element and we obtain

$$\forall \varphi_i \in V_h, \quad \sum_{T \in D_i} \int_T \nabla \cdot \tilde{F}_h(u_h) \varphi_i \, dx = 0$$

$$\sum_{T \in D_i} \frac{1}{3} \int_T \nabla \cdot \tilde{F}_h(u_h) \, dx$$

$$\sum_{T \in D_i} \frac{1}{3} \Phi^T.$$

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

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

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]):

$$\alpha \cdot \nabla u = 0. \tag{4.10}$$

A new class of schemes has been developed [73, 25, 72] in order to stabilize the 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:

$$\int_\Omega \nabla \cdot \tilde{F}_h(u_h) \varphi_i \, dx \quad \sum_{T \in D_i} \int_T \left(\frac{\partial \tilde{F}_h}{\partial u} \cdot \nabla \varphi_i \right) \nabla \cdot \tilde{F}_h(u_h) \, dx = 0, \quad \forall \varphi_i \in V_h. \tag{4.11}$$

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

$$k_i^T = \int_T \frac{\partial \vec{F}}{\partial u} \cdot \vec{n}_i \, dx; \tag{4.12}$$

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

$$\forall P \in M_h; \quad \sum_{T \in D_i} \left(\frac{1}{3} - \frac{k_i^T \bar{\ }}{|T|} \right) \Phi^T;$$

which means they fit the RDS formalism with distribution coefficients

$$\beta_i^T = \frac{1}{3} - \frac{k_i^T \bar{\ }}{|T|}.$$

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 $\vec{n}_i$ 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 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 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.

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 RDS is a particular *finite element* weak formulation with modified basis functions. That for, we define what we call the *Bubble Functions* β^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

$$N_i^T = \Phi_i^{T;1} + \beta_i^T \beta^T \tag{4.13}$$

as a new linear form function over the element, with β_i^T a fitting parameter. The extra nodal form function $\beta_i^T \beta^T$ will also be denoted by β_i^T. In order the scheme stays conservative, we need to ensure the following condition:

$$\sum_{i \in T} N_i^T = 1 \Rightarrow \sum_{i \in T} \beta_i^T = 0. \tag{4.14}$$

Let us apply the *finite element* theory to equation (4.1) with the approximated functional space being spanned by the N_i^T. We furthermore assume that

$$\forall T \in M_h, \forall P \in M_h, \quad \beta_i^T = -3 \beta_i^T. \tag{4.15}$$

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

Then in P^1, the scheme writes

$$@PM_h; \quad \int_{TPD_i} N_i^T \operatorname{div}\left(\tilde{F}_h(u_h)\right) dx \qquad 0$$

$$\int_{TPD_i} N_i^T dx \frac{T}{|T|} \int_{TPD_i} \frac{T}{i} T;$$

which is exactly the P^1 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

$$\frac{T}{i} \quad \frac{\int_T \frac{T;k}{i}}{\int_T {}^T \operatorname{div}\left(\tilde{F}_h(u_h)\right) dx} \frac{T;k}{i} \tag{4.16}$$

were always de ned. But it is not always the case, as $\operatorname{div}\left(\tilde{F}_h(u_h)\right)$ is no more constant in $P^k; k \geq 1$, and can take positive as negative values inside T.

4.2 Properties of RDS

This section is devoted to the de nition of the numerical properties ofRDS . 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-Wendro -Like Theorem has been

We have here used a convention that is going to be really useful in the rest of the manuscript. In the last equation, $\mathbf{n}$ represents the generic outward unit normal to the edges of the triangle, while $\mathbf{n}_i$ 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 $\tilde{\mathbf{F}}$ 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 $\hat{\mathbf{F}}_h$ of $\tilde{\mathbf{F}}$ must be.

Assumption 4.3

The approximation $\hat{\mathbf{F}}_h$ of the flux $\tilde{\mathbf{F}}$ verifies:

i) $\tilde{\mathbf{F}}_h$ is a continuous function from X_h^k into X_h^k,

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

$$\lim_{h\to 0} \left\| \hat{\mathbf{F}}_h(u_h) - \tilde{\mathbf{F}}(u) \right\|_{L^1_{loc}(\mathbb{R}^2)} = 0.$$

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

$$\hat{\mathbf{F}}_h(v) = \sum_{i \in \mathcal{M}_h} \tilde{\mathbf{F}}(v_i)\, \beta_i^k. \tag{4.17}$$

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 $u \in L^2(\mathbb{R}^2)$ such that

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

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

Then u is a weak solution of (4.1).

Proof: Let Υ be any 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

$$\Psi_i^T(u_h) = \int_T \nabla \varphi_i^k \cdot \tilde{\mathbf{F}}(\hat{\mathbf{F}}_h(u_h))\, dx; \tag{4.18}$$

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:

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

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

$$\sum_{i\in\mathrm{T}}\left(\Phi_i^{\mathrm{T}}\left(u_h\right) \quad \Psi_i^{\mathrm{T}}\left(u_h\right)\right) \quad \Phi^{\mathrm{T}} \quad \Phi^{\mathrm{T}} \quad 0;$$

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

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

We rst begin with term II :

$$\begin{aligned}
\mathrm{II} \quad & \sum_{\mathrm{T}\in\mathcal{M}_h}\sum_{i\in\mathrm{T}}\int_{\mathrm{T}}{}'_i^{k}(x)\,r\,:F_h(u_h)\Upsilon_i\,dx && (4.20a)\\
& \int\left({}_h^{k}\Upsilon\right)(x)\,r\,:F_h(u_h)\,dx && (4.20b)\\
& \int r\,p_h^{k}\Upsilon\,:F_h(u_h)\,dx \quad \int r\,\Upsilon:F(u)\,dx\\
& \int r\,\Upsilon:F(u)\,dx && (4.20c)\\
& \int r\,\Upsilon:F(u)\,dx \quad \int\left(r\,\Upsilon \quad r\,p_h^{k}\Upsilon\right):F_h(u_h)\,dx\\
& \int r\,\Upsilon:\left(F(u) \quad F_h(u_h)\right)dx && (4.20d)\\
& \int r\,\Upsilon:F(u)\,dx \quad o_h(1) && (4.20e)
\end{aligned}$$

In equation (4.20b), we just use the fact that$\sum_{i\in\mathrm{T}}\Upsilon_i{}'_i^{k}$ is the P^k projection of C^1 test function Υ. In equation (4.20c), we apply the Green formula, enjoying the compact support ofΥ 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 theL^1 norm of $r\,\Upsilon \quad r\,p_h^{k}\Upsilon$because the sequence of$u_h$ is bounded inL^8 norm and F_h is a continuous function onX_h^{k}. And since Υ is a C_0^1 function in Ω,

$$\|r\,\Upsilon \quad r\,p_h^{k}\Upsilon\|_{\mathcal{L}^1(\mathbb{R}^2)} \quad o_h(1);$$

BecauseΥ is 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 $\|F(u) \quad F_h(u_h)\|_{\mathcal{L}^1(\mathbb{R}^2)}$ which tends to 0 by assumption 4.3(ii), as$\|u_h\|_8$ is bounded independently of$h$, and $u_h \underset{h\to 0}{\to} u$ in L^2_{loc}.

Let give a look to term I. We rst obviously have

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

and sinceΥ is C_0^1 in Ω, $|\Upsilon_i \quad \Upsilon_j|$ is dominated by$h:\sup\|r\,\Upsilon\| \quad Ch$. Then

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

and by assumption 4.2, we obtain

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

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

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

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:

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

$$\int_{\mathbb{R}^2} \varphi \frac{\partial u}{\partial t} + F(u) \cdot \nabla \varphi \, dx = o_h(1)$$

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 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 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

$$\frac{\partial u}{\partial t} + \vec{\nabla} \cdot \vec{F}(u) = 0, \quad \forall (x,t) \in \Omega \times [0,T] \tag{4.24}$$

is approximated by

$$\forall \sigma \in M_h, \quad \frac{u_i^{n+1} - u_i^n}{\Delta t} + \sum_{T \in D_i} \Phi_i^T(u_h^n) = 0. \tag{4.25}$$

The proof of the Lax Wendroff-Like theorem now needs a test function φ, C_0^1 both in space and time and prove that the term $\frac{u_i^{n+1} - u_i^n}{\Delta t}$ implies

$$\int_{\Omega \times [0,T]} u \frac{\partial \varphi}{\partial t} \, dx \, dt + \int_{\Omega} u_0(x,0)\varphi(x,0) \, dx + o_{h:dt}(1) \tag{4.26}$$

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 **o**vershoot 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 L^2) but not in another (for example L^∞). Then during a certain amount of time steps, the distance of the computed solution to the real one could decrease in L^2 norm, but exponentially grow in the L^∞ one. Such a situation always leads to a numerical blow up.

Property 4.6 (Local Extremum Decreasing)

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

$$\tilde{c}_{ij} \geq 0; \quad \forall j \neq i \tag{4.32}$$

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 $\sum_{j \in \mathcal{D}_i} \tilde{c}_{ij} (u_i^n - u_j^n)$ is negative. At next time step, we will have : $u_i^{n+1} \leq u_i^n$.

Exactly like in the maximum case, ifu_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 L^∞ 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.

De nition 4.7 (Monotonicity Preserving Property)

The above explicit scheme is called*Monotonicity Preserving* if

$$\forall M \in \mathcal{M}_h, \forall T \in \mathcal{D}_i, \forall T' \quad c_{ij}^T \geq 0: \tag{4.33}$$

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 L^∞ 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 L^∞ norm for the solution obtained by this method. The solution of an implicit monotonicity preserving 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

$$\forall M \in \mathcal{M}_{np}(\mathbb{R}), \quad M \geq 0 \Leftrightarrow x^T M x \geq 0; \forall x \in \mathbb{R}^n : \tag{4.34}$$

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* $(\epsilon_i)_{i \in \mathcal{M}_h}$ by

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

$\Pi_h^k u$ 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:

$$\Theta(\Pi_h^k u) = \sum_{i \in \mathcal{M}_h} \Upsilon_i \epsilon_i = \sum_{i \in \mathcal{M}_h} \Upsilon_i \sum_{T \in \mathcal{D}_i} \Phi_i^T(\Pi_h^k u) \tag{4.36}$$

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

$$\Theta(\Pi_h^k u) = O(h^k)$$

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

$$\Psi_i^T(u_h) = \int_T \phi_i^k \vec{\nabla} \cdot \vec{F}_h(u_h) dx,$$

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

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

we obtain

$$\Theta\left(u_h^k\right) = \underbrace{\frac{1}{|\Omega|}\sum_{T\in\mathcal{M}_h}\sum_{i\in T}\Phi_i^T\left(u_h^k\right)\Psi_i^T\left(u_h^k\right)\left(\Upsilon_i-\Upsilon_j\right)}_{I} + \underbrace{\sum_{T\in\mathcal{M}_h}\sum_{i\in T}\Psi_i^T\left(u_h^k\right)\Upsilon_i}_{II} \tag{4.37}$$

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

$$\int_\Omega \Upsilon_h^k \vec{\nabla}\cdot\vec{F}(u)\,dx = 0;$$

and

$$II = \int_\Omega \Upsilon_h^k\,\vec{\nabla}\cdot\vec{F}_h\left(u_h^k\right) + \vec{\nabla}\cdot\vec{F}(u)\,dx - \int_\Omega \vec{\nabla}\cdot\left(\Upsilon_h^k\,\vec{F}_h\left(u_h^k\right)\right) + \vec{\nabla}\cdot\left(\Upsilon_h^k\,\vec{F}(u)\right)dx$$

Now, u_h^k is a P^k approximation of u, $\vec{F}$ is supposed to be continuous and $\vec{\nabla}\Upsilon_h^k$ is bounded, because $\Upsilon \in P_0^1(\Omega)$. Then if $\vec{F}_h$ is an approximation of flux $\vec{F}$ of order $k+1$, we have:

$$II = O\left(h^{k+1}\right) \tag{4.38}$$

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 $O\left(h^{-2}\right)$ and because the gradient of Υ is bounded in Ω, $\Upsilon_i - \Upsilon_j = O(h)$. What gives:

$$I = O\left(h^{-2}\right)O(h)\,O\left(\Phi_i^T\left(u_h^k\right)\right)O\left(\Psi_i^T\left(u_h^k\right)\right) \tag{4.39}$$

But

$$\begin{aligned}\Psi_i^T\left(u_h^k\right) &= \int_\Omega \varphi_i^k\,\vec{\nabla}\cdot\vec{F}_h\left(u_h^k\right)dx \\ &= \int_\Omega \varphi_i^k\,\vec{\nabla}\cdot\vec{F}_h\left(u_h^k\right) + \vec{\nabla}\cdot\vec{F}(u) - \vec{\nabla}\cdot\vec{F}(u)\,dx \\ &= \int_\Omega \vec{\nabla}\cdot\left(\varphi_i^k\,\vec{F}_h\left(u_h^k\right) - \vec{F}(u)\right) - \vec{\nabla}\varphi_i^k\cdot\left(\vec{F}_h\left(u_h^k\right) - \vec{F}(u)\right)dx \\ &= \int_{\partial T}\left(\varphi_i^k\,\vec{F}_h\left(u_h^k\right) - \vec{F}(u)\right)\cdot\vec{n}\,dx - \int_T \vec{\nabla}\varphi_i^k\cdot\left(\vec{F}_h\left(u_h^k\right) - \vec{F}(u)\right)dx \\ &= O\left(h^{k+2}\right)\end{aligned}$$

Then the *truncation error* $\Theta\left(u_h^k\right)$ is of desired order $k+1$, if $\Phi_i^T\left(u_h^k\right)$ 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 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) $\hat{\mathbf{F}}_h$, the flux approximation, is of order $(k+1)$;

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

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

Condition (4.40) guarantees that the scheme has formally a $O(h^{k+1})$ 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 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 $(k+1)^{th}$ accuracy needs in particular that $\Phi_i^T(\pi_h^k u) = O(h^{k+2})$. 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.

$$\begin{aligned}
\Phi^T(\pi_h^k u) &= \int_T \nabla \cdot \hat{\mathbf{F}}_h(\pi_h^k u)\,dx \\
&= \oint_{\partial T} \left(\hat{\mathbf{F}}_h(\pi_h^k u) - \mathbf{F}(u) \right) \cdot \mathbf{n}\, dx \\
&= O(h^{k+2}).
\end{aligned}$$

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

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

and

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

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

Definition 4.10

A RD scheme is called *Linearity Preserving* (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:

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

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-LP scheme into a LP one. This method will be used to transform any known RDS of any order of accuracy into a scheme having the maximal order of accuracy.

4.3 Godunov Theorem

Before presenting some classical RD schemes, and analyze their properties, we wish to present the following theorem that is restricting the panel of possible 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 P^k formulation, one wishes then to obtain a scheme that is both $k+1^{th}$ *order accurate* and *monotonicity preserving*. The following theorem claims [50, 76]:

Theorem 4.12 (Godunov)

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

Proof : This proof is given here because it is valuable for any 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 LP linear scheme on an element T having q DoFs. Then the distribution coefficients Φ_i^T; $i \in T$ as well as the c_{ij} are independent of the solution u. We recall:

$$\Phi_i^T = \beta_i^T \Phi^T = \sum_j c_{ij} (u_i - u_j) \tag{4.42}$$

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

$$\sum_{i \in T} \Phi_i^T = \Phi^T \sum_{i \in T} \sum_{j \in T} c_{ij} (u_i - u_j) = \sum_{i \in T} k_i u_i$$

where k_i coefficients are also independent of u and moreover verifying

$$\sum_{i \in T} k_i = \sum_{i,j \in T} c_{ij} (u_i - u_j) = 0; \tag{4.43}$$

what allows us to write

$$\Phi^{\mathrm{T}} \quad \sum_{j \quad i} k_j \, pu_j \quad u_i q: \tag{4.44}$$

Then by (4.42), one gets

$$\sum_{j \quad i} c_{ij} \, pu_i \quad u_j q \quad \sum_{j \quad i} \quad {}_i^{\mathrm{T}} k_j \, pu_i \quad u_j q; \tag{4.45}$$

and by identi cation, because all the coe cients of the sums are independent ofu,

$$c_{ij} \qquad {}_i^{\mathrm{T}} k_j: \tag{4.46}$$

Finally, that means that $\sum_{j \mathrm{PT}} c_{ij} \quad 0$ and at least onec_{ij} is negative. This contradict the fact that the scheme ismonotonicity preserving, see equation (4.33). ■

4.4 Some 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 (FV) and the Lax-Wendroff scheme (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

$$\frac{Bu}{Bt} \quad \sim : r \, u \quad 0; \quad x \, P \, \Omega \, € \, R^2: \tag{4.47}$$

$\sim$ 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 n_i as the inward normal to the opposite edge of node i, scaled by the length of this edge. Then the quantity

$$k_i \quad \frac{\sim : n_i}{2} \tag{4.48}$$

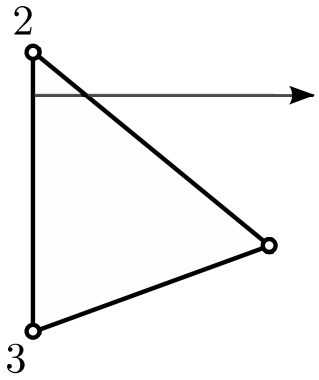


Figure 4.5: Left: 1-*Target* triangle. Node 1 is the only downstream node. It receives the global residual Φ^T entirely. Right: 2-*Target* triangle. Node 1 is upstream and receives nothing from the global residual.

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:

$$\forall T \in \mathcal{M}_h, \forall P \in T, \quad k_i \cdot n_{iT} \leq 0 \Rightarrow \Phi_i^T = 0. \tag{4.49}$$

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* 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* 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, β 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 n_{iT}, 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 Λ is a diagonal matrix, then $|\Lambda|$ is the diagonal matrix formed by the absolute values of the diagonal elements of Λ. Now if $K = R\Lambda L$ is a diagonalizable matrix, then

$$|K| = R|\Lambda|L$$

and we now define

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

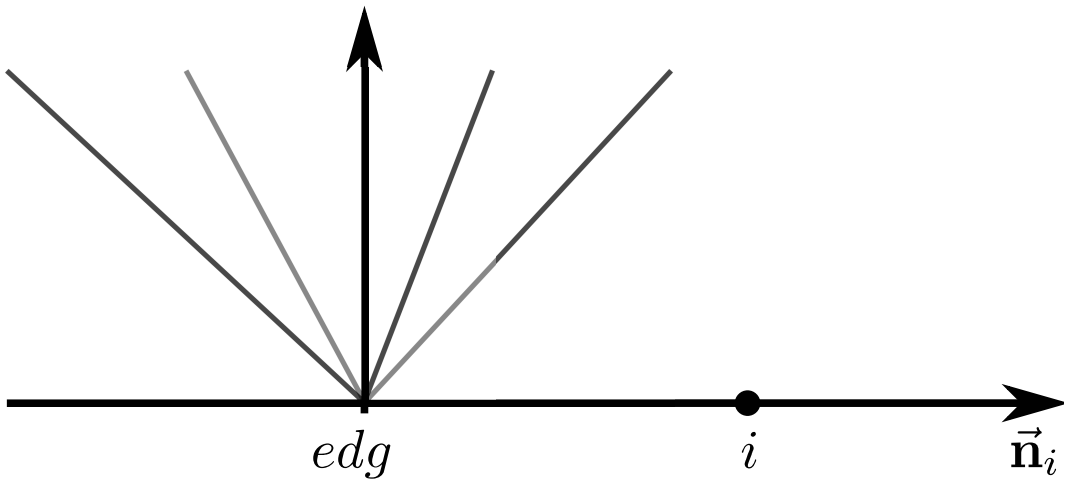


Figure 4.6: (Multidimensional Upwind) One dimensional characteristic problem. $_1$; $_2$ 0, $_3$; $_4$ ¡ 0. Then node i should receive information only on the eigendirections r_3 and r_4: $\Phi_i^T : r_1$ $\Phi_i^T : r_2$ 0.

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

$$k \quad \min(k; 0)u, \quad k \quad \max(k; 0)u$$

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 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:

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

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

$$\sum_{i \in T} \Phi_i^N \quad \sum_{i \in T} (k_i (u_i - \tilde{u})) \quad \sum_{i \in T} k_i (u - \tilde{u})$$
$$\Phi^T \quad \int_T (r u) \, dx \quad \sum_{i \in T} u_i \int_T (r \varphi_i^1) \, dx$$
$$\sum_{i \in T} k_i u_i :$$

And because k_i k_i k_i and

$$\sum_{i \in T} k_i \quad 0 \Rightarrow \sum_{i \in T} k_i \quad \sum_{i \in T} k_i$$

we have:

$$\tilde{u} \quad \frac{\sum_{i \in T} k_i \, u_i}{\sum_{i \in T} k_i} \tag{4.51}$$

A big problem of this scheme, is that nothing ensures $\sum_{i \in T} k_i$ 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

$$N \quad \left(\sum_{i \in T} k_i \right)^{-1} : \tag{4.52}$$

The N scheme is then recast into the form

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

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

$$c_{ij}^N \quad k_i \, N k_j \geq 0; \quad \forall j \in T:$$

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

$$\frac{\Phi_i^T}{\Phi^T};$$

and the N scheme is not 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 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 LP.

4.4.3 The LDA Scheme

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

$$\Phi_i^{LDA} \quad \beta_i^{LDA} \, \Phi^T; \quad \beta_i^{LDA} \quad k_i \, N: \tag{4.53}$$

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

$$c_{ij}^{LDA} \quad k_i \, N k_j : \tag{4.54}$$

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 P^k formulation, $k \geq 1$. Let us keep the example of the scalar advection problem (4.47) to illustrate this. The scheme can easily be extended to 2D P^2 problems, with

$$k_i \int_T \tilde{\nabla}\cdot(r\,\varphi_i^2)\,dx;$$

φ_i^2 being the P^2 Lagrangian function associated to node i. In that particular case, the scheme is well defined, because $\int_T \tilde{\nabla}\cdot(r\,\varphi_i^2)\,dx$ is non null for all the degrees of freedom i. But if we go now to a 3D problem,

$$\varphi_i^2 = \varphi_i(2\varphi_i - 1) \quad \tilde{\nabla}\cdot(r\,\varphi_i^2) = \tilde{\nabla}\cdot(r\,\varphi_i)(4\varphi_i - 1), \quad i = 1:::3$$

and because $\tilde{\nabla}\cdot(r\,\varphi_i^1)$ is constant over the tetrahedron and $\int_T \varphi_i^1\,dx = \frac{|T|}{4}$, we have

$$\int_T \tilde{\nabla}\cdot(r\,\varphi_i^2)\,dx = 0, \quad i = 1:::3. \tag{4.55}$$

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 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:

$$\int_T \tilde{\nabla}\cdot(r\,\varphi_{10}^2)\,dx = 0; \tag{4.56}$$

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 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

$$\Phi^{T_X} = \sum_{i=1}^{6} u_i \int_{T_X} \tilde{\nabla}\cdot(r\,\varphi_i^2)\,dx; \quad X = I;:::;IV; \tag{4.57}$$

Compute the first order distribution coefficients in T_X using

$$k_j^{T_X} = \frac{\tilde{\nabla}\cdot(n_j)_{T_X}}{2}; \quad j \in T_X; X = I;:::;IV; \tag{4.58}$$

Distribute the global residual

$$\Phi^T = \int_T \tilde{\nabla}\cdot(r\,u_h)\,dx = \sum_{X=I}^{IV} \int_{T_X} \tilde{\nabla}\cdot(r\,u_h)\,dx; \tag{4.59}$$

by sub-triangle, using equation (4.53).

Figure 4.7: Convention of numbering of theP^2 sub-triangles.

Because it uses the rst order distribution coe cients by sub-triangles, this method is always de ned 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 computing1 global residual and distributing it to $\frac{k+1)(k+2)}{2}$ DoFs, one has to interpolatek^2 global residuals on the sub-triangles and distribute each of them to the3 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 LP (as the N-Scheme), the other one being on the contraryLP but not monotone (as the LDA-Scheme). One can nd good examples of these schemes in [7, 2].

The idea is to de ne a new scheme by

$$\beta_i^B = \theta\, \beta_i^N + (1-\theta)\, \beta_i^{LDA}; \tag{4.60}$$

whereθ of course depends on the solutionu_h. Then the challenge is to nd the correct criterion de ning the blending parameterθ, 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 RD scheme ever designed, for it is multidimensional upwind, conservative, 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 coe cients. We then would like to build new distribution coe cients $\beta_i^{T;+}$; $i \in T$, such that:

$$\circ \quad \sum_{i \in T} \beta_i^{T;+} \leq 1, \text{ in order to keep the conservative property;}$$

Limit the first order distribution coefficients in T_X

$$\beta_i^{T_X;} \quad \frac{\beta_i^{T_X;1}}{\sum_{j \in T_X} \beta_j^{T_X;1}}$$

Distribute the global residuals

$$\Phi_i^{T_X;2} \quad \beta_i^{T_X;} \Phi^{T_X;2}; \quad X \quad I;:::;IV:$$

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 $\beta_i^{T_X;}$ 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 RD scheme when P^1 formulation is used. The scheme writes:

$$\Phi_i^{SUPG} \quad \frac{\Phi^T}{3} \quad \int_T \tilde{p}:r\,\widehat{\phi}_i^k q - \tilde{p}:r\,\widehat{u}_h^k q dx; \tag{4.62}$$

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 P^1 case, the matrix $^-$ being defined in subsection 4.1.3, it is classical calculation to determine the distribution coefficients

$$\beta_i^{SUPG} \quad \frac{1}{3} \quad \frac{k_i^T\,^-}{|T|} \quad \frac{1}{3} \quad \frac{k_i^T}{\sum_{j \in T} |k_j^T|}: \tag{4.63}$$

It is then straightforward the β_i^{SUPG} are bounded, and the scheme is 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 P^k formulations ($k \geq 1$) 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 (P^k, Q^k, or whatever). As we are going to see, it is also monotone and first order, and can be turned into an LP scheme easily, using the same technique recasting the first order N-scheme into the 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:

$$\Phi_i^{LxF} \quad \frac{1}{q} \quad \Phi^T \quad {}^T \sum_{j \in T} \quad u_i \quad u_j \, q \quad : \tag{4.64}$$

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

$$\sim : r \quad u \quad 0: \tag{4.65}$$

The unknown u may be scalar or vectorial.

$$\begin{aligned} \Phi^T \quad & \int_T \sim : r \quad u_h \, dx \\ & \sum_{i \in T} u_i \int_T \sim : r \quad {}_i^k \, dx \\ & \sum_{i \in T} \bar{k}_i^k u_i \quad \sum_{j \in T} \bar{k}_j^k \, u_i \quad u_j \, q \end{aligned}$$

Then, we can rewrite the scheme as

$$\Phi_i^{LxF} \quad \sum_{j \in T} \frac{{}^T \quad \bar{k}_j^k}{q} \, u_i \quad u_j \, q, \tag{4.66}$$

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

$$c_{ij}^T \quad \frac{{}^T \quad \bar{k}_j^k}{q} : \tag{4.67}$$

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

$$@T \in M_h; \quad {}^T \geq \quad \bar{k}_j^k \quad ; @ \in T \tag{4.68}$$

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

$$\text{@T PM}_h; \quad {}^{T} \text{¥ } \bar{k}_j^k; \text{@ PT} \tag{4.68}$$

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

$$\begin{matrix} T \\ i \end{matrix} \quad \frac{\Phi_i^{LxF}}{\Phi^T}$$

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 pk 1qth 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: $U = D \in R^m$. 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 U to take any value in 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.

$$D = \{U = (\rho, \rho u, \rho E) \in R^m, \ \rho > 0, \rho E - \frac{\rho |u|^2}{2} > 0\}$$

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

$$\text{Find } U \in D, \text{ such that } \begin{cases} \sum_{T_i \in \mathcal{T}_h} \int_{T_i} \nabla \cdot F(U) = 0, & \forall x \in \Omega \\ \text{Boundary Conditions.} \end{cases} \tag{5.1}$$

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 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 P^2 triangles, but details could be given for even higher representation of the data in triangles or for 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 RD scheme into an 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 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

$$\Phi^T \quad \int_T \operatorname{div} \tilde{\mathbf{F}}_h(\mathbf{U}_h) \, dx \tag{5.2a}$$

$$\int_{\partial T} \tilde{\mathbf{F}}_h(\mathbf{U}_h) \cdot \mathbf{n} \, ds. \tag{5.2b}$$

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

$$\tilde{\mathbf{F}}_h(\mathbf{U}) \quad \sum_{i \in \mathcal{M}_h} \tilde{\mathbf{F}}_i \varphi_i^k; \tag{5.3}$$

where

$$\tilde{\mathbf{F}}_i \quad \tilde{\mathbf{F}}(\mathbf{U}_i) \quad \tilde{\mathbf{F}}(\mathbf{U}(\mathbf{x}_i)).$$

Then, the approximated flux $\tilde{\mathbf{F}}_h$ 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

$$\Phi^T \quad \sum_{edge \in \partial T} \sum_{i \in edge} \frac{\tilde{\mathbf{F}}_i}{k\mathbf{n}_{edge}k} \int_{edge} \varphi_i^k ds \quad \cdot \mathbf{n}_{edge}; \tag{5.4}$$

which is just a linear combination of the values taken by $\tilde{\mathbf{F}}$ at the DoFs of T, with coefficients $\frac{1}{k\mathbf{n}_{edge}k} \int_{edge} \varphi_i^k ds$. 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 $k \quad 1:::3$ in a triangle, the numbering being defined on Figure 3.7 page 50, and $\mathbf{n}_i$ 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.

P^1:

$$\Phi^T \quad \sum_{i \quad 1}^{3} \frac{\tilde{\mathbf{F}}_i \cdot \mathbf{n}_i}{2} \tag{5.5}$$

P^2:

$$\Phi^T \quad \sum_{i \quad 1}^{3} \frac{\tilde{\mathbf{F}}_i \cdot \mathbf{n}_i}{6} \quad \sum_{i \quad 4}^{6} \frac{2}{3} \tilde{\mathbf{F}}_i \cdot \mathbf{n}_i \tag{5.6}$$

P^3:

$$\Phi^T = \sum_{i=1}^{3} \frac{\tilde{F}_i \cdot n_i}{8} \quad \sum_{i=4}^{9} \frac{3\tilde{F}_i \cdot n_i}{8} \tag{5.7}$$

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 P^3 triangle, the value of the *global residual* does not depend of the value of $\tilde{F}_{10}$. 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

$$\Phi_i^{LxF} = \frac{1}{q} \Phi^T = \sum_{j \in T} \alpha_T (U_i - U_j) \quad ; \tag{5.8}$$

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 $\frac{\Phi^T}{q}$ will become insignificant compared to the second term $\sum_{j \in T} \alpha_T (U_i - U_j)$ 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

$$\alpha_T \geq \bar{k}_j^k \quad ; \quad \forall j \in T: \tag{5.9}$$

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:

$$\alpha_T = \max_{i \in T} \|k_i n_i\| \quad c_i \|n_i\| \leq \max_{edge} |edge| \tag{5.10}$$

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 LP^1 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*:

$$\beta_i^T = \begin{cases} \frac{\Phi_i^{LxF}}{\Phi^T}; & \text{if } \Phi^T \neq 0; \\ 0; & \text{else.} \end{cases} \tag{5.11}$$

and use the limitation technique already presented in (4.61) to get the $(k+1)^{th}$ order *Distribution Coefficients*:

$$\beta_i = \frac{\beta_i^T}{\sum_{j \in T} \beta_j^T}. \tag{5.12}$$

We recall that this formula is always defined as $\sum_{j \in T} \beta_j^T \geq 1$ if $\Phi^T \neq 0$. Using this procedure, the new limited scheme with β_i distribution coefficients has the following properties:

The scheme is **conservative**

$$\sum_{i \in T} \beta_i = 1. \tag{5.13}$$

The scheme is **linearity preserving** . β_i is always defined because $\sum_{j \in T} \beta_j^T \geq 1$ when $\Phi^T \neq 0$ and:

$$0 \leq \beta_i \leq 1. \tag{5.14}$$

If the first order scheme is **monotonicity preserving** then the $(k+1)^{th}$ order one is as well because

$$\forall T, \quad \beta_i : \beta_i^T \geq 0. \tag{5.15}$$

If one has

$$\Phi_i^{LxF} \quad \beta_i^T \Phi^T \quad \sum_{j \in T} c_{ij} (u_i - u_j),$$

with positive c_{ij} coefficients, one obtains

$$\Phi_i \quad \beta_i \Phi^T \quad \sum_{j \in T} \frac{\beta_i^T}{\sum_{i \in T} c_{ij}} c_{ij} (u_i - u_j),$$

where $\sum_{i \in T}^{*} c_{ij} \geq 0; \quad \forall j \in T.$

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 L_i of the hyperbolic problem (5.1), evaluated using the average state:

$$\bar{U} \quad \frac{1}{d} \sum_{i \in T} U_i;$$

and in the direction tangential to the stream $\vec{t}_n \quad \frac{n}{\|n\|}$. n denotes here the mean velocity in the triangle *ie.* the velocity vector associated to $\bar{U}$. The left eigenvectors are defined in Subsection 2.2.9. The d projected residuals for a given linear form $L_i \quad \Phi_j^{LxF}$ are then limited using scalar formula (5.12), with

$$\beta_{ij} \quad \frac{L_i \quad \Phi_j^{LxF}}{\sum_{j \in T} L_i \quad \Phi_j^{LxF}} \quad \frac{L_i \quad \Phi_j^{LxF}}{L_i \cdot \Phi^T}.$$

This gives q limited coefficients x_{ij} ; $j = 1:::q$. The limited vector Φ_j is then reconstructed as the vector having coordinates $(x_{ij} q)_{1:::m}$ in the basis of the m right eigenvectors R_i, duals of the L_is. This last paragraph dealing with the limitation of multidimensional RD scheme is summarized in algorithm 1.

Algorithm 1 Vectorial Limitation

```
for i = 0 to m do
  for all j ∈ T do
    α_ij = (L_i(Φ_j^{LxF}) q) / (L_i(Φ^T) q)
    x_ij = α_ij q^+ / Σ_{j∈T} α_ij q
  end for
end for
for all j ∈ T do
  Φ_j = Σ_{i=1}^{m} x_ij R_i
end for
```

Geometrical Representation in the Scalar 2D P^1 Case : Ideally, one would like the limitation also takes into account the *Upwind* property. This would provide a stable $(k+1)^{th}$ order scheme, a perfect scheme. There exists such a limitation technique in the scalar 2D 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 P^1 triangles. In the scalar case, the three distribution coefficients β_i^T define a unique point B in 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 $\tilde{B}$ the image of B by the limitation process, one has: $\tilde{B} = B$. 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 $\beta_j = 0; \forall j \neq i$. $\tilde{B}$ 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 $\tilde{B}$ is situated on the edge opposite to node i. Furthermore, Struijs limitation technique conserves the ratio between the two strictly positive distribution coefficients:

$$\frac{\beta_j}{\beta_k} = \frac{\beta_j^T}{\beta_k^T}.$$

As shown on the left on Figure 5.1, $\tilde{B}$ is limited along the straight line joining B and node i and $\tilde{B}$ is then situated at the intersection between this straight line and the edge opposite to i. Unfortunately, nothing ensures the new distribution point $\tilde{B}$ to be downstream. In the case of Figure 5.1 for example, it is thoroughly possible $\tilde{B}$ stays in region 4, as point $\tilde{B}_1$. $\tilde{B}$ 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 $\tilde{\beta}$, the direction of advection. One possibility

Figure 5.1: Geometrical interpretation of the limitation technique. Point B has barycentric coordinates $T_1; T_2; T_3$. The geometrical transformation $B \to \tilde{B}$ 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 P^1 case, if one considers the unique line defined by B and direction vector $\vec{}$, 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 ∞. We then define B^1 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 $B^1 = B$. Then $\tilde{B}$ is obtained as the Struijs limitation of the barycentric point B^1. This is shown on the right part of Figure 5.1. This gives a very efficient scheme in 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 $\vec{} = (0;1)$:

$$\begin{cases} \vec{}.\nabla u = 0 \\ u(x;0) = \sin^2(\pi x) \\ u(0;y) = u(1;y) = 0 \end{cases} \tag{5.16}$$

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 P^1 scalar problems is not easy at all. We

Figure 5.2: 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 nest 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.

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 L^∞ stability of the scheme, as we will now see.

Scalar Case : If one uses formulation (4.27) on page 72, one has:

$$\forall P \in \mathcal{M}_h, \quad u_i^{n+1} = \left(1 - \omega_i^n \sum_{j \in \mathcal{D}_i} \tilde{c}_{ij}\right) u_i^n + \omega_i^n \sum_{j \in \mathcal{D}_i} \tilde{c}_{ij}\, u_j^n, \tag{5.21}$$

$\tilde{c}_{ij}$ being defined like in (4.31), page 72 as:

$$\tilde{c}_{ij} = \sum_{T \in \mathcal{D}_i \cap \mathcal{D}_j} \beta_i^T c_{ij}^T, \tag{5.22}$$

with c_{ij}^T coming from the first order scheme and $\beta_i^T = \frac{\beta_i^*}{\sum_{T \in \mathcal{D}_i} \beta_i^*} \neq 0$ when $\beta_i^T > 0$ or $\beta_i^T < 0$ 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 u_j^n $_{j \in \mathcal{D}_i}$ if and only if

$$0 \leq \omega_i^n \leq \frac{1}{\sum_{j \in \mathcal{D}_i} \tilde{c}_{ij}}. \tag{5.23}$$

It is then sure

$$\forall P \in \mathcal{M}_h, \quad \min_{j \in \mathcal{M}_h} u_j^n \leq u_i^{n+1} \leq \max_{j \in \mathcal{M}_h} u_j^n;$$

and therefore

$$\forall P \in \mathcal{N}, \forall P \in \mathcal{M}_h, \quad \inf_{x \in \Omega} u_0(x) \leq u_i^n \leq \sup_{x \in \Omega} u_0(x),$$

which is the L^∞ 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 L^∞ stability. As seen in (4.67), page 86, for the LxF scheme the first order monotonicity coefficients verify $\sum_{j \in \mathcal{T}} c_{ij}^T = 0$ and because $0 \leq \beta_i^T \leq 1$,

$$\frac{1}{\sum_{j \in \mathcal{D}_i} \tilde{c}_{ij}} \geq \frac{1}{\sum_{T \in \mathcal{D}_i} \beta_i^T} \geq 0.$$

Then a good and easy estimation of the pseudo time stepping parameter ω_i^n to ensure the monotonicity of the scheme is

$$\omega_i^n = \frac{1}{\sum_{T \in \mathcal{D}_i} \beta_i^T}. \tag{5.24}$$

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 ^{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 $_{\mathcal{T}\mathcal{P}\mathcal{D}_i}\Phi_i^T\mathbf{U}^n\mathbf{q}_{i\mathcal{P}\mathcal{M}_h}$. We usually use the 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:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t^n}{|\mathcal{TPD}_i|}\Phi_i^T\mathbf{U}_h^{n+1}, \quad \forall\mathcal{PM}_h. \tag{5.25}$$

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 Δt, if the nodal residuals can be expressed as (4.27), the scheme (5.25) in its scalar form verifies the global discrete maximum principle

$$\forall\mathcal{PM}_h, \quad \min_{j\mathcal{PM}_h} u_j^n \leq u_i^{n+1} \leq \max_{j\mathcal{PM}_h} u_j^n \tag{5.26}$$

Proof : We start by defining the vector of unknown $\mathbf{U}^n$ by

$$\forall\mathcal{PM}_h, \mathbf{U}^n\mathbf{q}_i = u_i^n,$$

and the two constant vectors $\mathbf{U}_{min}^n$ and $\mathbf{U}_{max}^n$ by

$$\forall\mathcal{PM}_h, \quad \mathbf{U}_{min}^n\mathbf{q}_i = \min_{j\mathcal{P}\mathcal{M}_h} u_j^n, \; \mathbf{U}_{max}^n\mathbf{q}_i = \max_{j\mathcal{P}\mathcal{M}_h} u_j^n.$$

Then one can write $U^n_{min} \leq U^n \leq U^n_{max}$.

If one considers equation (4.27), scheme (5.25) is reformulated into:

$$AU^{n+1} = BU^n \tag{5.27}$$

with

$$\begin{array}{ll} A_{ii} = \frac{1}{\omega_i^n} + \sum_{j \in \mathcal{D}_i} \tilde{c}_{ij} & A_{ij} = -\tilde{c}_{ij} \\ B_{ii} = \frac{1}{\omega_i^n} & B_{ij} = 0 \end{array}$$

$\tilde{c}_{ij}$ being defined by (4.31), page 72. Matrix B has only positive coefficients, then

$$AU^{n+1} = BU^n \geq BU^n_{min} = AU^n_{min}. \tag{5.28}$$

If the scheme is Local Extremum Decreasing the $\tilde{c}_{ij}$ are all positive and A is diagonal dominant. This implies A is invertible and A^{-1} has only positive coefficients [118]:

$$A^{-1}_{ij} \geq 0, \quad \forall j \in \mathcal{M}_h.$$

We can then multiply both sides of (5.28) by 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 U^n_{min} and U^n_{max} such that

$$\forall \mathcal{M}_h, \; (U^n_{min})_i \leq (U^n)_i \leq (U^n_{max})_i.$$

Equation (5.25) is recast into

$$AU^{n+1} = BU^n \tag{5.29}$$

with

$$\begin{array}{ll} A_{ii} = \frac{|}{\Delta t^n_i} + \sum_{j \in \mathcal{D}_i} \tilde{c}_{ij} & A_{ij} = -\tilde{c}_{ij} \\ B_{ii} = \frac{|}{\Delta t^n_i} & B_{ij} = 0 \end{array}$$

where $|$ 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 A being a diagonal block dominant matrix. What is missing is a theorem showing that A must be invertible and that 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 U^n is known, it is impossible to compute $\Phi_i^T(U^{n+1}_h)$. 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

$$\Phi_i^T(U^{n+1}) - \Phi_i^T(U^n) \simeq \sum_{j \in \mathcal{M}_h} \frac{\partial \Phi_i^T(U^n)}{\partial U_j} (U_j^{n+1} - U_j^n) \; : \tag{5.30}$$

Thus, if one uses notation

$$\Delta U_j^n = U_j^{n+1} - U_j^n, \tag{5.31}$$

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

$$\frac{|}{\Delta t^n_i} \sum_{T \in D_i} \frac{\partial \Phi_i^T(U^n)}{\partial U_i} \Delta U_i^n + \sum_{\substack{j \in D_i \\ j \neq i}} \sum_{T \in D_i \cap D_j} \frac{\partial \Phi_i^T(U^n)}{\partial U_j} \Delta U_j^n = \sum_{T \in D_i} \Phi_i^T(U_h^n), \tag{5.32}$$

which is a matrix system in ΔU^n. $|$ is the $m \times d+2$ identity matrix and the right hand side (RHS) is the *explicit* residual.

The main point is at this time to compute the Jacobians of the nodal residuals: $\frac{\partial \Phi_i^T(U^n)}{\partial U_j}$. 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 $\Delta t_i^n \in \mathbb{R}^+$. This matrix is sparse. We have $\frac{1}{\Delta t_i^n}$ 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 Δt_i^n are, the more dominant the diagonal coefficients are. Thus at the limit $\Delta t_i^n \to 0$, we obtain the fully *explicit* scheme. On the other hand, if we consider Δt_i^n going to infinity, the scheme turns into something looking as

$$u_{n+1} = u_n - f'(u_n)^{-1} \cdot f(u_n),$$

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 Δt_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 β^T is considered to be constant. The matrices of the vector of matrices $\frac{\partial \tilde{F}}{\partial U}$ 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

$$\frac{\partial \Phi_i^T(U^n)}{\partial U_j} \begin{cases} \frac{1}{d} w_i \frac{\partial \tilde{F}}{\partial U}(U_i^n, n_i) + (\beta^T - 1) I & \text{if } j = i \\ \frac{1}{d} w_j \frac{\partial \tilde{F}}{\partial U}(U_j^n, n_j) - \beta^T I & \text{if } j \neq i \end{cases} \tag{5.33}$$

where the vector w is the set of coefficients of the linear combination of the $\tilde{\vec{F}}_j \cdot \vec{n}_j$ in the computation of Φ^T, see equations (5.5), (5.6) and (5.7). We recall that $\vec{n}_i$ 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 P^k case

k 1:

$$w = \left[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right]$$

k 2:

$$w = \left[\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}\right]$$

k 3:

$$w = \left[\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{3}{8}, \frac{3}{8}, \frac{3}{8}, \frac{3}{8}, \frac{3}{8}, \frac{3}{8}, 0\right]$$

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 di usive 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 $\frac{\partial \tilde{\vec{F}}}{\partial U}$ 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 $\varepsilon_2 < \| \sum_{T \in \mathcal{D}_i} \Phi_i^T(U^n) \|_{i \in \mathcal{M}_h} \|_2$ 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 ($\varepsilon_2 \approx 10^{-6}$), 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 $l \in [1; mw(l^{th}\text{ variable})]$ of the $pd \times 2q \times pd \times 2q$ block situated at line i and row j is filled in with

$$\frac{\Phi_i^T(U^n + \varepsilon_l V_{jl}) - \Phi_i^T(U^n)}{\varepsilon_l}^T; \tag{5.34}$$

where V_{jl} is a vector having the same size as U^n, having 1 on the line corresponding to the l^{th} variable of node j, and zeros everywhere else. ^{T} represents the vector transposition. ε_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

$$\varepsilon_l = \max(10^{-10}; 10^{-8} \cdot \max_{j \in \mathcal{M}_h} |U_{jl}^n|) \tag{5.35}$$

As one can see, this method requires to compute $\frac{mk(pk+1)q}{2}$ 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 x, but also the directional derivatives $\frac{\partial f}{\partial x}:xd$. 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 $p\{0{:}0; 2{:}0; 0{:}0; 0{:}0; 0{:}0\{q$ for 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 U^n. The critical non differentiable points have been regularized. For example, the absolute value function is replaced by

$$|x| \quad \begin{cases} |x|; & \text{if } |x| \text{ ¤ } \varepsilon \\ \frac{x^2+\varepsilon^2}{2\varepsilon}; & \text{else} \end{cases} \tag{5.36}$$

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

$$\begin{cases} -y\dfrac{\partial u}{\partial x} + x\dfrac{\partial u}{\partial y} = 0, & \forall (x,y) \in [0;1]^2 \\ u(0,y) = \cos^2(\pi(y-\tfrac{1}{4})), & \forall y \in [0;1]. \end{cases} \tag{5.37}$$

   The advection speed $\vec{\beta} = \begin{pmatrix} -y \\ x \end{pmatrix}$ 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

$$t(x,y) \in \Omega, y \leq x \text{ and } y \leq 1 - x,$$

   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 L^∞ 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

$$\sum_{T \in \mathcal{D}_i} \Phi_i^T(u_h) = 0, \quad \forall i \in \mathcal{M}_h. \tag{5.39}$$

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 (rst 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 rst 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 nally rst order. The PSI solution is used as a reference.

Figure 5.5: This gure 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.

properties as $\bar{}$, it is conceivable to use a constant instead, as $\frac{h}{\rho\{u\}\ cq}$ 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 $\bar{}$ will converge faster that its twin using h instead. However, for simplicity, we are usually going to consider that $\bar{}$ 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:

$$u_i^{n\ 1} \quad 1 \quad !_i^n \sum_{j\,PD_i} {}_i^T \tilde{c}_{ij} \quad h \int_T \tilde{p}{:}r\,{}_i^k q^2 dx \quad u_i^n$$
$$\sum_{j\,PD_i} !_i^n \sum_{T\,PD_i X D_j} {}_i^T c_{ij}^T \quad h \int_T \tilde{p}{:}r\,{}_i^k q \tilde{p}{:}r\,{}_j^k q dx \quad u_j^n \tag{5.46}$$

$u_i^{n\ 1}$ is a barycenter of the u_j^n; $j\ P\ D_i$ and the sum of the barycentric coefficients is 1. The scheme verifies a maximum principle if and only if

$$\sum_{T\,PD_i X D_j} {}_i^T c_{ij}^T \quad h \int_T \tilde{p}{:}r\,{}_i^k q \tilde{p}{:}r\,{}_j^k q dx \quad \yen\ 0; \quad @ \quad i:$$

This condition is unreachable as there must exist an element T in which $\int {}_i^T \quad 0$ ñ $\int {}_i^T; \quad 0$ ñ ${}_i^T \quad 0$, and as soon as $\int_T \tilde{p}{:}r\,{}_i^k q^2 dx \ ¡\ 0$, there exists $j\ P\ T$ such that $\int_T \tilde{p}{:}r\,{}_i^k q \tilde{p}{:}r\,{}_j^k q dx$ 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 ${}_T p x; u_h q$ defined by

$${}_T \quad \begin{cases} 1; & \text{where } u_h \text{ is smooth} \\ h; & \text{in the discontinuities} \end{cases} \tag{5.47}$$

There are many possible choices for the parameter ${}_T$. The best choice we have experimented so far is

$${}_T \quad 1 \quad \max_{i\,PT} \quad \max_{T\,PD_i} \max_{j\,PT} \frac{|u_j \quad \bar{u}_T|}{|u_j| \quad |\bar{u}_T| \quad "} \quad ; \tag{5.48}$$

where $" \quad 10^{\ 12}$ or any positive number near to machine zero, and $\bar{u}_T \quad p \underset{j\,PT}{\circ} u_j q \{p \underset{j\,PT}{\circ} 1 q$. 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 $\tilde{}$ 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 $\tilde{}$ into 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 $_T$ (cf. equation (5.48)) is kept compact by updating and swapping the monitors and $\tilde{}$.

1: Initialize by 1 for all DoFs,
2: Set " 10^{12},
3: **for** each iteration k **do**
4: Set $\tilde{}$ 0 for each ,
5: **for** each element T **do**
6: Evaluate the local shock capturing coefficient $_T$, with

$$_T \quad 1 \quad \max_{PT} \quad ; \tag{5.49}$$

7: Evaluate a mean value in T

$$\bar{u}_T \quad \frac{\sum_{j\,PT} u_j}{\sum_{j\,PT} 1} \tag{5.50}$$

8: Evaluate

$$_T \quad \max_{PT} \quad \frac{|u \quad \bar{u}_T|}{|u \;|\; \bar{u}_T| \quad "} \tag{5.51}$$

9: **for** each P T **do**
10: $\tilde{}$ max$p\tilde{}$; $_T$q
11: **end for**
12: **end for**
13: Swap : $\tilde{}$;
14: **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 $pk \quad 1q^2$, and one needs a quadrature formula of at least $pk \quad 1q^2$-th order of accuracy. Term (5.44) is computed as

$$D_i^T \quad hjTj_T \sum_{x_q \mathbb{P} quad} !_q \ \tilde{}px_q q_r \acute{\Upsilon}\tilde{N}_i^k px_q q \quad \tilde{}px_q q_r \acute{\Upsilon}\tilde{N}_h^k px_q q : \tag{5.52}$$

The problem is that a quadrature formula of $pk \quad 1q^2$-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.

$$hjTj_T \ \tilde{}px_q q_r \acute{\Upsilon}\tilde{N}_i^k px_q q \quad \tilde{}px_q q_r \acute{\Upsilon}\acute{\Upsilon}\tilde{N}_h^k u \ px_q q \quad Oph^k \ ^2q, \quad @x_q \mathbb{P} quad:$$

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

$$D_i^T pu; vq \quad hjTj_T \sum_{x_q \mathbb{P} quad} !_q \ \tilde{}px_q q_r \acute{\Upsilon}\tilde{N} upx_q q \quad \tilde{}px_q q_r \acute{\Upsilon}\tilde{N} vpx_q q \tag{5.53}$$

to be positive definite. This reduces to ensure

$$D_i^T pu; uq \quad 0 \quad \grave{u}\tilde{n} \quad \tilde{}:r \acute{\Upsilon}\tilde{N} u \quad 0: \tag{5.54}$$

This condition is met when all the weight coefficients $!_q$ are positive and the quadrature formula uses enough quadrature points to define uniquely the $pk \quad 1q^{th}$ order polynomial $r \acute{\Upsilon}\tilde{N} u_h$. The computation of the stabilization term is sumed up in the three following points:

The formal order of accuracy is unconditionally met;

$@q \mathbb{P} quad; !_q \mathbf{i} \ 0$, for example, $!_q$ is always 1 or $\frac{1}{\#tquadu}$;

Quadrature formula uses $\frac{kpk \quad 1q}{2}$ quadrature points:

$$\#tquadu \quad \frac{kpk \quad 1q}{2}$$

and if we finally consider the general case of a vectorial problem, the practical computation of the stabilization term writes:

$$D_i^T \quad hjTj_T \sum_{q \ 1 \ j \mathbb{P} T}^{kpk \ 1q\{2} U_j \ \tilde{}px_q q_r \acute{\Upsilon}\tilde{N}_i^k px_q q \quad \tilde{}px_q q_r \acute{\Upsilon}\tilde{N}_j^k px_q q \tag{5.55}$$

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 5th 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

$$(J_{Dissip})_{ij} = h|T| \sum_{q=1}^{k(k+1)/2} \sum_{j \in T} \left(\vec{\lambda}(x_q) \cdot \nabla\varphi_i^k(x_q)\right)\left(\vec{\lambda}(x_q) \cdot \nabla\varphi_j^k(x_q)\right) \quad (5.56)$$

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: $u_h(x) = 0;\ x \in \partial\Omega$. 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.

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

$$\begin{matrix} u_x^{wall} & 1 & 0 & 0 \\ u_y^{wall} & 0 & 1 & 0 \end{matrix} : \tag{5.58}$$

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 $v \cdot n = 0$.

As explained in Subsection 2.1.5, page 20, U is the solution of problem (5.1) with boundary conditions, if it verifies, for any $\varphi \in C^1(\Omega)$

$$\begin{aligned} \int_\Omega \nabla \cdot \vec{F}(U)\varphi \, dx &= \int_{\partial\Omega} \vec{F}(U)\varphi \cdot n \, ds = 0, \\ \forall T \in M_h, \quad \int_T \nabla \cdot \vec{F}(U)\varphi \, dx &= \int_{\partial T \cap \partial\Omega} \vec{F}(U)\varphi \cdot n \, ds = 0 \end{aligned} \tag{5.59}$$

with 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, U_h approximates the exact solution as the unique solution of $W_h^k \in \mathrm{Span}_{i \in M_h}(\varphi_i^k)$ 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 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

$$\vec{F}(U)\cdot n = \begin{pmatrix} 0 \\ p n_x \\ p n_y \\ 0 \end{pmatrix} : \tag{5.60}$$

Then for a DoF on the boundary, after applying the Green formula inside T to the left integral, the weak formulation over the mesh M_h reads:

$$\begin{aligned} \forall T \in D_i, \quad & \int_T \varphi_i^k \, \mathrm{div}\left(\vec{F}_h(U_h)\right) dx \\ & + \int_{\partial T \cap \partial\Omega} \varphi_i^k \left(\vec{F}_h(U_h) \cdot n_{|\partial\Omega} - \vec{F}_h(U_h) \cdot n \right) ds = 0; \end{aligned} \tag{5.61}$$

which is the residual distribution plus a additional boundary term enforcing flux

$$\vec{F}_{slip}(U, n) - \vec{F}(U) \cdot n_{|\partial\Omega} + \vec{F}(U) \cdot n = \begin{pmatrix} 0 \\ u \cdot n \\ v \cdot n \\ h \cdot n \end{pmatrix} \tag{5.62}$$

on the boundary edges. $h \quad E \quad p\{$ denotes the specific *enthalpy*.

Without any further explanation, this is exactly what we do in the case of a RDS. We first compute the global residuals and distribute them to their respective DoFs. Afterward, we go all over the edges of M_h lying on the boundary, compute the terms

$$B_i^{edge} \quad \int_{edge} \quad ' \; {}_i^k \tilde{\mathbb{F}}_{slip}(U^n; \mathbf{n}) q ds; \tag{5.63}$$

and add them to the residual of the corresponding boundary DoFs. One has to remark that as $\tilde{\mathbb{F}}_h$ is built as the P^k projection of the continuous flux $\tilde{\mathbb{F}}$, 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

$$M^k_{ij} \quad \int_0^1 \quad ' \; {}_i^k ' \; {}_j^k ds: \tag{5.64}$$

The computational formula writes:

$$B_i^{edge} \quad \sum_{j \in edge} \quad M^k_{ij} \quad \tilde{\mathbb{F}}_h(U_j) q_{(p\mathbf{n}:\mathbf{n} \; 0q} \quad \tilde{\mathbb{F}}_h(U_j) q \; :\mathbf{n}_{edge}; \tag{5.63}$$

where $\mathbf{n}_{edge}$ is still the outward normal to the boundary but its norm is the length ($|edge|$) 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

$$\tilde{\mathbb{F}}(U) q \quad \frac{B\tilde{\mathbb{F}}(U) q}{BU} U \quad \sim (U) qU: \tag{5.65}$$

Now the flux crossing an edge has two components. Because the problem is hyperbolic, if $\mathbf{n}_{edge}$ is the outward normal scaled by the length of the edge, one has

$$\begin{aligned} \tilde{\mathbb{F}}(U) q;\mathbf{n}_{edge} \quad & \sim (U) q;\mathbf{n}_{edge} U \\ & K_{(U;\mathbf{n}_{edge} q} U \\ & K_{(U;\mathbf{n}_{edge} q} U \quad K_{(U;\mathbf{n}_{edge} q} U: \end{aligned} \tag{5.66}$$

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

$$\tilde{\mathbf{F}}_{SW}(U, U_\infty, \mathbf{n}) = K^+_{(U,\mathbf{n})} U_\infty + K^-_{(U,\mathbf{n})} U. \tag{5.67}$$

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

$$\begin{aligned} B_i^{edge,SW} &= \int_{edge} \varphi_i \, \tilde{\mathbf{F}}_{SW}(U^n, U_\infty, \mathbf{n}_{edge}) \, \tilde{\mathbf{F}}_h(U^n, \mathbf{n}_{edge}) \, ds \\ &= \int_{edge} \varphi_i \, K^+_{(U,\mathbf{n}_{edge})}(U_\infty - U) \, ds. \end{aligned} \tag{5.68}$$

Once more the flux is supposed to be of the same polynomial order as the solution, and the Steger-Warming contribution is computed as

$$B_i^{edge,SW} = \sum_{j \in edge} M^k_{ij} \, K^+_{(U_j,\mathbf{n}_{edge})}(U_\infty - U_j) \tag{5.69}$$

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

$$M^k_{ij} \, K^+_{(U_j,\mathbf{n}_{ij})}. \tag{5.70}$$

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,

$$J_{slip} = \frac{\partial \tilde{\mathbf{F}}_{slip}}{\partial U},$$

and the Jacobian of the boundary contributions at line i and row j writes

$$M^k_{ij} \, J_{slip}(U_i, \mathbf{n}). \tag{5.71}$$

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 P^2. 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 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

$$\Phi^T \quad \sum_{i\ 1}^{3} \frac{\tilde{F}_i \cdot \tilde{n}_i}{6} \quad \sum_{i\ 4}^{6} \frac{2\tilde{F}_i \cdot \tilde{n}_i}{3}$$

Compute ^{T} as

$$^T \quad \max_{i \in T} \|\tilde{n}_i\| \quad c_i q \cdot \max_{\text{edge}} |\text{edge}|$$

and for each degree of freedom of T, compute the Nodal Residual

$$\Phi_i^T \quad \frac{1}{6} \quad \Phi^T \quad ^T \sum_{j \in T} (U_i \quad U_j)$$

In the case of a vectorial problem, apply algorithm 1 page 93. In the scalar case, compute the first order Distribution Coe cients

$$_i^T \quad \frac{\Phi_i^T}{\Phi^T};$$

limit them

$$_i \quad \frac{^T_i}{\sum_{j \in T} {}^T_j}$$

and get the second order Nodal Residual

$$\Phi_i \quad _i \Phi^T.$$

Compute the Stabilization Term

$$D_i^T \quad |T|^2 \ ^T \sum_{q\ 1}^{k_q} \sum_{j \in T} U_j \quad \tilde{n}(x_q, q, r) \, \tilde{N}_i^k(x_q, q) \quad \tilde{n}(x_q, q, r) \, \tilde{N}_j^k(x_q, q)$$

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

$$(J_{Dissip}(q)_j \quad |T|^2 \ ^T \sum_{q\ 1}^{k_q} \sum_{j \in T} \tilde{n}(x_q, q, r) \, \tilde{N}_i^k(x_q, q) \quad \tilde{n}(x_q, q, r) \, \tilde{N}_j^k(x_q, q)$$

Gather the received signals

$$@P T; \quad RHS(i) \quad \Phi_i \quad D_i^T$$

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 RD Lax-Friedrichs scheme we are presenting in this thesis, is its easy generalization to any type of polyhedral element. Using the 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 Q^1 diffeomorphism $'$ transforming the reference element $\hat{Q} =]0;1[^2$ into Q, completely described by formula (3.11). The Q^k basis functions defined on the reference element are transported to Q thanks to $'$ and we obtain $(k+1)^2$ basis functions on Q that are polynomial of order k along the edges of Q and that verify:

$$\forall j \in P_Q, \quad Q_i^k(x_j) = \delta_{ij}.$$

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 $(k+1)^{th}$ 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 P^k case. The next paragraph is devoted to this aspect.

Figure 6.1: Global Residual computation in Q^1, Q^2 and 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 Q^k functions over the reference quadrangle composed with the Q^1 transformation ϕ. We recall that the Jacobian of this transformation is denoted by J. Moreover, the gradient of a Q^k function does not have to be Q^{k-1}. The only thing that is sure is that the gradient of the solution is a 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:

$$
\begin{aligned}
D_i^Q \quad & h^{Q+2} \sim \nabla w_i \sim \nabla u \, dx \\
& h^{Q+2} \sim \phi'(\hat{x}_q) \nabla \hat{w}_i(\hat{x}_q) \sim \phi'(\hat{x}_q) \nabla \hat{u}(\hat{x}_q) |J(\hat{x}_q)| \, d\hat{x} \\
& h^{Q+2} \sum_{q=1}^{(k+1)^2} \sum_{j=PQ} u_j \sim \phi'(\hat{x}_q) \nabla \hat{w}_i(\hat{x}_q) \sim \phi'(\hat{x}_q) \nabla \hat{w}_j(\hat{x}_q) |J(\hat{x}_q)| .
\end{aligned}
$$

If ψ denotes the inverse function of ϕ, one has

$$
w_i \quad \hat{w}_i \circ \psi .
$$

Then

$$
\nabla w_i \quad J^{-1} \cdot \nabla \hat{w}_i \circ \psi ,
$$

and

$$
D_i^Q \quad h^{Q+2} \sum_{q=1}^{(k+1)^2} \sum_{j=PQ} u_j \sim (x_q) J^{-1} \nabla \hat{w}_i(x_q) \sim (x_q) J^{-1} \nabla \hat{w}_j(x_q) |J(x_q)| \qquad (6.1)
$$

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 ($h \; P \; t0{:}0125; 0{:}01u$). 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 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 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 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).

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 $p1q$ and the value 0 will be maintained on boundaries $p2q$ and $p3q$.

We are going to impose a shape function on boundary $p1q$, with compact support in $r\ \frac{3}{4}; 0s$, and observe the advected function on the output boundary $r0; 1s$. We start by the regular function

$$\sin^8\ \frac{4\,x}{3}\quad ; \tag{6.5}$$

on boundary $p1q$. If value 0 is maintained on the other *inflow boundaries*, the exact solution is obviously

$$\begin{array}{ll} \sin^8\ \frac{4\,r}{3}\quad ; & \text{if } r\quad \sqrt{x^2\quad y^2}\quad \frac{3}{4} \\ 0; & \text{else} \end{array} \tag{6.6}$$

The value of the solution at the degrees of freedom of the output edge $x\quad r\ 0; 1s$ 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 $p1q$ is the characteristic function of interval $r\frac{1}{4}; \frac{3}{4}s$, $_{r\frac{1}{4};\frac{3}{4}s}p|x|q$ and the exact solution is given by

$$\begin{array}{lll} 1{:}0; & \text{if } \frac{1}{4}\quad r & \sqrt{x^2\quad y^2}\quad \frac{3}{4} \\ 0{:}0; & \text{else} & \end{array} \tag{6.7}$$

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 to annihilate the effects of this term in the vicinity of discontinuities. Here we have set 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 . 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 $r\ 10; 10s^2$. 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 p1q, p2q and p3q are the in ow boundaries.

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 $\rho(x)$.

Figure 6.8: Convergence of the lift coe cient. 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 gure corresponds to the third order ones.

Figure 6.10: Same60 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).

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 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.

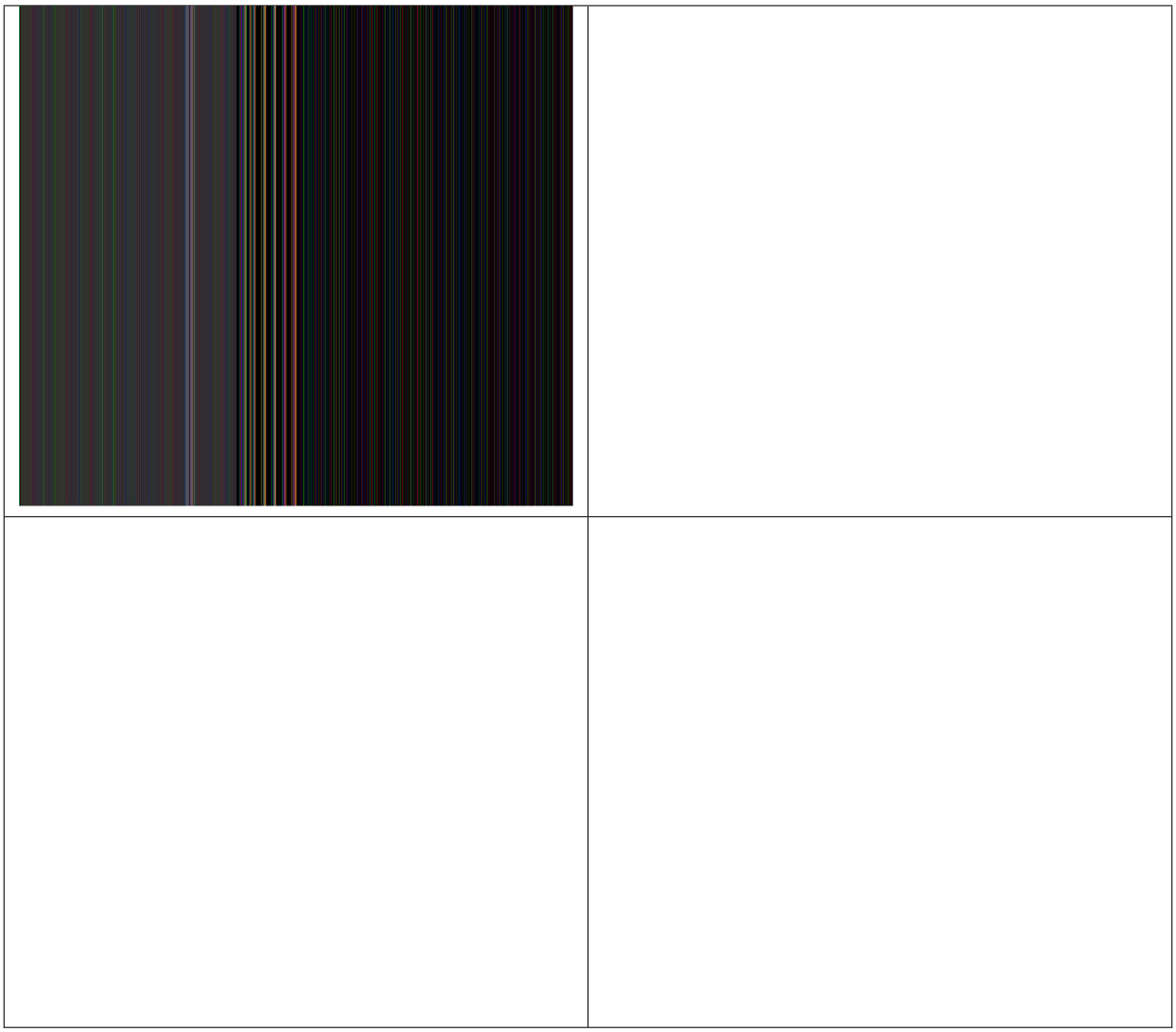


Figure 6.11: Entropy isolines and lift coefficient convergence for the sphere problem. Up-Left figure is the entropy contours for second order simulation, Up-Right is for third order simulation with linear representation of the boundaries. Down-Left is for third order simulation with isoparametrical 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

$$U \quad \begin{matrix} u \\ v \\ w \\ E \end{matrix} \quad : \tag{7.1}$$

The three dimensional unsteady Navier-Stokes equations read:

$$\frac{BU}{Bt} \quad \text{div} \quad \tilde{N}\acute{I} \quad pUq \quad p\,K_{ij}\,U_{;j}\,q_{;i} \quad \text{div} \quad K:r\acute{y}\acute{N}U \quad : \tag{7.2}$$

where, using $_i$ to denote the i^{th} column of the $3 \quad 3$ identity matrix,

$$\tilde{N}\acute{I} \quad p\,F_1; F_2; F_3q; \quad F_i \quad \begin{matrix} u_i \\ u_i u \quad p_{\ i} \\ pE \quad pqu_i \end{matrix} \quad ; i \quad 1:::3$$

is the *advection flux* and K is a $d \quad d$ diffusive matrix of $m \quad m$ ($m \quad d \quad 2$) matrices that are detailed in Appendix A. In Appendix B we have also reported the Jacobians of the advective flux A $\frac{BF_1}{BU}$, B $\frac{BF_2}{BU}$ and C $\frac{BF_3}{BU}$. The diagonalization of the 3D advection speed in any direction $\tilde{n}$ 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 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 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 $\mathcal{O}(n_{edges})$

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 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

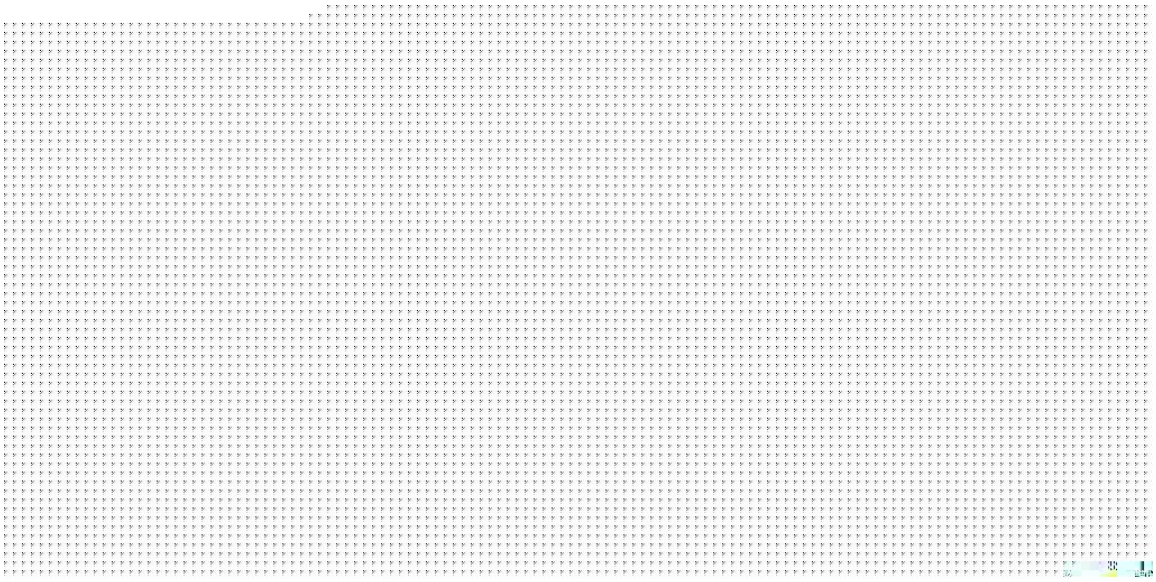


Figure 7.1: A example of a domain decomposition on 16 processors for a subsonic NACA012 mesh.

Figure 7.2: Detail around the stagnation point of the upper gure.

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 ne P^2 NACA012 simulation.

Figure 7.7: Numbering convention for P^1 and P^2 tetrahedra. When splitting the tetrahedron into sub-tetrahedra, the inside rhombohedron is split by its 7 9 diagonal.

$\hat{F}_h$ being a P^k function, this last integral is just a linear combination of the uxes on the DoFs sharing the face, the coe cients being the integral of the 3D Lagrangian basis function over the considered faces. The global residual is computed in practice as:

In P^1,

$$T \quad \sum_{i=1}^{4} \hat{F}_i \cdot n_i \frac{}{3}:$$

In P^2,

$$T \quad \sum_{i=5}^{10} \hat{F}_i \cdot n_i \frac{}{3}:$$

One can notice that in 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 rst 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 de ned uniquely.

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 first order matrices.

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.

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 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: Pro le of pressure around the wing at$z$ 0.