Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Présentée et soutenue par :
Steven Diot

Le jeudi 30 août 2012

Titre :
La méthode MOOD
--- Multi-dimensional Optimal Order Detection ---
la première approche a posteriori aux méthodes Volumes Finis d'ordre très élevé

École doctorale et discipline ou spécialité :
ED MITT : Domaine Mathématiques : Mathématiques appliquées

Unité de recherche :
Institut de Mathématiques de Toulouse

Directeur(s) de Thèse :

Stéphane Clain	Professeur, Université de Toulouse
Raphaël Loubère	Chargé de recherche CNRS, Université de Toulouse

Rapporteurs :

Rémi Abgrall	Professeur, Université de Bordeaux
Michael Dumbser	Professeur assistant, Università degli studi di Trento

Autre(s) membre(s) du jury :

Bruno Després	Professeur, Université Paris 6
Florian De Vuyst	Professeur, ENS Cachan
Pierre-Henri Maire	Ingénieur de recherche, CEA Cesta
Philippe Villedieu	Maître de recherche, ONERA Toulouse

On n'a pas le temps d'en perdre...

Remerciements

Je ne suis pas de nature expansive et mes remerciements se limiteront aux gens qui ont été et sont toujours importants dans mon travail de recherche et dans ma vie.

Il est évident que les deux personnes que je dois très sincèrement remercier sont Stéphane et Raphaël. Ils m'ont tout permis pendant ces trois années : de la découverte d'une recherche passionnante à la découverte de plusieurs pays, des journées de travail acharné à des soirées plus relaxantes... Je crois qu'on a été sur la même longueur d'onde du début à la fin. Un grand merci, j'espère que nos collaborations professionnelles et amicales vont continuer longtemps.

Un travail de thèse ne vaut pas grand chose sans l'expertise de personnes extérieures. Je veux donc remercier avec force Rémi Abgrall et Michael Dumbser qui m'ont fait l'honneur d'être rapporteurs de ma thèse. Leur validation scientifique de mon travail est d'autant plus pertinente que leurs travaux ont été des références pour moi. Il en va de même pour Bruno Després, Florian De Vuyst, Pierre-Henri Maire et Philippe Villedieu qui viennent renforcer la valeur scientifique de cette thèse chacun avec sa spécificité en ayant accepté d'être examinateurs. Un grand merci.

La science n'est rien sans les discussions parfois enflammées entre chercheurs et sans les rencontres internationales qui rendent ce métier extrêmement agréable. Je pense bien sûr à Bichon avec qui j'ai partagé beaucoup scientifiquement et encore plus politiquement, ou encore à PH et Dima. Je pense aussi à toutes ces rencontres : Eleonora, François, Pavel, Milan, Jan, Richard, Madi, Eva, Jacob, Gaspar, Rui, Paul, Mayu... C'est là un des grands bonheurs d'être chercheur.

Ces trois ans ont été éprouvants mais j'ai pu compter sur mes amis malgré le peu de temps que j'avais à leur consacrer. Vous le savez déjà mais merci à vous : Brice, Carine, Flo, Gilles, Noémie, Ju, Laura, Niko, Jenny, Nadine, Laure, Alexis...

Les derniers mots vont comme toujours à ceux sans qui on ne serait jamais arrivés jusqu'en thèse, la famille. Merci à John, Kathleen, et Sabrina de m'avoir soutenu et à Alan de m'avoir montré la voie ! Enfin merci à toi, Maman, qui m'a toujours poussé à réussir dès le début et à faire des études, tu peux être sûre que ça aura été utile !

Enfin il y en a une seule qui m'a supporté au jour le jour. Merci ma Poulette, en trois ans je t'ai délaissée, énervée, engueulée, poussée dans la voie de la recherche mais surtout aimée. Continuons à partager.

Table des matières

Introduction

Version française

La simulation numérique est devenue un outil stratégique pour la recherche scientifique et technologique au même titre que la théorie et l'expérience. Elle représente aussi un apport essentiel dans le monde industriel où elle est synonyme de réduction des coûts de conception. Une des raisons majeures est la forte croissance des moyens informatiques à coûts fixes qui permet aujourd'hui d'avoir une machine de calcul personnelle pour quelques milliers d'euros là où un super calculateur de puissance équivalente coûtaient plusieurs millions d'euros à l'achat et à l'entretien quelques décennies en arrière. De ce fait, les méthodes numériques se démocratisent et leur conception doit allier simplicité, flexibilité et efficacité de manière à les rendre accessibles à des non spécialistes.

Ces trois objectifs représentent les principales motivations de ce travail de thèse. En effet, nous proposons de développer un nouveau type de méthodes Volumes Finis d'ordre très élevé pour la simulation en dynamique des gaz compressibles non-visqueux régie par les équations d'Euler. Il existe déjà de nombreuses classes de méthodes numériques telles que les éléments finis [47, 44, 43], Galerkin discontinu[10, 24, 63] ou encore les schémas distribuant le résidu [3, 2] mais nous avons choisi la méthode Volumes Finis pour deux raisons principales. La première est sa simplicité qui est en adéquation avec nos objectifs et explique sa grand diffusion dans le monde du calcul scientifique et industriel. La deuxième est sa propriété de conservation des quantités physiques du système (masse, quantité de mouvement et énergie totale) qui se révèle cruciale dans la simulation de phénomènes physiques que nous considérons.

Dans la méthode Volumes Finis, qui est la généralisation du schéma de Godunov de 1959 [34], les inconnues du problème discret sont les valeurs moyennes par maille de la solution. Sur chaque cellule, l'évolution en temps de l'approximation numérique de la valeur moyenne s'obtient comme la somme des flux traversant les faces de la cellule. Plus précisément, le flux au travers de chaque face est approché (*i.e.* flux numérique) de telle sorte que ce qui rentre dans une cellule sort de l'autre impliquant automatiquement la propriété de conservation. De plus pour une même classe d'équation (lois de conservation par exemple), il est aisé de passer d'un problème à un autre par un simple changement de flux numérique en adéquation avec la flexibilité recherchée.

Cependant la méthode originelle de Volumes Finis n'est que d'ordre un et génère une grande quantité de diffusion numérique qui dégrade la qualité de la solution et empêche de capter toute

la finesse de la physique sous-jacente (*e.g.* discontinuités de contact). Ce défaut motive notre interêt pour les méthodes Volumes Finis d'ordre (très) élevé qui utilisent une reconstruction locale plus pertinente de la solution à partir des valeurs moyennes pour obtenir une simulation plus fine. Nous rappelons que la notion d'ordre de précision fait référence à la capacité à réduire l'erreur sur la solution lorsque le maillage est raffiné : une méthode d'ordre un divise par 2 l'erreur commise sur une solution lisse quand la taille caractéristique du maillage est divisée par deux, alors que pour une méthode d'ordre quatre, l'erreur est divisée par 2^4. En conséquence, une méthode d'ordre élevé permet d'améliorer considérablement la qualité de la solution à maillage donné, et donc de fournir une solution de qualité fixée en utilisant moins de ressources. On est donc en adéquation avec l'objectif d'efficacité.

Toute la difficulté de ce travail provient du type de physique que l'on souhaite simuler. En effet, une caractéristique majeure de la dynamique des gaz compressibles est la création d'ondes de choc et de discontinuités de contact qui rendent la solution localement singulière. L'emploi du schéma d'ordre un permet de traiter les discontinuités sans difficultés, cependant l'approximation à l'ordre élevé de telles singularités génère des oscillations parasites connues sous le nom de phénomène de Gibbs [12]. Celles-ci altèrent la qualité de la solution finale, et peuvent aller jusqu'à créer des situations non physiques qui peuvent interrompre la simulation. En conséquence la méthode numérique utilisée doit être capable d'approcher les parties lisses de la solution à l'ordre élevé tout en dégénérant vers l'ordre un sur les zones non lisses pour éviter toute oscillation parasite. Cette capacité est obtenue par la *procédure de limitation* et c'est dans sa conception que réside toute la difficulté.

On peut distinguer deux grandes familles de méthode Volumes Finis d'ordre élevé : les méthodes d'ordre deux et celles d'ordre supérieur. Les premières sont pour la plupart basées sur une reconstruction linéaire de la solution sur chaque cellule permettant d'atteindre l'ordre deux en précision et sur une limitation qui prévient les phénomènes parasites. La plus populaire est la méthode MUSCL [87, 75, 42] dont la limitation consiste à réduire la pente de la reconstruction linéaire pour prévenir les phénomènes parasites. Elle se caractérise par une mise en œuvre simple et efficace qui fait d'elle la méthode Volumes Finis d'ordre deux la plus répandue. Néanmoins l'ordre deux réduit fortement la capacité à capter des structures fines et des méthodes d'ordre plus elevé sont apparues dans les années 90.

Parmi ces dernières, la méthode WENO [45, 33, 77, 31, 82] (extension de la méthode ENO [40, 1]) a su s'imposer dans la communauté scientifique, en particulier dans la dynamique des fluides. Le fondement de toute méthode d'ordre très élevé s'appuie sur une reconstruction polynomiale locale (de degré aussi elevé que nécessaire) de la solution. Cependant les principes de limitation de la méthode MUSCL n'étant plus valides pour l'ordre supérieur, la méthode WENO utilise plusieurs reconstructions polynomiales par cellule et les combine pour obtenir une représentation de la solution qui est Essentiellement Non-Oscillante (ENO) réduisant ainsi le phénomène de Gibbs. De nombreuses applications en domaines multidimensionnels ont été publiées [90, 14, 60, 6], ce qui en fait une référence. Malheureusement la recontruction polynomiale est très coûteuse dans le cas de maillages non structurés (en particulier en 3D) et la plupart des applications physiques complexes requièrent de tels maillages. Par conséquent le coût de la méthode WENO est important de par la nécessité d'avoir plusieurs reconstructions d'ordre

élevé de la solution dans chaque cellule et handicape sérieusement sa popularisation. Pour autant, elle était jusqu'à présent la seule méthode Volumes Finis d'ordre très elevé réellement opérationnelle.

Un point commun à ces deux types de méthodes, et plus généralement aux procédures de limitation existantes, est leur traitement *a priori* des phénomènes parasites. En d'autres termes, dans toutes ces méthodes, la procédure de limitation agit sur les éléments précédents la mise à jour de la solution numérique en temps. Cette approche implique que l'on doit prévoir la validité de la solution même dans le pire scenario. Cependant les parasites numériques apparaissent au voisinage des singularités de la solution qui ne représentent la plupart du temps que quelques pourcents du domaine tandis que le schéma non limité produit des solutions pertinentes sur les zones régulières. Il est donc clair que ces méthodes *a priori* qui agissent sans discernement sur toutes les mailles effectuent un travail généralement inutile dans le sens où le schéma non limité aurait fonctionné directement. Ceci est particulièrement clair dans la méthode WENO pour laquelle une seule reconstruction polynomiale pourrait être utilisée la majorité du temps.

L'ensemble de ces constats nous a conduit à proposer un nouveau type de méthodes Volumes Finis d'ordre très élevé dont les deux paradigmes de base sont : une seule reconstruction polynomiale par cellule et un traitement *a posteriori* des problèmes parasites. Nous avons nommé cette méthode MOOD pour Multidimensional Optimal Order Detection, ce qui pourrait se traduire par Detection Multidimensionnelle de l'Ordre Optimal. Elle est conçue de manière très différente des méthodes déjà existantes : seuls des polynômes non limités sont utilisés et les problèmes sont traités *a posteriori* en recalculant l'évolution en temps des cellules problématiques après avoir localement réduit l'ordre du schéma utilisé. Ce concept s'appuie sur l'assurance que dans le pire des cas le schéma d'ordre un sera utilisé et fournira une solution valide. Ce principe permet donc d'éviter à la fois un traitement de toutes les cellules lorsque ce n'est pas nécessaire et l'utilisation de plusieurs reconstructions polynomiales par cellule. On peut donc espérer un gain en temps de calcul et en stockage mémoire important comparé aux méthodes actuelles (WENO par exemple). Le cœur de la méthode réside de ce fait dans les critères de détection utilisés pour définir si la solution calculée avec un schéma d'ordre élevé est acceptable ou non.

Nous montrons dans cette thèse, qu'avec un cadre et des outils bien définis, un tel concept *a posteriori* est non seulement viable mais surtout plus efficace que l'état de l'art. L'approche est validée par un nombre conséquent de tests numériques en dimension deux et trois sur l'équation de convection et le systéme d'Euler.

La présente étude est divisée en trois chapitres. Nous rappelons le cadre théorique des méthodes Volumes Finis multidimensionnelles ainsi que l'état de l'art des méthodes Volumes Finis d'ordre élevé dans le chapitre 1. Dans le chapitre 2, nous développons le nouveau cadre théorique associé au concept de limitation *a posteriori* pour ensuite définir clairement la méthode MOOD. Nous présentons l'application de cette méthode à l'équation scalaire de convection linéaire et au système d'équations d'Euler pour l'hydrodynamique. Le dernier chapitre réunie les trois publications majeures élaborées sur la méthode, dans le but de mieux comprendre la solution *a posteriori* que nous proposons au challenge des méthodes Volumes Finis d'ordre très élevé.

Enfin nous terminons notre étude par une discussion sur les perspectives importantes de la méthode MOOD.

English version

Numerical simulation has become a strategic tool for scientific and technological research as well as theory and experiments. It plays a key role in the industry where it is synonym of design cost reduction. One of the main reasons is the strong increase in the computational resources at a constant price. For instance a today's personal workstation costs thousands of euros whereas a supercomputer of same processing power cost millions of euros decades ago. As a matter of fact, numerical methods get generalized and their design has to combine simplicity, flexibility and efficiency to open them to non-specialists.

These three objectives represent the strategic points of this doctoral work. Indeed we propose to develop a novel type of very high-order Finite Volume methods to simulate the dynamics of non-viscous compressible gas ruled by the Euler equations.
There already exist numerous classes of numerical methods such as finite elements [47, 44, 43], discontinuous Galerkin [10, 24, 63] or residual distribution schemes [3, 2], but we have chosen the Finite Volume method for two reasons. First, the simplicity of the method fits our first objective and explains its large diffusion in scientific computation and industrial simulation. The second reason is the built-in conservativity property of physical quantities of the system (mass, momentum and total energy). It is of crucial importance for the simulations of physical phenomena we shall consider.

In the Finite Volume method, which is the generalization of the Godunov's scheme [34], the unknowns of the discrete problem are the solution mean values on cells. Over each cell, the time evolution of the numerical approximation of the mean value is obtained by the sum of the physical fluxes crossing the cell interfaces. More precisely, the flux through each face is approximated by a numerical flux so that what enters in a cell leaves the other one, so that the conservativity property is automatically ensured. Moreover for a same class of equations (*e.g.* conservation laws), Finite Volume methods can be easily adapted to different problems in the sense that only the numerical flux has to be substituted. Such a property suits to our flexibility objective to address a wide range of problems with few adjustments.

However the original Finite Volume method is first-order accurate and generates a large amount of numerical diffusion which implies accuracy discrepancies and prevent from capturing the details of the underlying physics (*e.g.* contact discontinuities). This drawback motivates our interest for (very) high-order Finite Volume methods which use a more relevant local reconstruction of the solution from mean values to obtain a finer simulation. We recall that the notion of accuracy order refers to the ability of the scheme to reduce the error on the solution when the mesh is refined: a first-order method divides the error by 2 when the characteristic length of the mesh is divided by two, while a fourth-order method divides it by 2^4. Consequently, a high-order method increases the quality of the solution on a fixed mesh, and so provides a fixed-quality solution with less computational resources. This fits to the efficiency objective.

The main difficulty of this study comes from the physical phenomena we intend to simulate. Indeed, one major characteristic of the compressible gas dynamics is the creation of shock waves and contact discontinuities yielding to singularities in the solution. First-order schemes treat these discontinuities without any difficulty, however the high-order approximation of such singularities generates spurious oscillations, the so-called Gibbs phenomenon [12]. These affect the final solution quality, and may create non-physical situations leading to a code crash. As a consequence, the numerical method must be able to approximate the smooth part of the solution with high-order while degenerating to a first-order scheme close to singularities to prevent any spurious oscillation. Such a mechanism is obtained through the *limitation procedure* and the difficulty lies in its design.

Two main families of high-order Finite Volume methods may be distinguished: the second-order methods and the higher-order ones. The former are essentially based on a linear reconstruction of the solution over each cell to reach second-order and on a slope limitation to prevent spurious phenomena. The most popular is the MUSCL method [87, 75, 42], the limitation of which basically consists in multiplying the slope of the linear reconstruction by a coefficient ranging between zero and one (limiter). The simplicity and efficiency of the MUSCL method led it to be the most widespread second-order Finite Volume method. Nevertheless the second-order strongly reduces its capacity to capture very fine structures and higher-order methods have been developed since the 90'.

Amongst these, the WENO method [45, 33, 77, 31, 82] (an extension of the ENO one [40, 1]) has become well-established in the scientific community, particularly for fluid dynamics. The basis of all higher-order methods is a local polynomial reconstruction (the degree of which is as high as necessary) of the solution. However since the limitation principles of the MUSCL method are not valid anymore, the WENO method uses several polynomial reconstructions per cell and combine them to obtain an Essentially Non-Oscillatory representation of the solution thus reducing the Gibbs phenomenon. The WENO method is a reference due to its wide range of applications [90, 14, 60, 6] in multidimensional domains. Unfortunately the polynomial reconstruction procedure is very costly on unstructured meshes (especially in 3D) and most of complex physics applications requires such meshes. Consequently, the cost of the WENO method is important since several high-order reconstructions per cell are needed and it seriously handicaps its popularization. Nevertheless it was thus far the only very high-order Finite Volume method effectively operational.

A common point to these two types of methods, and more generally to the existing limitation techniques, is their *a priori* treatment of spurious phenomena. In other words, the limitation procedure of these methods acts on the elements preceding the time update of the numerical solution. Such an approach implies that we must predict the validity of the solution and take into account the worst-case scenario. However the spurious phenomena appear in the vicinity of solution singularities which most of the time represent few percents of the domain while the unlimited scheme provides relevant solutions on regular zones. It is then clear that *a priori* methods, which act without discrimination on every cells, generally carry out a useless effort in the sense that the unlimited scheme would have performed well. This is particularly true in the case of the WENO method for which only one polynomial reconstruction could be mostly used.

All these assessments led us to propose a novel type of very high-order Finite Volume methods based on the two following paradigms: only one polynomial reconstruction per cell and an *a posteriori* treatment of spurious problems. We named it the MOOD method, standing for Multidimensional Optimal Order Detection. It is designed in a very different manner than the existing methods. Indeed the MOOD method only uses unlimited schemes and *a posteriori* treat problems by recomputing the time evolution of problematic cells after reducing the local scheme order. This concept relies on the claim that the first-order scheme is used in the worst-case scenario and provides a valid solution. This principle avoids a useless treatment of all cells when unnecessary, as well as the use of several polynomial reconstructions per cell. We may thus expect a computational cost and memory storage improvement compared to the existing methods (WENO as instance). The core of the method thus lies in the detection criteria used to determine if the solution computed with a higher-order scheme is acceptable or not.

We show in this thesis that with well defined tools and framework, such an *a posteriori* concept is not only viable but more efficient than the state-of-the-art very high-order Finite Volume methods. This approach is validated with numerous numerical tests for the two- and three-dimensional convection equations and hydrodynamics Euler system.

The present study is divided in three chapters. We recall the theoretical framework of the multidimensional Finite Volume methods and the state-of-the-art higher-order Finite Volume methods in chapter 1. In chapter 2, we develop the novel theoretical framework associated to the *a posteriori* limitation concept in order to clearly define the MOOD method. We present its application to the scalar convection equation and the hydrodynamics Euler system. In the last chapter, we merge the three main publications written about the MOOD method with the clear intention to better understand the *a posteriori* solution we propose to the challenge of very high-order Finite Volume methods. Finally we conclude our study with a discussion on the important perspectives of the MOOD method.

List of contributions

We give in this section the contributions produced in the framework of this thesis. The first paragraph gathers the publications in international peer reviewed journals, while the proceedings to international conferences are given in the second paragraph.

⋆ Journals

– S. Clain, S. Diot, R. Loubère, *A high-order finite volume method for systems of conservation laws – Multi-dimensional Optimal Order Detection (MOOD)*, J. Comput. Phys. 230 (2011) 4028–4050.

– S. Diot, S. Clain, R. Loubère, *Improved detection criteria for the Multi-dimensional Optimal Order Detection (MOOD) on unstructured meshes with very high-order polynomials*, Comput. Fluids 64 (2012) 43–63.

– S. Diot, R. Loubère, S. Clain, *The MOOD method in the three-dimensional case: Very-High-Order Finite Volume Method for Hyperbolic Systems*, submitted to Int. J. Numer. Meth. Fl. (2012).

⋆ Proceedings

– S. Clain, S. Diot, R. Loubère, *Multi-dimensional Optimal Order Detection (MOOD) A very high-order Finite Volume Scheme for conservation laws on unstructured meshes*, FVCA 6, Final Volume for Complex Applications, Prague, June 6–10 (2011).

– S. Diot, S. Clain, R. Loubère, *Three-dimensional preliminary results of the MOOD method: A Very High-Order Finite Volume method for Conservation Laws.* YIC2012, First ECCOMAS Young Investigators Conference, Aveiro, April 24–27 (2012).

– S. Clain, G. Machado, R. Pereira, R. Ralha, S. Diot, R. Loubère, *Very high-order finite volume method for one-dimensional convection diffusion problems*, 2nd International Conference on Mathematical Models for Engineering Science (MMES 11), Tenerife December 10–12 (2011).

Chapter 1

From first- to higher-order Finite Volumes schemes

Introduction

In 1959, S. K. Godunov proposed in [34] a conservative scheme to solve PDE's where he considered piecewise-constant approximation of the solution on cells, or Finite Volumes, as variables. The conservative property was intrinsically ensured by the variables update that was performed through the computation of fluxes at cells interfaces such that what enters in a cell goes out from the other one. The fluxes were obtained from the exact solution of a Riemann Problem at the interface, that is the exact solution of the problem at an interface separating two constant states. The idea has been widely extended, for instance to multidimensional domains and to different classes of problems, and is nowadays known as Finite Volume method. There are two main reasons to choose the Finite Volume method for solving PDE's. First, the method is simple: the scheme is identical whichever types of cells or space dimension we intend to use, since it only uses mean values of the solution and inter-cells fluxes computation, and so, leads to a simplistic and robust implementation for a simulation code. The second reason is the built-in conservativity property of the method in the sense that it preserves the physical quantities of the problem, that is of crucial importance to reproduce the physics when dealing with hydrodynamics for instance.

However the original Finite Volume method is only first-order accurate. This generates a large amount of numerical diffusion and so requires the use of very fine meshes to get accurate simulations. Actually, contrarily to Finite Elements methods for instance, there is no straightforward higher-order extension of Finite Volume and reconstructing a high-order representation of the solution from mean values is mandatory.

In section 1.1, we build the higher-order Finite Volume scheme that we consider in this thesis from the first-order one using the high-order polynomial reconstructions of the solution from mean values detailed in section 1.2. Then in section 1.3, we present the state-of-the-art high-order Finite Volume methods and remind the mandatory points that must be taken into account to effectively reach higher-order. Finally in the whole presentation, we consider problems governed by a system of conservation laws, and although it already corresponds to many physical problems, an extension to balanced laws could be easily developed to cover even more physics.

1.1 Very high-order Finite Volumes schemes

We first introduce the notation for chapters 1 and 2 and illustrate them in Figure 1.1. All denominations are based on the three-dimensional case, *e.g.* a face corresponds to an edge in 2D or a point in 1D. We assume that the computational domain Ω is a polyhedral bounded set of $\mathbb{R}^n$ with $n = 1, 2$ or 3 and denote by $\mathbf{x}$ any of its points. A mesh of Ω is given by a set of non-overlapping convex polyhedral cells, or Finite Volumes, $K_i, i \in \mathcal{E}_{el}$, where $\mathcal{E}_{el}$ is the cells index set. For the sake of simplicity, we only consider cells with coplanar faces and recall that cells with non coplanar faces could be treated by a decomposition into coplanar ones. For each cell K_i we denote its volume by $|K_i| = \int_{K_i} 1 \, d\mathbf{x}$ and its centroid by $\mathbf{c_i} = |K_i|^{-1} \int_{K_i} \mathbf{x} \, d\mathbf{x}$. We moreover define two index sets of cells linked to a cell K_i: the index set $\underline{\nu}(i)$ of all cells K_j sharing a face f_{ij} with K_i, *i.e.* $\underline{\nu}(i) = \left\{ j \in \mathcal{E}_{el} \backslash \{i\} | \overline{K_i} \cap \overline{K_j} = f_{ij} \right\}$ and the index set $\overline{\nu}(i)$ of cells sharing a geometrical element with K_i, *i.e.* $\overline{\nu}(i) = \left\{ j \in \mathcal{E}_{el} \backslash \{i\} | \overline{K_i} \cap \overline{K_j} \neq \emptyset \right\}$, see Figure 1.2 for illustrations in 2D and 3D. Finally for each face f_{ij} we denote by $\mathbf{n}_{ij}$ its unit normal vector going from K_i to K_j (that is unique for coplanar faces) and by $\left(q_{ij,r}, \xi_{ij,r} \right., r = 1, \ldots, R_{ij}$, the quadrature points and weights, *i.e.* the quadrature rule, where R_{ij} is the number of quadrature points on f_{ij}. We detail in appendix A, the quadrature rules we have used to reach up to 6^{th}-order.

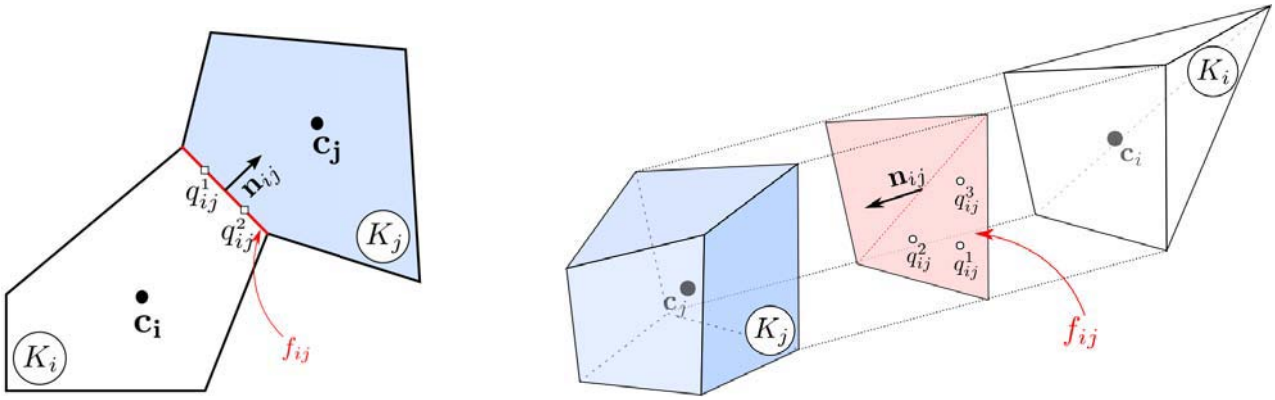


Figure 1.1: Notations in 2D (left) and in 3D (right).

We consider a generic conservation law

$$\partial_t U + \nabla \ F(U) = 0, \quad \forall \mathbf{x} \in \Omega, t > 0, \tag{1.1a}$$

$$U(\ , 0) = U_0, \ \forall \mathbf{x} \in \Omega, \tag{1.1b}$$

where $U = U(\mathbf{x}, t)$ is the vector of unknown functions, also referred to as conservative variables, t is the time, $F(U) = F(U(\mathbf{x}, t), \mathbf{x}, t)$ is the so-called physical flux function and $U_0 = U_0(\mathbf{x})$ stands for the initial condition. In the sequel, we omit to write the space and time dependence of F for the sake of clarity and we shall precise boundary conditions.

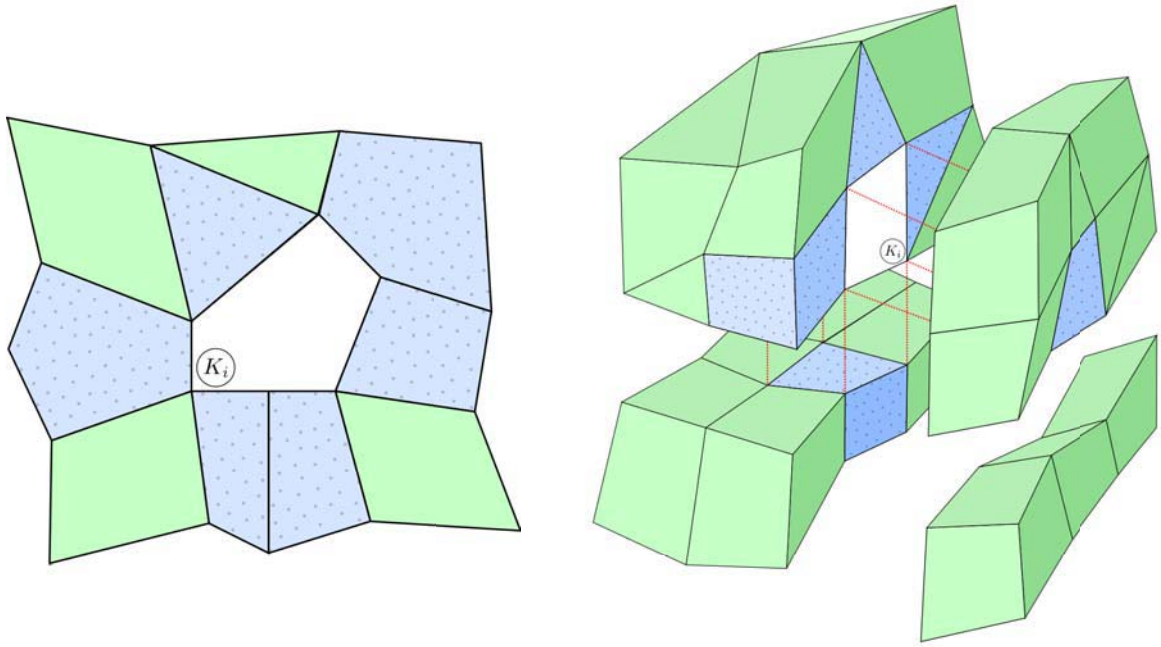


Figure 1.2: Illustrations of index sets in 2D and 3D: $\underline{\nu}(i)$ represents the blue cells with dots while $\overline{\nu}(i)$ represents every non white cells.

Let us recall the first-order Finite Volume Godunov's scheme for equation (1.1). We start with the first-order spatial discretization by computing the mean value of (1.1a) over a generic cell K_i

$$\partial_t \frac{1}{|K_i|}\int_{K_i} U(\mathbf{x},t)\ d\mathbf{x} + \frac{1}{|K_i|}\int_{K_i} \nabla \quad F(U(\mathbf{x},t))\ d\mathbf{x} = 0.$$

Using the divergence theorem along with $\partial\Omega = \cup_{i\in\underline{\nu}(i)} f_{ij}$, we obtain

$$\partial_t \frac{1}{|K_i|}\int_{K_i} U(\mathbf{x},t)\ d\mathbf{x} + \frac{1}{|K_i|}\sum_{j\in\underline{\nu}(i)}\int_{f_{ij}} F(U(\mathbf{x},t)) \quad \mathbf{n}_{ij}\ ds = 0, \tag{1.2}$$

where ds corresponds to the measure on face f_{ij}. Finally the spatial discretization of equation (1.2) is obtained by approximating the flux integral. To his end, we introduce the so-called numerical flux $\mathbb{F}(U_i(t), U_j(t), \mathbf{n}_{ij})$ which depends on mean values on both sides of f_{ij} and on the normal vector $\mathbf{n}_{ij}$. It is an approximation of $F(U(\mathbf{x},t)) \quad \mathbf{n}_{ij}$ computed from the solution of a Riemann Problem. More precisely, $\mathbb{F}(U_i(t), U_j(t), \mathbf{n}_{ij})$ is obtained from the exact or an approximated solution to the problem defined by equations (1.1a)–(1.1b) rotated in the normal direction $\mathbf{n}_{ij}$ at an interface separating the two constant states $U_i(t)$ and $U_j(t)$. A detailed description can be found in [78]. Therefore the spatial discretization writes

$$\partial_t U_i(t) + \sum_{j\in\underline{\nu}(i)} \frac{|f_{ij}|}{|K_i|}\mathbb{F}(U_i(t), U_j(t), \mathbf{n}_{ij}) = 0, \tag{1.3}$$

where notation $U_i(t) \approx |K_i|^{-1}\int_{K_i} U(\mathbf{x},t)\ d\mathbf{x}$ stands for an approximation of the time-dependent

mean value of U on K_i.

We recall that the numerical flux $\mathbb{F}$ has to fulfill the following properties to ensure the meaningfulness of the scheme. First, $\mathbb{F}$ must be continuous and Lipschitz with respect to the first and second arguments. Then to ensure the conservativity of the scheme, $\mathbb{F}$ must fulfill $\mathbb{F}(U, V, \mathbf{n}) = -\mathbb{F}(V, U, -\mathbf{n})$ for any constant states U and V and any normal vector $\mathbf{n}$. At last, the consistency of the scheme is ensured by asking $\mathbb{F}$ to fulfill $\mathbb{F}(U, U, \mathbf{n}) = F(U) \cdot \mathbf{n}$ for any constant state U and any normal vector $\mathbf{n}$.

Finally the explicit first-order Godunov's scheme for equation (1.1) is classically obtained by a forward Euler method for time discretization of equation (1.3)

$$U_i^{n+1} = U_i^n - \Delta t^n \sum_{j \in \underline{\nu}(i)} \frac{|f_{ij}|}{|K_i|} \mathbb{F}(U_i^n, U_j^n, \mathbf{n}_{ij}), \tag{1.4}$$

where $U_i^n \approx |K_i|^{-1} \int_{K_i} U(\mathbf{x}, t^n)\, d\mathbf{x}$ is an approximation of the mean value of U on cell K_i at time t^n and $\Delta t^n = t^{n+1} - t^n$ is the time step. Notice that we use the superscript ^{n} for the time step t^n and for the dimension of the domain $\Omega \in \mathbb{R}^n$, however no confusion can be made in the context. The treatment of the temporal discretization separately from the spatial one is a classical technique that permits the use of traditional methods for Ordinary Differential Equations (ODE) to solve equation (1.3) and is usually denominated as method of lines.

Although this scheme is easy to obtain and very robust to use, the major drawback of the scheme is the large amount of numerical diffusion which generates strong accuracy discrepancies. Following this claim, higher-order methods have been designed to reduce numerical diffusion while trying to keep simplicity and robustness as main objectives. Different strategies have been developed: a first trend for which the scheme is independent of the problem whereas for the other trend, the equations of the problem are explicitly used to reach higher-order. The former, so-called method of lines, consists in treating separately the spatial and temporal discretizations, whereas the latter, that is more complex and less widespread, aims at reaching higher-order in space and time in one step by mixing both discretizations.
We have chosen in this work to follow the first trend of higher-order Finite Volume methods to design a problem-independent scheme. This choice is made for the sake of flexibility in the sense that the physics of the problem only appears in the numerical flux, and so treating a different problem is essentially equivalent to considering a different numerical flux. Our higher-order schemes will thus rely on the method of lines for which both spatial and time discretizations have to be high-order. From equation (1.2), we remark that reaching a spatial high-order is equivalent to using high-order approximations of the flux integrals, which requires quadrature rules on faces and higher-order approximations of the flux at quadrature points. Moreover since solution U is only known through its mean values, we need a high-order representation of U on each cell to compute high-order flux approximations. To this end, a polynomial approximation of U on each cell is reconstructed from the neighboring mean values. This technique is detailed in section 1.2 as it is a very important and non trivial point.

In the sequel, the strategy consists in considering the first-order scheme of equation (1.4) as a building block and defining the high-order scheme by means of linear convex combinations of this building block so that the high-order update of an existing first-order code is simplified.

We consider on each face f_{ij} a quadrature rule $(\xi_{ij,r}, q_{ij,r})$ and high-order approximations $U^n_{ij,r}$, $U^n_{ji,r}$ of U at the quadrature points $q_{ij,r}$ and at time t^n respectively computed from K_i and K_j. The high-order spatial scheme then writes

$$U_i^{n+1} = U_i^n - \Delta t \sum_{j\in\underline{\nu}(i)} \frac{|f_{ij}|}{|K_i|} \sum_{r=1}^{R_{ij}} \xi_{ij,r}\, \mathbb{F}(U^n_{ij,r}, U^n_{ji,r}, \mathbf{n}_{ij}), \tag{1.5}$$

where we still use a forward Euler time discretization. We recall that $\sum_{r=1}^{R} \xi_{ij,r} = 1$ with $\xi_{ij,r} > 0$ for any face f_{ij}, consequently an important property follows from equation (1.5)

Proposition 1.1 *If higher-order approximations $U^n_{ij,r}$, $U^n_{ji,r}$, in equation (1.6) are replaced by first-order ones, U^n_i and U^n_j respectively, we recover the first-order Godunov scheme (1.4).*

In 1D, 2D and 3D purely tetrahedral meshes, all faces are of same type. Consequently R_{ij} and $\xi_{ij,r}$ are independent of i and j and equation (1.5) can be written as a convex combination of the first-order FV scheme

$$U_i^{n+1} = \sum_{r=1}^{R} \xi_r \left(U_i^n - \Delta t \sum_{j\in\underline{\nu}(i)} \frac{|f_{ij}|}{|K_i|} \mathbb{F}(U^n_{ij,r}, U^n_{ji,r}, \mathbf{n}_{ij}) \right), \tag{1.6}$$

When dealing with 3D polyhedral meshes, the quadrature rules may be different from a face to another and such a convex combination form is not directly obtained. However by triangular decomposition of polygonal faces, we may consider each cell with polygonal faces as a cell with more faces but only triangular ones so that equation (1.6) still holds.

Let us now turn to the time discretization. Since we use a method of lines, the order of accuracy in time should be equal to the spatial one to preserve the formal order of accuracy of the complete scheme. We will use the 3^{rd}-order Runge–Kutta 3 Total Variation Diminishing (RK3 TVD) method (see [45]). Let us first rewrite equation (1.5) under the following operator form

$$U_h^{n+1} = U_h^n + \Delta t \mathcal{H}^R(U_h^n), \tag{1.7}$$

where notation U_h^n stands for $\sum_{i\in\mathcal{E}_{el}} U_i^n \mathbb{1}_{K_i}$. The RK3 TVD method then writes as a convex combination

$$U_h^{n+1} = \frac{U_h^n + 2U_h^{(3)}}{3} \quad \text{with} \quad \left\{ \begin{array}{rcll} U_h^{(1)} & = & U_h^n + \Delta t\, \mathcal{H}^R(U_h^n) & \text{at } t = t^n, \\ U_h^{(2)} & = & U_h^{(1)} + \Delta t\, \mathcal{H}^R(U_h^{(1)}) & \text{at } t = t^n + \Delta t, \\ U_h^{(3)} & = & \widehat{U}_h^{(2)} + \Delta t\, \mathcal{H}^R(\widehat{U}_h^{(2)}) & \text{at } t = t^n + \Delta t/2, \end{array} \right. \tag{1.8}$$

where $\widehat{U}_h^{(2)}$ is the convex combination $(3U_h^n + U_h^{(2)})/4$ and t is the current time for the update.

Notice that we write the RK3-TVD as a convex combination of three forward Euler time steps. This is important for our purpose since the properties fulfilled by the first-order Godunov scheme (1.4) are preserved. More specifically, the proposition 1.2 implies that each stage of the RK3-TVD is a first-order Godunov scheme (1.4) when first-order mean values are used in place of the higher-order approximations. Furthermore since only convex combinations are used in the RK3-TVD (1.8), the following proposition holds.

Proposition 1.2 *A solution provided by the RK3-TVD method (1.8) fulfills all the properties of the first-order Godunov scheme (1.4) that are preserved by convex combinations (DMP, bounds on variables, etc.), if higher-order approximations of the three stages are replaced by the first-order mean values.*

Although this method introduces a 3^{rd}-order error in time, $\mathcal{O}(\Delta t^3)$, it is possible to reach higher-order by setting $\Delta t = \Delta \mathbf{x}^{r/3}$, where $\Delta \mathbf{x}$ is a characteristic spatial length and r is the target order of accuracy. However there exist higher-order time discretizations but either they use more steps than the order (the Runge–Kutta Strong Stability Preserving methods) and significantly increase the computational cost, or they explicitly use equations of the problem (*e.g.* the ADER method, see further) and are less flexible.

Unfortunately accuracy and robustness are hardly compatible and high-order schemes are usually plagued with robustness problems. In particular, they generate spurious oscillations close to steep gradients and may even produce unphysical solutions (*e.g.* negative density or pressure for a gas). In section 1.3, we review the existing methods used to prevent these stability problems. In order to emphasize the characteristics of each method, we propose a simple sketch based on Figure 1.3 in which we gather the relevant steps (for our purpose) of the unlimited scheme defined by equations (1.5) and (1.8). But first and foremost, we present in next section the crucial point to reach higher-order: the polynomial reconstruction.

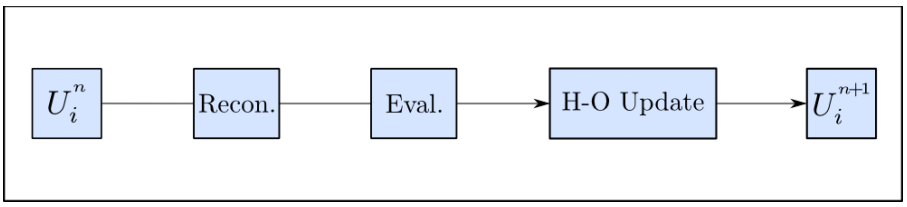


Figure 1.3: Simplified flowchart to compute solution U_i^{n+1} from U_i^n by means of a very high-order unlimited spatial discretization. Only the main steps are sketched: the polynomial reconstruction (Recon.), the evaluation of high-order approximations at quadrature points (Eval.) and the high-order update of the solution (H-O Update).

1.2 Arbitrary degree polynomial reconstruction

Contrary to other type of methods (*e.g.* Finite Elements or Discontinuous Galerkin methods) based on local decomposition of the solution on a basis that can easily provide a high-order representation of solution, the Finite Volume method only considers mean values of the solution.

In consequence, reaching higher-order in a Finite Volume context necessarily implies that we must reconstruct a high-order representation of the underlying function from its mean values. Classically (but not necessarily, *e.g.* [4, 5]) such a representation is obtained by means of a polynomial approximation (see [9, 57, 36] for instance), and the high-order nature is straightforwardly ensured through a Taylor expansion.

In this section, we only focus on the multidimensional case since the 1D case, for which building a polynomial approximation is analytically tractable, is not representative of arising problems. In practice the polynomial reconstruction is performed on each variable independently, so we hereafter detail such a process in the case of a scalar function u.

In the sequel, we consider a generic cell K_i where we seek to reconstruct a polynomial $\widetilde{u}_i(\mathbf{x};\mathsf{d})$ of arbitrary degree $\mathsf{d}\geq 1$ for $\mathbf{x}\in K_i\subset\Omega\subset\mathbb{R}^n (n=1,2 \text{ or } 3)$, of the form

$$\widetilde{u}_i(\mathbf{x};\mathsf{d}) = u_i + \sum_{1\leq|\boldsymbol{\alpha}|\leq\mathsf{d}} \mathcal{R}_i^{\boldsymbol{\alpha}}\left((\mathbf{x}-\mathbf{c}_{K_i})^{\boldsymbol{\alpha}} - \frac{1}{|K_i|}\int_{K_i}(\mathbf{x}-\mathbf{c}_{K_i})\,d\mathbf{x}\right), \tag{1.9}$$

where c_{K_i} is a point linked to reference cell K_i (*e.g.* the cell centroid), $\boldsymbol{\alpha}=(\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_n)\in\mathbb{N}^n$ is a multiindex with $|\boldsymbol{\alpha}|=\boldsymbol{\alpha}_1+\text{—}+\boldsymbol{\alpha}_n$ and $\mathcal{R}_i^{\boldsymbol{\alpha}}\in\mathbb{R}$ are called (unknown) polynomial coefficients.

As stated, the reconstruction of such a polynomial uses the mean values of the solution since they are the only available information. Moreover since $\widetilde{u}_i$ must locally represent the solution on K_i, we only consider mean values on a set of cells in the neighborhood of K_i. Such a set is called *(reconstruction) stencil* for degree d and denoted by $\mathcal{S}_i^{\mathsf{d}} = \left\{\mathcal{S}_i^{\mathsf{d}}(1),\ldots,\mathcal{S}_i^{\mathsf{d}}(N_{\mathcal{S}_i}^{\mathsf{d}})\right\}$ where $\mathcal{S}_i^{\mathsf{d}}(r)$ corresponds to the index of the r^{th} neighboring cell and $N_{\mathcal{S}_i}^{\mathsf{d}}$ is the number of neighbors in the stencil.

In the rest of the thesis, the expression *reconstruction problem* is employed to characterize the process of determining the unknown polynomial coefficients $\mathcal{R}_i^{\boldsymbol{\alpha}}$ and we equally refer to the polynomial $\widetilde{u}_i$ itself as *polynomial representation*, *polynomial reconstruction* or *reconstructed polynomial* of u on K_i.

The number of unknowns $\mathcal{R}_i^{\boldsymbol{\alpha}}$ in the reconstruction problem, denoted $\mathcal{N}(\mathsf{d})$, depends on the space dimension n, and on the polynomial degree d, through the formula

$$\mathcal{N}(\mathsf{d}) = \frac{\prod_{i=1}^{n}(\mathsf{d}+i)}{n!} - 1.$$

At least $\mathcal{N}(\mathsf{d})$ equations are necessary to fully define the reconstructed polynomial on K_i. Those are classically obtained by asking for the conservation property to be fulfilled for all cells of the stencil, that is to say $|K_j|^{-1}\int_{K_j}\widetilde{u}_i(\mathbf{x})\,d\mathbf{x} = u_j,\ \forall j\in\mathcal{S}_i^{\mathsf{d}}$. However it is well known that in the general case (for $n=2,3$), using same number of equations as the number of unknowns ($N_{\mathcal{S}_i}^{\mathsf{d}} = \mathcal{N}(\mathsf{d})$) does not necessarily provide a solution to the reconstruction problem. Therefore more neighbors must be selected ($N_{\mathcal{S}_i}^{\mathsf{d}} > \mathcal{N}(\mathsf{d})$) leading to a least-squares problem (see [9] for instance).

Remark: The form of $\widetilde{u}_i$ in (1.9) is our choice and may be different. Actually it is usual to use c_{K_i} to localize the polynomial with respect to the reference cell, but subtracting the mean values of monomials is less common. The purpose of this last term is to ensure that the conservation property $|K_i|^{-1}\int_{K_i}\widetilde{u}_i(\mathbf{x};\mathsf{d})\,d\mathbf{x} = u_i$ is always fulfilled. This is not the case of all polynomial reconstruction method, see [9, 33] for instance. It results that we only have $\mathcal{N}(\mathsf{d})$ unknowns instead of $\mathcal{N}(\mathsf{d})+1$ which is the number of monomials of a polynomial of degree d in $\mathbb{R}^n$.

The next paragraph is dedicated to the setting of the least-squares reconstruction problem, then we propose a technique to solve and store it. Finally the choice of the reconstruction stencil is discussed in the last paragraph.

⋆ Setting the least-squares problem

We consider a stencil $\mathcal{S}_i^{\mathsf{d}}$ with cardinal $N_{\mathcal{S}_i}^{\mathsf{d}} > \mathcal{N}(\mathsf{d})$. As mentioned, we seek to minimize in a least-squares sense the difference between the mean value of the reconstructed polynomial and the solution mean value on the cells of $\mathcal{S}_i^{\mathsf{d}}$; this is equivalent to minimizing the following functional

$$E(\mathcal{R}_i^{\boldsymbol{\alpha}}) = \frac{1}{2}\sum_{j\in\mathcal{S}_i^{\mathsf{d}}}\left(\frac{1}{|K_j|}\int_{K_j}\widetilde{u}_i(\mathbf{x};\mathsf{d})\,d\mathbf{x} - u_j\right)^2.$$

However solving this minimization at each time step would be very costly. In consequence, the reconstruction problem is cast under a matrix form such that the solution of this matrix problem minimizes the functional E, see [76] for details.

We now define the matrix form of the reconstruction problem. Let us first write the mean value of $\widetilde{u}_i$ on a given cell K_j

$$\frac{1}{|K_j|}\int_{K_j}\widetilde{u}_i(\mathbf{x};\mathsf{d})\,d\mathbf{x} = u_i + \sum_{1\leq|\boldsymbol{\alpha}|\leq\mathsf{d}}\mathcal{R}_i^{\boldsymbol{\alpha}}\mathsf{X}_{i,j}^{\boldsymbol{\alpha}},$$

where we set

$$\mathsf{X}_{i,j}^{\boldsymbol{\alpha}} = \left(\frac{1}{|K_j|}\int_{K_j}(\mathbf{x}-\mathbf{c}_{K_i})^{\boldsymbol{\alpha}}\,d\mathbf{x} - \frac{1}{|K_i|}\int_{K_i}(\mathbf{x}-\mathbf{c}_{K_i})^{\boldsymbol{\alpha}}\,d\mathbf{x}\right).$$

Then writing the equality between the mean value of $\widetilde{u}_i$ and the solution u_j, we obtain

$$\sum_{1\leq|\boldsymbol{\alpha}|\leq\mathsf{d}}\mathcal{R}_i^{\boldsymbol{\alpha}}\mathsf{X}_{i,j}^{\boldsymbol{\alpha}} = u_j - u_i. \tag{1.10}$$

Finally we write the matrix system obtained when equation (1.10) is considered for all cells of

the stencil $\mathcal{S}_i^{\mathsf{d}}$

$$\begin{pmatrix} \mathsf{X}_{i,\mathcal{S}_i(1)}^{(1,0,0)} & \mathsf{X}_{i,\mathcal{S}_i(1)}^{(0,1,0)} & \mathsf{X}_{i,\mathcal{S}_i(1)}^{(0,0,1)} & \cdots & \mathsf{X}_{i,\mathcal{S}_i(1)}^{(0,0,d)} \\ \mathsf{X}_{i,\mathcal{S}_i(2)}^{(1,0,0)} & \mathsf{X}_{i,\mathcal{S}_i(2)}^{(0,1,0)} & \mathsf{X}_{i,\mathcal{S}_i(2)}^{(0,0,1)} & \cdots & \mathsf{X}_{i,\mathcal{S}_i(2)}^{(0,0,d)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathsf{X}_{i,\mathcal{S}_i(N_{\mathcal{S}_i})}^{(1,0,0)} & \mathsf{X}_{i,\mathcal{S}_i(N_{\mathcal{S}_i})}^{(0,1,0)} & \mathsf{X}_{i,\mathcal{S}_i(N_{\mathcal{S}_i})}^{(0,0,1)} & \cdots & \mathsf{X}_{i,\mathcal{S}_i(N_{\mathcal{S}_i})}^{(0,0,d)} \end{pmatrix} \begin{pmatrix} \mathcal{R}_i^{(1,0,0)} \\ \mathcal{R}_i^{(0,1,0)} \\ \mathcal{R}_i^{(0,0,1)} \\ \vdots \\ \mathcal{R}_i^{(0,0,d)} \end{pmatrix} = \begin{pmatrix} u_{\mathcal{S}_i(1)} - u_i \\ u_{\mathcal{S}_i(2)} - u_i \\ \vdots \\ u_{\mathcal{S}_i(N_{\mathcal{S}_i})} - u_i \end{pmatrix}, \tag{1.11}$$

where we have dropped the superscript d for the sake of clarity. Let us denote the matrix by X_i, the unknown coefficients vector by $\mathcal{R}_i$ and the vector of differences between mean values on neighboring cells and on K_i by $U_{\mathcal{S}_i}$. Hence the above linear system writes

$$\mathsf{X}_i \mathcal{R}_i = U_{\mathcal{S}_i}. \tag{1.12}$$

This overdetermined linear system is the basis of the polynomial reconstruction and contrary to the functional form, the information on geometry and on the solution are decoupled. Indeed all coefficients of matrix X_i are derived from the geometry of the mesh while information on the solution are contained in the right-hand vector $U_{\mathcal{S}_i}$. This is of crucial importance to ensure an efficient method for which an important part of the problem is preprocessed since the mesh information remains identical in time. The next paragraph is dedicated to solving this matrix problem.

⋆ Solving the least-squares problem

To ensure existence and uniqueness of a solution, we need to consider that the rank of matrix X_i is maximal, or equivalently that its columns are linearly independent. In that case, the matrix $\mathsf{X}_i^T \mathsf{X}_i$ is invertible (where ^{T} stands for the transpose) and equation (1.12) yields

$$\mathcal{R}_i = (\mathsf{X}_i^T \mathsf{X}_i)^{-1} \mathsf{X}_i^T \, U_{\mathcal{S}_i} =: \mathsf{X}_i^\dagger \, U_{\mathcal{S}_i}, \tag{1.13}$$

where the matrix $\mathsf{X}_i^\dagger := (\mathsf{X}_i^T \mathsf{X}_i)^{-1} \mathsf{X}_i^T$ corresponds to the Moore-Penrose pseudoinverse of X_i in the particular case of maximal rank. More details about the pseudoinverse can be found in [76]. This expression considerably simplifies the reconstruction procedure in the simulation since the polynomial coefficients $\mathcal{R}_i$ are obtained by a matrix-vector product of $\mathsf{X}_i^\dagger$ with $U_{\mathcal{S}_i}$. Furthermore since X_i only depends on geometry, it is also true for its pseudoinverse and consequently $\mathsf{X}_i^\dagger$ is computed in a preprocessing stage and stored in memory.

Several methods are available in the literature to decompose X_i and compute the pseudoinverse. We expose here two possibilities, the QR and the SVD decompositions.

The first method is the QR decomposition, with Q an orthogonal matrix and R an upper triangular matrix, using Householder transformations (see [76] for instance). Once such decomposition is available, the pseudoinverse is simply given by

$$\mathsf{X}_i^\dagger = ((QR)^T (QR))^{-1} \mathsf{X}_i^T = (R^T R)^{-1} \mathsf{X}_i^T,$$

where the inverse of R^TR is obtained by performing a forward- and a backward-substitution consecutively.

The second method is based on the Singular Value Decomposition (SVD) of X_i under the form UDV^T, where U and V are orthogonal and D contains the singular values of X_i on the diagonal and is zero elsewhere. We recall that the squares of the singular values are the eigenvalues of the matrix $\mathsf{X}_i^T\mathsf{X}_i$. Due to the assumed maximal rank of X_i, there is no null singular value in D (D is invertible) and the pseudoinverse of UDV^T writes

$$\mathsf{X}_i^\dagger = (UDV^T)^{-1} = (V^T)^{-1}D^{-1}U^{-1} = VD^{-1}U^T.$$

We have adopted the QR decomposition in our code, however it seems that the SVD may be more suitable since the conditioning number of $\mathsf{X}_i^\dagger$ is only impacted by the one of D (or equivalently of X_i), whereas for the QR both R and R^T are inverted.

To conclude this description of the reconstruction problem, we draw few remarks about what may affect the conditioning number of matrix X_i, and so the stability of the reconstruction.
The first element is the form (1.9) of the polynomial $\widetilde{u}_i$ that we choose to localize around K_i by considering monomials under the form $(\mathbf{x} - \mathbf{c}_{K_i})$ in order to balance the rows of matrix X_i and thus obtain analogous matrices from a cell to another. However this is not enough to cure all problems. For instance, in the above description the conditioning number is dependent of the cell size and grows when the mesh is refined, see [1, 33]. Therefore we adopt the trick of O. Friedrichs [33] to make the conditioning number independent of cell size. Remark that a barycentric technique has been proposed in [1] and a projection technique on a reference element has been proposed in [32]. Nevertheless both techniques may not avoid very large conditioning number problems when high aspect ratios are present between neighbors of the reconstruction stencil.
Another possibility consists in using a preconditioning technique. For instance a weight may be associated to each neighbor according to its distance to the reference cell, such that a closer neighbor has a bigger weight. As a result, the reconstruction may be more accurate but we have observed that it may also be less stable, particularly in 3D.
Finally the reconstruction stencil may be one the most important element that affects the conditioning number, so we dedicate next paragraph to describe our algorithm to choose the stencils.

⋆ Choosing the reconstruction stencil

Although being an important step, the choice of neighbors for the reconstruction problem is still an open problem in 2D and 3D, especially when neighboring cells have very different sizes or large aspect ratios. Therefore we only expose in this paragraph the choice we made to build the reconstruction stencils.

A classical way to pick the neighbors is to iteratively consider layers of cells around the reference one until a chosen number of neighbors is reached (between 1 to 1.5 times $\mathcal{N}(\mathsf{d})$ in 2D and 1.5 to 2.0 times $\mathcal{N}(\mathsf{d})$ in 3D are demanded in practice). It follows from empirical experiences

that the more neighbors there are, the more robust the process is (but also the less accurate the reconstruction is, though still at right order of convergence). For instance in a (W)ENO method, the so-called central stencil is computed this way, while the so-called one-sided stencils (for the same cell) are chosen such that each of them is representative of a different direction.

Reminding that the quality of matrix X_i (invertibility, conditioning) is strongly impacted by the stencil, we propose to constrain the choice of neighbors with the conditioning number of the matrix. Fixing a target conditioning number, we try different neighborhoods until the resulting conditioning number of matrix X_i is below the target one. Note that the computational cost is not a problem since this operation is performed during the preprocessing step and the generated matrices, that are solution-independent, can be stored.

The picking algorithm for stencil $\mathcal{S}_i$ of a reference cell K_i consists in the following steps:

---

0. Set the number $N_{\mathcal{S}_i}$ of neighbors to pick
Set the target conditioning number $CN_{\mathcal{S}_i}$ for matrix X_i
Define a set $\overline{\mathcal{S}_i}$ of possible neighbors by considering layers of cells around K_i

Do while (conditioning number of $\mathsf{X_i} > \mathbf{CN}_{\mathcal{S}_\mathbf{i}}$)
  Do while ($\mathcal{S}_i$ contains less than $N_{\mathcal{S}_i}$)
    1. Randomly pick a cell in the set of possible neighbors $\overline{\mathcal{S}_i}$
    2. If it is a neighbor by face of an existing one, add the cell to the candidate stencil $\mathcal{S}_i$
  end do
  3. Compute the decomposition of the corresponding X_i and its conditioning number
end do

We now draw some remarks concerning this algorithm. First only the concept is represented by the above description, since the *do while* may not stop if it is not possible to reach the prescribed conditioning number with the prescribed number of cells. Consequently in that case, a balance has to be found between both constraints, and it is not an obvious task for now as it depends on the mesh and not only on the polynomial degree. Then, the target number of neighbors is the same as reminded above for classical methods and the target conditioning number is set by experiments trying to lower it at most. Furthermore the set of possible neighbors is computed by iteratively picking the neighbors by nodes of already picked cells with cell K_i as initialization, and condition 2. is imposed to ensure the *compactness* of the stencil.

Besides we would like to emphasize that lowering the stencil size at a given conditioning number is an interesting challenge since it would greatly lower the memory storage for reconstruction matrices (which grows linearly with the number of neighbors).

Finally we believe that a fine understanding of the reconstruction problem in the multidimensional case is a challenging problem for years to come and is the only way to make very high-order Finite Volume methods get into the engineering world where robustness is mandatory.

1.3 State-of-the-art high-order Finite Volume methods

In the previous sections, we dealt with Finite Volume numerical schemes to rich arbitrary high-order of accuracy on regular solutions. However the high-order approximations of steep gradients or discontinuities cause spurious phenomena and lead to the main difficulty in designing a high-order method. Indeed a so-called limitation technique has to be designed in order to still reach high-order on smooth solutions and prevent spurious effects on non-regular ones. This point is of crucial importance since the oscillations generated by the *unlimited* schemes may ruin solution accuracy or even worse, may lead to unphysical situations.

In this section, we give an overview of the state-of-the-art treatments to prevent such undesired behavior and give a short description of the two major types of limitation techniques currently used in multidimensional (very) high-order Finite Volume methods on unstructured meshes. We first expose in section 1.3.1 the second-order MUSCL [1] method which is historically one of the first high-order Finite Volume method and the most widespread in industrial simulation codes. Then section 1.3.2 is devoted to a reminder of the four major obstacles which create a second-order error and should be taken into account when designing higher-order methods. Finally in section 1.3.3, we present the higher-order (W)ENO [2] Finite Volume methods and the ADER method, which are the most established techniques to deal with very high-order Finite Volume methods available nowadays.

Let us draw a preliminary remark. Most of the existing limitation procedures have a common point, they act *a priori*, that is to say before the time update of the solution. That leads to two major consequences, we always treat every cells and we usually consider the worst-case scenario, while *unlimited* schemes perform well at least for smooth part of the solution. We shall highlight these intrinsic drawbacks in this section.

1.3.1 The MUSCL method

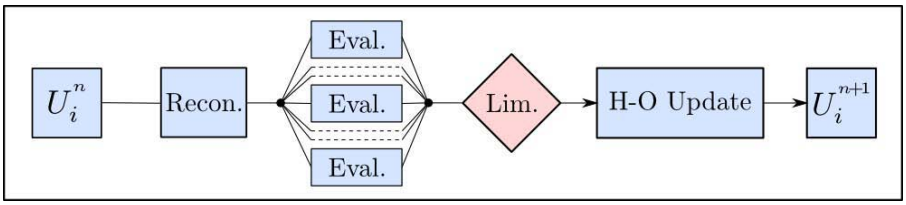


Figure 1.4: Simplistic flowchart of the MUSCL method: on each cell, limitation is performed after several evaluations of the unlimited polynomial reconstruction in order to enforce bounds for the evaluations at quadrature points.

In the 70's, Kolgan [48, 49, 50] and Van Leer [83] proposed a pioneering technique to improve the scheme accuracy for scalar conservation law. In a series of papers [83, 84, 85, 86, 87], Van

1. Monotone Upstream-centered Scheme for Conservation Laws 2. (Weighted) Essentially Non-Oscillatory

Leer designed the MUSCL scheme for the one-dimensional case by means of essentially two ingredients: a linear reconstruction for second-order accuracy in space coupled with a limitation procedure to enforce stability. As mentioned in section 1.1, the linear reconstruction (polynomial of degree one) is the key to reach second-order of accuracy and may produce spurious oscillations. In consequence the *limitation* of the reconstruction was introduced to prevent theses oscillations from appearing.

The MUSCL method was first developed in the one-dimensional case in order to fulfill the Total Variation Diminishing (TVD) property which ensures that the BV norm of the solution at time t^{n+1} is lower than the one at t^n. Then the MUSCL method has been applied to multidimensional Cartesian meshes through dimensional splitting [38, 75] where the TVD property is regarded in each direction.
Since we are interested in numerical methods for multidimensional unstructured meshes, we only present in the following the unstructured extension of the MUSCL method. In the 90's, the first unstructured extensions of the MUSCL method have been proposed in [8, 26] on the basis of a multidimensional linear reconstruction coupled with a slope limiter $\Phi_i \in [0, 1]$ under the form

$$\widetilde{u}_i(x) = u_i + \Phi_i \nabla_i(\mathbf{x} - \mathbf{c}_i),$$

such that we use the unlimited polynomial when $\Phi_i = 1$ and recover the mean value when $\Phi_i = 0$.

The notion of TVD in that case is not relevant anymore since Goodman & LeVeque proved in [35] that a multidimensional TVD-satisfying Finite Volume method would necessarily be first-order. Consequently the computation of the limiter Φ_i was first governed by the Maximum Principle that is fulfilled by the solution of the linear convection equation. More specifically the limitation was designed such that, for the convection equation, the numerical solution fulfills a Discrete Maximum Principle on mean values defined by

Definition 1.3 (Discrete Maximum Principle) *A Finite Volume numerical scheme providing mean values $\{u_i^{n+1}\}_{i\in\mathcal{E}_{el}}$ at time t^{n+1} from mean values $\{u_i^n\}_{i\in\mathcal{E}_{el}}$ at time t^n preserves a Discrete Maximum Principle (DMP) on mean values if the following property holds*

$$\min_{j\in\nu(i)}(u_i^n, u_j^n) \leq u_i^{n+1} \leq \max_{j\in\nu(i)}(u_i^n, u_j^n), \quad \forall i \in \mathcal{E}_{el},$$

where $\nu(i)$ is an index set of local neighbors of K_i.

As a consequence, the solution provided by such a scheme cannot contain new extrema and therefore undesired oscillations are prevented. In order to enforce this property, the basic idea of the multidimensional MUSCL method is to evaluate the unlimited linear reconstruction at several prescribed points of the cell and to compute Φ_i such that reconstructed values are bounded by neighboring mean values. Several versions can be found in [42] and references therein.

Finally for vectorial problems (*e.g.* the Euler equations), the same limitation is applied to each variable independently, even if the notion of maximum principle almost only applies in the

case of convection with divergence-free velocity. However the mathematical properties implied by such a limitation for those problems are not clear, and for instance, in the case of the Euler equations, the positivity of variables such as density or pressure can only be ensured by very restrictive timestep conditions, that might not be tractable for realistic complex simulations. Therefore the scheme is hardly robust and there is still on-going research to ensure positivity under lighter restriction, see [11, 61] for instance.

In conclusion, the MUSCL method clearly represents an important improvement of the first-order scheme, leading to more accurate numerical approximations. Furthermore its simplicity contributed to his popularity as the today most used Finite Volume method. Nevertheless its second-order nature is not enough to reproduce fine physical phenomena (*e.g.* contact discontinuities) on relatively coarse meshes and its *a priori* limitation may not ensure the physical meaning of the computed solution. At last, several second-order approximations are usually constitutive of the MUSCL method and prevent the concepts form being extended to higher-order methods. We review these points in next section and give a short presentation of the design principles developed for higher-order Finite Volume methods in section 1.3.3.

1.3.2 From second- to higher-order of accuracy

When designing higher-order (*i.e.* more than second-order) Finite Volume methods, several usual approximations of second-order methods are not valid anymore. In this section we recall the four points that must be carefully taken into account in the development of a higher-order Finite Volume scheme.

⋆ The analogy between cell mean value and cell centroid value

In second-order methods, the cell mean value is usually identified to the value at the cell centroid. Indeed in most of second-order methods, the slope is computed through a linear interpolation of the values at centroids (see [42]). Such an identification gives rise to a second-order error. The proof for the one-dimensional case follows: Let f be a smooth function and $[a, b]$ be an interval of centroid c and of length Δx. From a Taylor expansion we have $f(x) \approx f(c)+(x-c)\partial_x f(c)+\frac{(x-c)^2}{2}\partial_{xx} f(c)$ and using the definition of cell centroid $c = \frac{1}{\Delta x}\int_a^b x dx$ we obtain

$$\begin{aligned} \frac{1}{\Delta x}\int_a^b f(x)dx - f(c) &\approx \frac{\partial_x f(c)}{\Delta x}\underbrace{\left(\int_a^b x dx - c\right)}_{0} + \underbrace{\frac{\partial_{xx} f(c)}{2\Delta x}\int_a^b (x-c)^2 dx}_{\mathcal{O}(\Delta x^2).}, \\ &= \end{aligned}$$

The same reasoning is still relevant for the multidimensional case by definition of the centroid.

⋆ The Discrete Maximum Principle at smooth extrema

The MUSCL methods are usually designed to respect the Discrete Maximum Principle (definition 1.3) that applies on cell mean values because of the nature of Finite Volume methods.

As a consequence, an accuracy discrepancy at smooth extrema occurs. More precisely approximating the point-wise smooth extrema by the mean value one generates at best a second-order error since the first derivative vanishes in the Taylor expansion with respect to the point where extrema is reached. Consequently if the numerical scheme fulfills a strict DMP on mean values, this second-order error is propagated with time. In [27] (or chapter 3.2), a relevant example is drawn in appendix and a numerical assessment for the convection equation is also shown. Such considerations imply that the DMP is basically not used in the limitation process of existing higher-order methods. However we would like to recall that this order discrepancy only occurs at smooth extrema, and that consequently there is no reason to ban the use of DMP in the limitation process if we are able to distinguish smooth extrema from non-smooth one.

⋆ The non-linear combinations of mean values

From a general point of view, the approximation of the mean value of a non-linear combination of variables by the non-linear combination of the mean values of these variables is only second-order accurate. This claim is clear on this simple example: Let ρ and ϕ be two regular functions on cell K_i and ρ_i, ϕ_i, $(\rho\phi)_i$, denote their respective exact mean values. A Taylor expansion with respect to the centroid of the cell gives $(\rho\phi)_i = \rho_i\phi_i + O(h^2)$. For instance let us consider the following one-dimensional variables ρ, ϕ and $(\rho\phi)$ and their mean values on cell $K_1 = [0, h]$

$$\begin{aligned} &\rho(x) = 1 + x, \qquad &&\phi(x) = 1 - x, \qquad &&(\rho\phi)(x) = 1 - x^2, \\ &\rho_1 = 1 + h/2, \qquad &&\phi_1 = 1 - h/2, \qquad &&(\rho\phi)_1 = 1 - h^2/3. \end{aligned}$$

Then we obtain that $|(\rho\phi)_1 - \rho_1\phi_1| = h^2/12$ leading to a second-order error.

In our context, the problem arises when we consider vectorial problems, such as the hydrodynamics Euler system for which most of MUSCL methods use reconstructions of density, velocity components and pressure (*i.e.* so-called primitive variables) to ensure the physical meaning of reconstructed values at quadrature points. But velocity and pressure reconstructions are performed from mean values computed by non-linear combinations of the conservative variables mean values. Consequently even if polynomial reconstructions of degree greater than two are used, the high-order approximations at quadrature points are only second-order accurate, and so does the scheme. This has been numerically assessed in [27] (or section 3.2) for the Euler equations.

We would like to point out that there exist in the literature *higher-order* Finite Volume schemes that are based on polynomial reconstructions of primitive variables. Indeed this second-order error does not appear in some particular test cases to check the convergence (Ringleb flow, steady isentropic vortex) and it may be misleading. At last, it is important to remind that non-linear combinations of point-wise values or of polynomial coefficients conserve the higher-order property, see [31] for instance.

⋆ The approximation of curved boundaries by straight faces

In section 1.1, we assume that the domain Ω can be meshed by polygonal/polyhedral cells.

However in industrial simulations, the domain boundaries may be curved and consequently only approximated by straight faces. This linear approximation of boundaries also generates a second-order error. This claim is referenced in higher-order Finite Volume literature, see [58] for instance. In this thesis, we did not implement any treatment of curved boundaries since we do not consider such domains in our test cases.

1.3.3 Higher-order Finite Volume methods

In 1984, Woodward and Colella published the Piecewise Parabolic Method [91] as one of the first attempts to reach more than second-order of accuracy. The technique only dealt with one-dimensional geometry and up to our knowledge, never was extended in a truly multi-dimensional case though it is still used in some specific applications where dimensional splitting can be applied. Three years later, *Harten et al* proposed in [39, 40] the basic concepts of the ENO [3] method which has been widely studied and extended in [14, 1, 74] for instance. Then in 1994, *Liu & Osher* proposed in [54] the WENO [4] method as an important extension of the ENO one.

In next paragraph, we merge the presentations of ENO and WENO methods since their limitation techniques are based on same paradigms, and refer to them as (W)ENO methods. These methods are currently considered as the state-of-the-art higher-order Finite Volume methods. They have been successfully extended to a wide range of physical problems both on structured and unstructured meshes in 2D and 3D. Nonetheless, the first application of the WENO method for hydrodynamics Euler system on 3D mixed element meshes has been published in 2011 [82].

Finally an important improvement in the treatment of time discretization has been initiated in 2001 in [79] and led to the ADER [5] method which enables to reach arbitrary order of space-time discretization in only one step. It has been successfully applied to 3D tetrahedral meshes in [77, 31]. This technique, briefly presented in last paragraph of this section, may be seen as an interesting alternative to the method of lines.

⋆ The (W)ENO method

The fundamental idea of (W)ENO methods is to construct a non-oscillatory higher-order representation of the solution. To this end, for both ENO and WENO methods, several reconstruction stencils are chosen for each cell, and the reconstruction problem is solved for all of these stencils. Once all polynomials are available on a cell, ENO methods choose the least oscillatory one by comparing numerical approximations of an oscillatory norm while WENO methods compute the final polynomial as a convex combination with non-linear weights of all available polynomials such that the most oscillatory ones have lighter weights than the least oscillatory ones. At the end of the (W)ENO procedures, one essentially non-oscillatory polynomial per cell is available and used in the high-order Finite Volume scheme. The *a priori* limitation procedure is sketched in Figure 1.5.

3. Essentially Non-Oscillatory 4. Weighted Essentially Non-Oscillatory 5. Arbitrary accuracy DErivatives Riemann problem

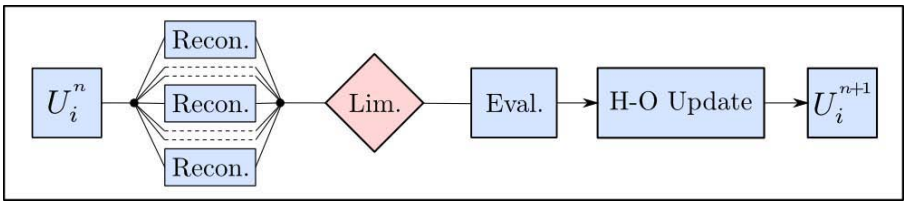


Figure 1.5: Simplistic flowchart of the (W)ENO method: on each cell, limitation is performed after computation of several polynomial reconstructions in order to obtain one Essentially Non-Oscillatory polynomial reconstruction for the evaluation step.

As for MUSCL, (W)ENO methods have been first developed and studied in the one-dimensional case for which several authors have proved the method ability to reach very high-order of accuracy on smooth profiles still maintaining an Essentially Non-Oscillatory behavior in the vicinity of steep gradients. Extensions to multidimensional problems on non-Cartesian meshes were first proposed by [33, 41] and are now effective on 3D mixed element meshes, see [82]. However these methods have their own drawbacks that we would like to mention in order to support the paradigms we used in the design of the MOOD method.

The first point is intrinsically attached to the limiting procedure; the need of several polynomial reconstructions per cell significantly increases the cost of the method in terms of memory and CPU. Indeed in a very high-order Finite Volume method, memory is mainly filled with the pseudoinverse matrices of the reconstruction problem even when only one polynomial per cell is considered. It is then important to note that in the very recent paper [82], the (W)ENO procedure uses at least N_f (number of faces) reconstructions per cell, and this is much less than the classical number (*e.g.* 7 for triangles and 9 for tetrahedral in [31]). Moreover as stated in [31] in page 239, obtaining the essentially non-oscillatory polynomial is the principal CPU time consuming part of the (W)ENO method (75% in their case). This remark, along with the fact that using only one centered reconstruction gives the same results on smooth profiles, led us to suggest the design of a very high-order Finite Volume method on the basis of only one polynomial per cell.

The second point depends on the problem to solve; the adaptation of (W)ENO methods to non-linear systems of conservation laws, such that the hydrodynamics Euler equations, is not straightforward in the sense that the set of variables on which the reconstructions are performed is usually not the conservative ones (*i.e.* the unknown of the scheme). Actually when the (W)ENO procedure is applied directly to the conservative variables, the essentially non-oscillatory behavior is not properly obtained, see results in [41] for instance. Consequently most of the (W)ENO implementations use characteristic variables since they fulfill independent scalar conservation laws. Two remarks follow from this claim. The first one is that the transformation from conservative variables to characteristic ones is costly in CPU time, *e.g.* the matrix of right eigenvectors and its (analytical) inverse have to be computed at every interface. The second point is that it makes the numerical scheme much more complex. Consequently our

design will only consider performing polynomial reconstruction on the conservative variables.

The last point is about mathematical analysis; as for the MUSCL method the main properties of the (W)ENO methods have been proved in the one-dimensional case and have not been extended to the multidimensional one yet. For instance, the positivity-preserving ability of (W)ENO methods is not ensured. We believe that these difficulties are generated by the *a priori* limitation which makes difficult the mathematical analysis.

⋆ The ADER method

In this paragraph, we draw the big picture of the the ADER method that is the only higher-order Finite Volume method able to reach both space and time higher-order of accuracy in a single step. For an history of the method and all the implementation details, the reader is referred to the thesis of C.E. Castro [15] for instance.

Thus far, we only considered the so-called method of lines to design our higher-order Finite Volume scheme by means of the RK3-TVD method for time discretization, see equation (1.8). We recall that this choice was induced by its simpler implementation and the fact that it is independent of equations, but has two drawbacks: being only 3^{rd}-order accurate and using several steps of the spatial high-order scheme. Actually, these two points can be overcome by using more intensively the equations of the problem. The starting point is to integrate equation (1.2) in the time interval $[t^n, t^{n+1}]$ to get the exact time update

$$U_i^{n+1} - U_i^n + \frac{1}{|K_i|} \sum_{j \in \mathcal{V}(i)} \int_{t^n}^{t^{n+1}} \int_{f_{ij}} F(U(\mathbf{x}, \tau)) \cdot \mathbf{n}_{ij} \, ds d\tau = 0.$$

It is thus clear that building a single step higher-order space-time scheme is equivalent to obtaining a higher-order approximation of the space-time flux integral. Following the same philosophy as in section 1.1 but taking into account the time integral, we consider a quadrature rule on the space-time element $f_{ij} - [t^n, t^{n+1}]$ for which we need to provide higher-order approximations of the flux. Notice that a space-time quadrature formula can easily be obtained by tensor product of a quadrature rule on the face with one on the time interval. Therefore the whole problem is to obtain higher-order evaluations of the solution at different time levels for each quadrature position on the face: this is the fundamental difference with our scheme for which the solution is always evaluated at the current time of the RK3-TVD step.

We now present the original ADER method developed by Titarev and Toro in [80]. The final goal of this method is to obtain at each spatial quadrature positions a unique polynomial approximation of the solution in time. The fundamental idea is to completely use the spatial informations contained by the two polynomials in K_i and K_j, instead of only using the extrapolated values as we do. To this end, the Riemann Problem (that we use in equation (1.3)) is replaced by a Derivative Riemann Problem (DRP) in which the left and right states are not constant anymore but entire polynomials. The solution of the DRP is therefore not constant anymore with time, and the solution is sought under the form of a power series expansion in time. Once such a time representation is available, the solution can be evaluated at the time

quadrature points to compute the numerical flux with a high-order of accuracy.

Nevertheless it remains to solve the DRP and this is not a trivial task. The Titarev and Toro technique relies on the following steps. First, compute the left and right polynomial states by rotating polynomials on K_i and K_j in the normal direction $\mathbf{n}_{ij}$, and consider their extrapolated values of the solution and its derivatives at the spatial quadrature position. At this step, for each spatial quadrature position, we have left and right constant values for the solution and its spatial derivatives. Then similarly to section 1.1, the idea is to obtain unique values of the solution and its spatial derivatives. To this end, a series of Riemann Problems is solved. More precisely, the leading term of the solution is obtained by solving the same Riemann Problem that we use in equation (1.3). Then considering the linearized equations (around the leading term) fulfilled by each spatial derivative, a unique value of this derivative is obtained by solving a Riemann Problem using the left and right corresponding (constant) values for this derivative. The final stage is to transform these spatial informations into informations in time; this is achieved by means of the so-called Cauchy-Kowalesky procedure: considering that the solution of the problem is smooth enough, the time derivatives can be computed from the spatial ones using the equations of the problem. This procedure is well known and clearly detailed in [31]. Finally for each spatial quadrature point a polynomial approximation of the solution in time is available and is used to compute the flux at the corresponding space-time quadrature points.

Remark: Variants of the ADER method use the same ingredients, but in another order. For instance in the Castro and Toro version, the Cauchy-Kowalesky procedure is first applied to the extrapolated values and derivatives at quadrature points on both sides of f_{ij}, and the time derivatives of the solution are directly obtained by solving classical Riemann problems for the linearized (around the leading term) equations fulfilled by the time derivatives.

Remark: In the recent paper [29], the authors propose a problem-independent ADER method for which an iterative procedure replaces the complex Cauchy-Kowalesky procedure. Although this particular ADER method is problem-independent, it is still much more complicated than the RK3 method and that explains why we did not use it.

Finally the limitation of oscillations for the ADER methods is ensured by a (W)ENO procedure so that the same drawbacks are present. Nevertheless it is worth noticing that the WENO procedure is only done once per time step while for Runge–Kutta-based methods the costly procedure is repeated for each time sub-step. Therefore the great interest of this approach is in the fact that arbitrary high-order of accuracy in time can be achieved. Hence in our context, the ADER method may be seen as an efficient time discretization which overcomes all problems of the RK3-TVD. However our purpose is to design a scheme independent of the problem and that can be easily implemented from a first-order code, consequently we did not implement an ADER type of time discretization. Nevertheless it is discussed in the perspectives part of this thesis as an interesting possibility.

Conclusion

The purpose of this chapter was to give a brief but clear introduction to the existing techniques to reach higher-order of accuracy in a Finite Volume framework in order to support the choices we made in the design of the MOOD method which is presented in next chapter.

First we have presented an arbitrary high-order Finite Volume scheme based on convex combinations of the standard first-order Godunov scheme and on a separated treatment of the space and time discretizations. It is simple to implement in an existing first-order code and that explains our motivation to choose it in the context of this thesis. However such unlimited schemes are not suitable for simulations since they are not able to properly deal with solutions that are not smooth. As a consequence, the main goal of this thesis is to propose a novel approach to limitation.

In a second part, we have given an overview of the limitation concepts of the state-of-the-art higher-order Finite Volume methods. From the second-order MUSCL method which is very simple and inexpensive but not accurate enough, to the (W)ENO methods which are the most widespread Finite Volume methods effectively capable to reach more than second-order of accuracy.

The limitation procedures of these methods are always performed *a priori*, or in other words before the update of the solution mean values. This implies that on smooth profiles a lot of unnecessary computational effort is performed and that the worst-case scenario is usually considered in the limitation design. We shall show that the computational cost can be significantly reduced by treating the arising problems *a posteriori*.

Furthermore the multidimensional (W)ENO methods are costly in CPU and memory storage because several polynomials per cell are reconstructed. Let us recall that on smooth profiles the central reconstruction stencil alone is enough to reach higher-order without any problems and we believe that using only one reconstruction per cell is an achievable goal.

Last but not least, the mathematical analysis of such *a priori* multidimensional methods is very difficult, and robustness problems may occur when dealing with physical problems. For instance, most of (W)ENO methods are not proved to be positivity-preserving, and the classical MUSCL methods are positivity-preserving only under a very restrictive timestep condition. We shall show in next chapter that the *a posteriori* nature of the MOOD method make easy to enforce such a properties by construction.

Chapter 2

The Multi-dimensional Optimal Order Detection (MOOD) method

Introduction

Following the conclusions drawn in previous chapter, we propose to design a method that avoids useless *a priori* limitations. The development of the MOOD method has been ruled by two main features: simplicity and efficiency. The MOOD philosophy holds in *"Compute a time-updated solution with the highest-order unlimited scheme and if it locally fails, recompute with a lower-order one till it succeeds"*, for which a simplified schematic is given in Figure 2.1. It explains the *Optimal Order Detection* denomination of the method since, in a sense, the principle is to find the scheme of maximal order which provides an *acceptable* solution. As simple as it seems, the idea of an *a posteriori* treatment of problems generated by unlimited schemes is the key point of the concept. It basically relies on the definition of *acceptable* solution as well as on the fact that in the worst case scenario a stable first-order scheme will always provide a solution free of spurious phenomena.

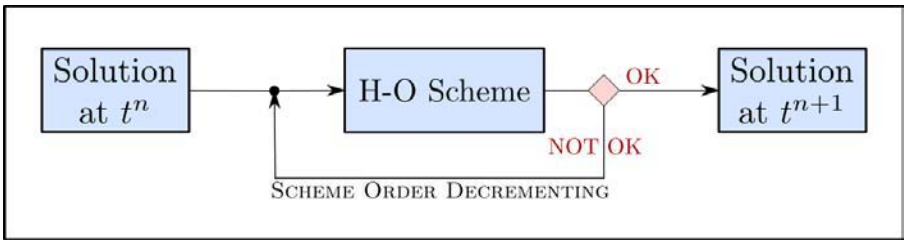


Figure 2.1: The simplistic concept of the MOOD method.

Nevertheless simplicity is not sufficient and the method also has to be efficient. Therefore as previously mentioned, we choose to develop the MOOD method on the basis of only one polynomial reconstruction per cell since the costly part (in terms of memory storage and CPU) of a higher-order Finite Volume multidimensional method is the polynomial reconstruction. An important computational gain is thus to be expected compared to (W)ENO methods for which

several higher-order polynomials per cell have to be considered, see section 1.3.3. Furthermore, we would like to point out that the *a posteriori* nature of the method is new with respect to traditional techniques. Therefore we advise the reader to keep in mind the simple schematic of Figure 2.2 that can be compared with those of classical methods presented in section 1.3.

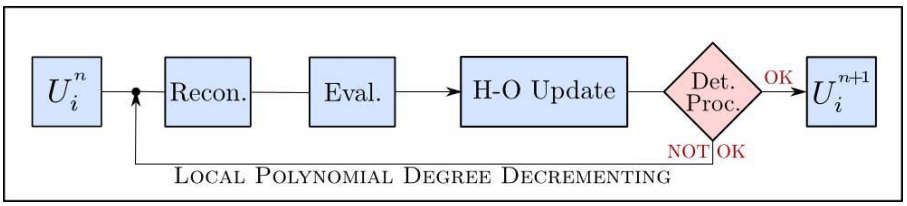


Figure 2.2: Simplistic flowchart of the MOOD method: each cell K_i is first updated by an unlimited high-order scheme, then an *a posteriori* Detection Process (Det. Proc.) is performed and cells that are not satisfactory are re-updated with a lower polynomial degree until all cells are satisfactory.

The appropriate framework for the MOOD method, from fundamental notions to a complete algorithm view, is defined in section 2.1. Then in sections 2.2 and 2.3, applications of the MOOD method are detailed for the scalar convection equation and for the Euler hydrodynamics system respectively. Finally we present in last section of this chapter some key optimizations to ensure the efficiency of the MOOD concept.

2.1 Design of the MOOD method

Although the basic idea seems to be unrefined, the definition of an effective framework to the MOOD method is patently not that simple and demands a careful design to preserve important properties usually mandatory in Finite Volume methods such as conservativity. Furthermore the whole *a posteriori* concept relies on the definition of *"a solution fails"*, or in other words, of what is an *acceptable solution*. Therefore a clear definition of *acceptable solution* is given in subsection 2.1.2, in which we begin with defining two notions related to the polynomial reconstructions (and thus to the scheme order). But first and foremost, we give three denominations associated to the *a posteriori* framework in next subsection.

Since we apply the MOOD concept to each on the three substeps of the RK3-TVD (1.8) in the same way, we restrict our presentation in this section to the spatial higher-order scheme (1.5) for the sake of clarity.

2.1.1 Some vocabulary to handle the *a posteriori* nature

Let us introduce three ingredients to easily deal with the *a posteriori* nature of the MOOD method. The first one, that we name *MOOD algorithm*, is the overall iterative process sketched in Figure 2.1 (or Figure 2.2) which intrinsically defines the MOOD method. That is to say, the process which consists of first computing a solution with the highest-order numerical scheme and then, iteratively recomputing a new solution with a locally lower-degree scheme until the

solution is *acceptable*.

Moreover the expression *candidate solution* refers to any solution provided by an unlimited numerical scheme for which we must check if it locally fails or succeeds.

Finally, the process of deciding if a candidate solution is *acceptable*, depicted in red in Figure 2.1 and Figure 2.2, is denominated *detection process*. The design of such detection processes in the cases of the convection equation and of the hydrodynamics Euler system is described in sections 2.2 and 2.3 respectively.

2.1.2 Fundamental notions and properties

This section is separated into two parts. Firstly we define two notions related to the polynomial reconstruction, or equivalently related to the local order of accuracy of the scheme in our context, and give the related important properties. Secondly we propose a definition of *acceptable solution* and then give the proof of the MOOD algorithm convergence.

⋆ Cell and Face Polynomial Degrees

While classical higher-order Finite Volume schemes use same order of accuracy in all cells, the MOOD method strongly exploits the local scheme order decrementing procedure that allows different spatial orders of accuracy from a cell to another. In consequence we have to define specific notions to handle the local order of accuracy; these are the roles of the Cell and Face Polynomial Degrees defined below.

We name Cell Polynomial Degree, shortened as CellPD and denoted by d_i, the degree of the polynomial reconstruction on cell K_i. It is used in the decrementing procedure to enforce that the update of the solution in cell K_i is performed with an at most (d_i+1)-order scheme. This property, ensured by Theorem 2.2, guarantees that in the worst case scenario, the first-order scheme is used.

We name Face Polynomial Degrees, shortened as FacePD and denoted by d_{ij} and d_{ji}, the degrees of the polynomial reconstructions actually used to compute approximations, $U_{ij,r}$ and $U_{ji,r}$, of the solution on face f_{ij} at quadrature points $q_{ij,r}$ respectively from K_i and K_j. The computation of d_{ij} and d_{ji}, named FacePD strategy, must be done in accordance with CellPD of the two neighbors and must be *upper-limiting*, that is

Definition 2.1 (Upper-limiting) *A FacePD strategy is said to be* upper-limiting *with respect to the CellPD if for any K_i and any degree $\bar{\mathsf{d}}$, the following property holds*

$$\mathsf{d}_i = \bar{\mathsf{d}} \Longrightarrow \mathsf{d}_{ij} \leq \bar{\mathsf{d}} \quad \textit{and} \quad \mathsf{d}_{ji} \leq \bar{\mathsf{d}}, \quad \forall j \in \underline{\nu}(i).$$

Notice that the least restrictive upper-limiting FacePD strategy is $\mathsf{d}_{ij} = \mathsf{d}_{ji} = \min(\mathsf{d}_i, \mathsf{d}_j)$, and that we will use this strategy in the whole thesis. This property ensures the local order of accuracy of the update, that is

Theorem 2.2 (Local order of accuracy) *We consider a higher-order Finite Volume scheme (1.8) designed to reach an order of accuracy* $(\mathsf{d}_{max}+1)$*. If the CellPD of a cell* K_i *equals* $\mathsf{d} \leq \mathsf{d}_{max}$ *and the FacePD strategy under consideration is upper-limiting then the order of accuracy of the solution update in* K_i *is at most* $(\mathsf{d}+1)$*th-order accurate.*

Proof: *In the cell* K_i*, the upper-limiting property of the FacePD strategy implies that, both FacePD (*d_{ij} *and* d_{ji}*) on all faces* f_{ij}*, for* $j \in \underline{\nu}(i)$*, are lower or equal to* d*. Consequently all approximations,* $U_{ij,r}$ *and* $U_{ji,r}$*, at quadrature points* $q_{ij,r}$ *are computed from an at most* d*-degree polynomial, so they are at most* $(\mathsf{d}+1)$*-order accurate. Since the leading error of the rest of the scheme is* $(\mathsf{d}_{max}+1) > (\mathsf{d}+1)$*, the error of the whole scheme is dominated by the spatial error of approximations. That is to say the scheme is at most* $(\mathsf{d}+1)$*th-order accurate.*

□

A straightforward but important corollary follows

Corollary 2.3 (First-order update) *We consider a higher-order Finite Volume scheme (1.8) designed to reach an order of accuracy* $(\mathsf{d}_{max}+1) > 1$*. If the CellPD of a cell* K_i *is equal to* 0 *and the FacePD strategy under consideration is upper-limiting then the solution in* K_i *fulfills all the properties of the first-order scheme (1.4).*

Proof: From theorem 2.2, the solution is locally first-order, and from proposition 1.1 the solution is computed from the first-order Godunov scheme (1.4) and so fulfills its properties.

□

Let us now draw two remarks. Firstly, according to proposition 1.2 the corollary only holds for the properties of the first-order scheme (1.4) that are preserved by convex combinations (DMP, bounds on variables, etc.) when considering the complete RK3-TVD (1.8) time discretization. Secondly, for the sake of simplicity we only consider in this work decrementing the local scheme order by reducing the local polynomial reconstruction degrees, hence neglecting the accuracy reduction of quadrature rules for instance. Nevertheless proposition 1.1 ensures it is sufficient.

⋆ $\mathcal{A}$-eligibility and acceptability of a solution

In previous paragraph, we provided some tools to handle the scheme-order decrementing. Only one part of the MOOD concept remains to be properly defined: the notion of *acceptable solution*. It is certainly the most important notion since the final solution properties will mainly depend on it. In this paragraph, we first define the set of *detection criteria* $\mathcal{A}$ that is used in the detection process, then give a clear definition of *acceptable solution* and finally we prove that the MOOD method always converges to an *acceptable solution* under non-restrictive assumptions.

Thanks to the *a posteriori* framework of the MOOD method, we expect to have a simplified control on the final solution properties. Actually this control is achievable through the detection process which consists in checking the compliance of a solution to a set of *properties* to decide if solution fails or succeeds. In the sequel, such a set of properties (that we shall prescribe further) is called *(set of) detection criteria* and denoted by $\mathcal{A}$, while a solution is said to be $\mathcal{A}$-eligible

if it fulfills all criteria of $\mathcal{A}$. Nevertheless it is obviously not feasible to ensure that the solution will be $\mathcal{A}$-eligible regardless of the set $\mathcal{A}$, and so, to ensure that the MOOD algorithm has a finite number of iterations. Therefore a definition of an *acceptable solution* is

Definition 2.4 (Acceptable solution) *Let $\{U_i^\star\}_{i\in\mathcal{E}_{el}}$ be a candidate solution and $\mathcal{A}$ be a set of prescribed detection criteria. For a given cell K_i, the candidate solution $U_i^\star$ is said to be* acceptable *if either $U_i^\star$ is $\mathcal{A}$-eligible, or $U_i^\star$ has been computed by the first-order scheme (*i.e. *CellPD is already zero). Accordingly the solution $\{U_i^\star\}_{i\in\mathcal{E}_{el}}$ is said to be* acceptable *if $U_i^\star$ is* acceptable *for all i in $\mathcal{E}_{el}$.*

We would like to emphasize the fact that an acceptable solution is not necessarily $\mathcal{A}$-eligible, and consequently this is important that the first-order scheme provides a solution free of spurious phenomena. The following theorem proves the viability of the MOOD method.

Theorem 2.5 (Finite number of iterations of the MOOD algorithm) *Let $\{U_i^n\}_{i\in\mathcal{E}_{el}}$ be a solution at time t^n and $\mathcal{A}$ be a prescribed set of detection criteria. If the FacePD strategy is upper-limiting then the MOOD algorithm always provides an acceptable solution at time t^{n+1} in a finite number of iterations.*

Proof: *The proof is performed by separating cases. The MOOD algorithm stops if the solution is acceptable on all cells, and solution is acceptable if either it is $\mathcal{A}$-eligible or updated with the first-order scheme. Therefore if we consider a given cell K_i, two situations are possible: either the solution is $\mathcal{A}$-eligible and there is no problem; Or the solution is not $\mathcal{A}$-eligible and the CellPD is decremented until either the new candidate solution is $\mathcal{A}$-eligible, or until the CellPD d_i is zero. If $\mathsf{d}_i = 0$, the corollary 2.3 implies that the solution is computed by the first-order scheme and so, the solution is acceptable. This reasoning is valid for all cells, thus the maximal number of iterations of the MOOD algorithm is the product of the number of cells times the initial degree of polynomial reconstruction (say d_{max}), because each cell can at most be decremented d_{max} times.*

□

In this section, we defined the theoretical framework to ensure that such an *a posteriori* approach for limitation is viable in the sense that it always provides a solution in a finite number of iterations. Though it is a mandatory point, it does not give any informations about the efficiency of the technique. However we shall see later that in real computations the number of iterations of the MOOD algorithm remains low and provides an efficient method when combined to the optimizations proposed in last section of this chapter.

2.1.3 The MOOD algorithm

We now complete the generic design of the MOOD method by presenting the different stages of the MOOD algorithm. We only give here the general steps that must be respected exactly whichever the implementation. We refer the reader to section 2.4 for the *pseudo-code* version of the algorithm as it is implemented in our code.

The MOOD algorithm takes place in each of the three substeps of the RK3-TVD method. It is thus embedded in the Runge-Kutta loop that is itself embedded in the loop from initial to final

times. Therefore the different stages of the MOOD algorithm are presented for one substep of the Runge-Kutta loop (the algorithm is depicted in Figure 2.3).

The zero-th stage is the initialization of the CellPD of each cell to the maximal degree, denoted d_{max}, *i.e.* we set $\mathsf{d}_i = \mathsf{d}_{max}$ to initialize the degree of the polynomial reconstruction that is performed on each cell at the first stage. The second stage consists in computing each FacePD in accordance with the CellPD and then evaluating the higher-order approximations at quadrature points with the degree of the FacePD. Then a candidate solution is computed during the third stage using the higher-order space scheme of equation (1.5). Therefore the fourth stage is the detection process coupled with the CellPD decrementing of cells on which solution is not acceptable. If there exist cells for which the solution is not acceptable then the algorithm goes back to the first stage and uses the new CellPD map. This algorithm stops when the solution is acceptable for all cells (theoretically ensured by theorem 2.5).

Do while ($\mathbf{t} < \mathbf{t_{final}}$)
  Do RK=1,3
    0. Initialization of CellPD $\mathsf{d}_i = \mathsf{d}_{max},\ \forall i \in \mathcal{E}_{el}$
    Do while (solution is not acceptable)
      1. Polynomial reconstruction of degree $\mathsf{d}_i,\ \forall i \in \mathcal{E}_{el}$
      2. Computation of FacePD d_{ij} and Evaluation of high-order approximations $U^n_{ij,r}$ at quadrature points $q_{ij,r}$ with degree d_{ij}, $\forall i \in \mathcal{E}_{el},\ \forall j \in \underline{\nu}(i),\ \forall r = 1, .., R_{ij}$
      3. Solution mean values update using equation (1.5)
      4. Detection process: if solution in K_i is not acceptable decrement CellPD d_i
    end do
  end do
end do

Figure 2.3: The MOOD algorithm: complete overview.

Notice that for the sake of clarity, we considered the RK3 method as a loop over an index $RK = 1, 2, 3$ though the loop is unrolled in the code for efficiency.

For specific applications, one has to provide the detection criteria with respect to physical and/or mathematical properties of the problem (positivity of variables, maximum principle, etc.). In the next two sections, we propose suitable detection criteria for the convection equation and for the hydrodynamics Euler system respectively.

2.2 Application to the Convection equation

2.2.1 Equation and Finite Volume scheme

Thus far, we have provided all details to implement the MOOD method except from one aspect: the detection process, or more precisely the set of detection criteria $\mathcal{A}$. This point is of

crucial importance though, since the final solution strongly depends on it. In this section, we propose such a set in the case of the simple (but nonetheless representative) scalar convection equation

$$\begin{cases} \partial_t U(\mathbf{x},t) + \nabla \cdot \big(V U(\mathbf{x},t) &= 0, & \text{with } \mathbf{x} \in \Omega \subset \mathbb{R}^n,\ t > 0, \\ U(\mathbf{x},0) &= U_0(\mathbf{x}), & \text{with } \mathbf{x} \in \Omega \subset \mathbb{R}^n, \end{cases}$$

where $U(\mathbf{x},t) \in \mathbb{R}$ is the unknown, $U_0(\mathbf{x}) \in \mathbb{R}$ the initial condition and $V \in \mathbb{R}^n$ is a constant prescribed velocity. Moreover we prescribe periodic conditions on the domain boundary $\partial\Omega$.

This scalar equation is the starting point of many numerical methods since the linearity avoids the complexity of non-linear behaviors and the analytical solution is given by

$$U(\mathbf{x},t) = U_0(\mathbf{x} - Vt) \quad \forall \mathbf{x} \in \Omega \subset \mathbb{R}^n,\ \forall t > 0. \tag{2.1}$$

In particular, it is clear from equation (2.1) that the exact solution fulfills a maximum principle. This property still holds if the velocity is space-dependent, $V = V(\mathbf{x})$ but divergence-free $\nabla \cdot V = 0$.

The numerical scheme is defined by equation (1.6) and equation (1.8), where the numerical flux $\mathbb{F}$ at time t^n for any quadrature point $q_{ij,r}$ is given by the classical upwind flux

$$\mathbb{F}(U^n_{ij,r}, U^n_{ji,r}, \mathbf{n}_{ij}) = \max(0, V \cdot \mathbf{n}_{ij}) U^n_{ij,r} + \min(0, V \cdot \mathbf{n}_{ij}) U^n_{ji,r}.$$

Finally the treatment of periodic boundary conditions is the following: for each quadrature point on a boundary face, the outside value is set by duplicating the inside value at the corresponding quadrature point of the corresponding periodic face. Moreover after each detection process, the CellPD map is treated to respect periodicity.

2.2.2 Detection process

In this section, we define the effective detection process we employ to solve the convection equation. We expect the numerical solution to respect the maximum principle since the exact solution does so. However in the Finite Volume context, the traditional maximum principle is the Discrete Maximum Principle (DMP) which compares the mean values at time t^{n+1} to the ones at t^n:

$$\min_{j \in \nu(i)} (U^n_i, U^n_j) \leq U^{n+1}_i \leq \max_{j \in \nu(i)} (U^n_i, U^n_j), \tag{2.2}$$

where $\nu(i)$ represents a neighborhood of cell K_i.

It is noteworthy that the fundamental concept of the MUSCL method to prevent spurious oscillations from appearing is to enforce the solution to fulfill such a DMP. It is then a natural choice to reduce the set of detection criteria $\mathcal{A}$ to the DMP in a first approach. In other words, once we have a candidate solution $\{U^\star_i\}$ (computed from an unlimited higher-order scheme), we check during the detection process if this candidate fulfills the DMP, *i.e.*

$$\min_{j \in \overline{\nu}(i)} (U^{RK}_i, U^{RK}_j) \leq U^\star_i \leq \max_{j \in \overline{\nu}(i)} (U^{RK}_i, U^{RK}_j), \quad \forall i \in \mathcal{E}_{el}, \tag{2.3}$$

where the superscript ^{RK} refers to the input mean values of three steps of RK3-TVD, *i.e.* $U_i^{RK} = U_i^n, U_i^1$ or $\widehat{U}_i^2$ in equation (1.8), since we recall that the MOOD algorithm is independently applied to each step of RK3.

Unfortunately as mentioned in section 1.3.2, such a DMP implies a second-order error at smooth extrema. However before going further in the development of a suitable detection process, we would like to highlight an appealing property of the *a posteriori* nature of the method. Contrary to the MUSCL method for which (even in 1D) the proof the DMP property is not mathematically straightforward and implies a necessary restricted timestep, the following theorem holds in our context:

Theorem 2.6 *Let $\{U_i^n\}_{i\in\mathcal{E}_{el}}$ be a solution at time t^n and let $\mathcal{A}$ be constituted of the Discrete Maximum Principle (DMP) principle of equation (2.3). If the FacePD strategy is upper-limiting then the MOOD algorithm always converges and provides a solution for time t^{n+1} which satisfies the DMP property of equation (2.2) under the CFL condition of the first-order scheme (1.4).*

Proof: *Let us first recall that the first-order Godunov scheme (1.4) provides a DMP-satisfying solution under a CFL stability condition. Then theorem 2.5 ensures that at each substep of RK3-TVD the MOOD algorithm always provides an acceptable solution, meaning that either the solution is $\mathcal{A}$-eligible,* i.e. *DMP-satisfying, or the solution is computed from the first-order scheme which also provides a DMP-satisfying solution. Remark that the DMP property is satisfied when considering U_h^1 compared to U_h^n, U_h^2 compared to U_h^1 and U_h^3 compared to $\widehat{U}_h^2$. However since every U_h^{RK} is a convex combination of solutions that satisfy a DMP property in regard to U_h^n and that the final solution U_h^{n+1} is computed by a convex combination of same type, we can conclude that the final solution satisfies the DMP of equation (2.2) under the CFL of the first-order scheme. Finally it is worth noticing that each substep of RK3 fulfills a DMP over the neighborhood $\overline{\nu}(i)$ while for the complete scheme a larger one is to be considered.*

□

Keeping in mind the remark we drew in 1.3.2, the strict DMP on mean values as detection criteria provides a second-order scheme. Therefore we have two alternatives, either we choose not to use the DMP as the main criterion and we have to find a different way to proceed; Or we have to introduce new mechanisms to overcome accuracy discrepancy at smooth extrema. We choose to consider the latter, for at least two reasons: firstly, the DMP is legitimate from a mathematical point of view; secondly, it is very easy and computationally efficient to test it on a solution even on 3D unstructured meshes. Therefore we propose a detection process on the basis of successive filters where the DMP is used as the first filter and a more refined test is applied as second filter on cells which do not respect the DMP. This last filter, denominated $u2$, has been designed in [27] and improved in [28] in order to detect if a cell where the solution violates the DMP corresponds to a smooth extrema. In the following the whole detection process is shortened to [DMP $\rightarrow u2$] which recalls the succession of detection criteria.

In this context, the notion of numerical smoothness is independent of the time discretization, hence we define the $u2$ detection criterion in the case of a candidate solution $U_h^\star$ computed from solution U_h^n. Moreover the definition is given for the 3D case, and extra-dimensions have to be

omitted for the 1D and 2D cases.

The purpose of the $u2$ detection criteria is to give a definition to the notion of *numerical smooth extrema* in order to allow a relaxation of the DMP at these locations while preventing DMP violation around steep gradients or discontinuities. So let us consider a cell K_i where the solution does not fulfill the DMP, we want to determine if the solution was smooth at time t^n.

The first step is to reconstruct quadratic (*i.e.* degree two) polynomials on K_i denoted by $\widetilde{U}_i$ and on its neighbors K_j for $j \in \overline{\nu}(i)$ denoted by $\widetilde{U}_j$. Then we define

$$\mathcal{X}_i^{min} = \min_{j\in\overline{\nu}(i)} \left(\partial_{xx}\widetilde{U}_i,\ \partial_{xx}\widetilde{U}_j\right) \quad \text{and} \quad \mathcal{X}_i^{max} = \max_{j\in\overline{\nu}(i)} \left(\partial_{xx}\widetilde{U}_i,\ \partial_{xx}\widetilde{U}_j\right),$$

$$\mathcal{Y}_i^{min} = \min_{j\in\overline{\nu}(i)} \left(\partial_{yy}\widetilde{U}_i,\ \partial_{yy}\widetilde{U}_j\right) \quad \text{and} \quad \mathcal{Y}_i^{max} = \max_{j\in\overline{\nu}(i)} \left(\partial_{yy}\widetilde{U}_i,\ \partial_{yy}\widetilde{U}_j\right),$$

$$\mathcal{Z}_i^{min} = \min_{j\in\overline{\nu}(i)} \left(\partial_{zz}\widetilde{U}_i,\ \partial_{zz}\widetilde{U}_j\right) \quad \text{and} \quad \mathcal{Z}_i^{max} = \max_{j\in\overline{\nu}(i)} \left(\partial_{zz}\widetilde{U}_i,\ \partial_{zz}\widetilde{U}_j\right),$$

where we emphasize that the second derivatives are constant and referred to as *curvatures*. The $u2$ detection criterion holds in the following definition

Definition 2.7 (u2 detection criterion) *A candidate solution $U_i^\star$ in cell K_i which violates the DMP is nonetheless eligible if*

$$\mathcal{X}_i^{max}\mathcal{X}_i^{min} > 0 \quad \textit{and} \quad \left|\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}}\right| \geq 1-\varepsilon,$$

$$\textit{and} \quad \mathcal{Y}_i^{max}\mathcal{Y}_i^{min} > 0 \quad \textit{and} \quad \left|\frac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}}\right| \geq 1-\varepsilon,$$

$$\textit{and} \quad \mathcal{Z}_i^{max}\mathcal{Z}_i^{min} > 0 \quad \textit{and} \quad \left|\frac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}}\right| \geq 1-\varepsilon,$$

where ε is a smoothness parameter.

This definition relies on the idea that comparing the second derivatives on a local neighborhood is sufficient to determine the numerical smoothness of a piecewise constant function. More precisely, we consider that the solution is smooth if in each direction, the curvatures have the same sign (no oscillation) and are *close* enough to each-other, in a sense to be characterized by the smoothness parameter ε.

More specifically, the value of ε defines the threshold between what is considered as a smooth extrema or as a discontinuity. Let us first remark that ε must range in $[0, 1]$ to make sense since the ratios between minimal and maximal curvatures are bounded by zero and one in the right inequalities of definition 2.7. Moreover the limitation generated by ε is such that the closer to zero ε is, the less smooth the functions will be considered. Therefore we may consider ε as a function that has to be close to zero on discontinuities and close to one on smooth functions.

We thus propose to define ε_x (in the x-direction) as a continuous increasing function of the (x-direction) curvatures ratio such that $\varepsilon_x(0) = 0$ and $\varepsilon_x(1) = 1$. After several attempts, it appears that the simple function $\varepsilon_x(r) = r$ is an relevant choice leading to the following criterion for the x-direction curvatures

$$\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}} \geq 1 - \frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}},$$

that yields

$$\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}} \geq 1/2.$$

Finally applying the same reasoning for y- and z-directions, we obtain

$$\frac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}} \geq 1/2 \quad \text{and} \quad \frac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}} \geq 1/2.$$

Note that the linearity of functions ε_x, ε_y and ε_z simplifies the final inequalities and leads to the constant value $\varepsilon = 1/2$ in definition 2.7.

Numerically, we have shown in [27] et [28] that the $u2$ detection criterion relaxes the DMP as expected: reaching up to 6^{th}-order of accuracy while maintaining a non-oscillatory behavior on discontinuous profiles. However no rigorous mathematical framework has been found yet to support these observations. Note that extensive numerical tests have been performed and compared to the ones obtained with the first value proposed for ε in [27] and no significant differences have been noticed resulting in same quality results.

To conclude the design of the detection process for the convection equation, we sketch in Figure 2.4 the stages of the [DMP $\rightarrow u2$] embedded in the MOOD algorithm.

---

0. Initialization of CellPD $\mathsf{d}_i = \mathsf{d}_{max},\ \forall i \in \mathcal{E}_{el}$

Do while (solution is not acceptable)

1. Polynomial reconstruction of degree $\mathsf{d}_i,\ \forall i \in \mathcal{E}_{el}$

2. Computation of FacePD d_{ij} and Evaluation of high-order approximations $U_{ij,r}^n$ at quadrature points $q_{ij,r}$ with degree $\mathsf{d}_{ij},\ \forall i \in \mathcal{E}_{el},\ \forall j \in \underline{\nu}(i),\ \forall r = 1, .., R_{ij}$

3. Solution mean values update using equation (1.5)

4. Detection process [DMP $\rightarrow u2$]:

If candidate $U_i^\star$ does not fulfill the Discrete Maximum Principle of equation (2.3) **then**

If it is not a smooth extrema according to the $u2$ detection criterion **then**

Decrement the CellPD d_i

end if

end if

end do

Figure 2.4: The detection process [DMP $\rightarrow u2$] for the convection equation.

Let us conclude this presentation by emphasizing that both DMP and $u2$ detection criteria are mesh-independent and intrinsically multidimensional which ensures flexibility and robustness of the concept. It also implies that an important notion is the neighborhood used in both criteria.

2.2.3 Numerical results

A substantial number of relevant numerical tests has been published in [18, 27] and shall be in [28]. Therefore we have chosen to reference all the numerical tests reproduced in chapter 3. Each reference is thus commented to emphasize the interest of the test case.

⋆ Convergence studies

We consider the classical convergence test in which the infinitely smooth initial function $U_0(x,y) = \sin(2\pi x)\sin(2\pi y)$ in 2D (resp. $U_0(x,y) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)$ in 3D) is convected with constant velocity V on the unit square $[0,1]^2$ (resp. unit cube $[0,1]^3$) with periodic boundary conditions. The final time $t_{final} = 2$ is such that the exact final solution corresponds to the initial one.

Section 3.1 or [18]: 2D Cartesian Meshes

Tables with L^1 and L^∞ errors and rates along with corresponding convergence curves figures are given for unlimited $\mathbb{P}1$ and $\mathbb{P}2$ schemes, the MOOD-$\mathbb{P}1$ and MOOD-$\mathbb{P}2$ methods and a MUSCL method. Note that the set of detection criteria $\mathcal{A}$ only contains the DMP and that the L^∞ order is locked to two.

→ Tables 3.3–3.5 p.68 and Figure 3.3 p.68 for a series of uniform Cartesian meshes.
→ Tables 3.6–3.8 p.72 and Figure 3.6 p.71 for a series of non-uniform Cartesian meshes.

Section 3.2 or [27]: 2D Polygonal Meshes

Tables with L^1 and L^∞ errors and rates along with corresponding convergence curves figures are given for the MOOD-$\mathbb{P}2$, MOOD-$\mathbb{P}3$ and MOOD-$\mathbb{P}5$ methods. The [DMP] and [DMP $\rightarrow u2$] detection processes are compared and it shows the second-order limit of the former while the latter provides up to 6^{th}-order convergence.

→ Table 3.11 p.101 and Figure 3.18 p.100 for series of Delaunay and Voronoi meshes.

Section 3.3 or [28]: 3D Mixed-element Meshes

Tables with L^1 and L^∞ errors and rates are given for the MOOD-$\mathbb{P}2$, MOOD-$\mathbb{P}3$ and MOOD-$\mathbb{P}5$ methods using the [DMP $\rightarrow u2$] detection process. Note that the version of the $u2$ is the one presented in this thesis and it enables to reach 6^{th}-order.

→ Table 3.17 p.139 and Figure 3.36 p.138 for series of hexahedral and mixed hexahedral-pyramidal meshes.

⋆ Non-oscillatory behavior assessment

In 2D, we consider the classical solid body rotation test in which three bodies (a slotted cylinder, a hump and a cone) are rotated around the center $(0.5, 0.5)$ of the unit square $[0,1]^2$. Periodic boundary conditions are still considered and the final time $t_{final} = 2\pi$ is such that the exact final solution corresponds to the initial one.

Section 3.1 or [18]: 2D Cartesian Meshes

We propose an 3D elevation view for the MOOD-$\mathbb{P}1$ method, MOOD-$\mathbb{P}2$ method and a MUSCL one. Note that the set of detection criteria $\mathcal{A}$ only contains the DMP and that the MOOD-$\mathbb{P}1$ results are better than the MUSCL ones tough both are second-order accurate.

→ Figure 3.4 p.69 for a uniform Cartesian meshes.
→ Figure 3.7 p.73 for a non-uniform Cartesian meshes.

Section 3.2 or [27]: 2D Polygonal Meshes

We propose profile views of the results with the unlimited $\mathbb{P}5$ method and the MOOD-$\mathbb{P}5$ method using [DMP] and [DMP $\rightarrow u2$] detection processes in order to show that the $u2$ maintains the accuracy of the unlimited scheme on smooth parts of the solution while it prevents the oscillations as well as the [DMP] on the discontinuous ones. Moreover a comparison between a MUSCL method and the MOOD-$\mathbb{P}1$, MOOD-$\mathbb{P}3$ and MOOD $\mathbb{P}5$ methods using [DMP $\rightarrow u2$] on isoline views of the slotted cylinder shows the improvement in using higher-order polynomials. All results are obtained on a non-regular Delaunay mesh mixing coarse and fine zones.

→ Figure 3.20 p.103 for the profile views.
→ Figure 3.21 p.104 for the isoline views.

In 3D, we propose a H-like shape rotation test detailed in [28], in which a discontinuous profile is rotated in the unit cube $[0,1]^3$ around the line going from the origin to $(1,1,1)$. Periodic boundary conditions are still considered and the final time $t_{final} = 2\pi$ is such that the exact final solution corresponds to the initial one.

Section 3.3 or [28]: 3D Mixed-element Meshes

Since it is a novel test, we propose several views of the initialization of the problem. A first one shows the H-like shape by means of an isosurface of value $1/2$ along with the axis of rotation. Then we propose to check the final result on the cut plane $z = 1/2$ in order to see if spurious oscillations are present. At last, we pot an 3D elevation view of the initial solution on this plane. The final solution is then presented for the unlimited $\mathbb{P}3$ and $\mathbb{P}5$ schemes and the MOOD-$\mathbb{P}3$ and MOOD-$\mathbb{P}5$ methods using [DMP $\rightarrow u2$] detection process, where we highlight in green cells which under/overshoot the exact solution.

→ Figure 3.37 p.140 for the complete initialization view.
→ Figure 3.38 p.141 for the final solution views.

2.3 Application to the Hydrodynamics Euler equations

2.3.1 Equations and Finite Volume scheme

In this section, we describe the application of the MOOD method to the Euler hydrodynamics system that governs the dynamics of non viscous gases. It is the non-linear hyperbolic system of conservation laws given by

$$\partial_t \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{pmatrix} + \partial_x \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{pmatrix} + \partial_y \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{pmatrix} + \partial_z \begin{pmatrix} \rho v \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{pmatrix} = 0. \tag{2.4}$$

We use the classical notation, that is ρ for density, u, v and w for velocity components in the x, y and z directions respectively, p for the pressure and E for the total energy. This system is closed by an Equation Of State (EOS) that relates two different thermodynamic variables; in this thesis, we only consider the perfect gas law, given by

$$p = (\gamma - 1)\rho\epsilon, \tag{2.5}$$

where ϵ is the specific internal energy and γ the ratio of specific heats, while the total energy writes

$$E = \rho\left(\frac{u^2 + v^2 + w^2}{2} + \epsilon\right).$$

We moreover recall that the conservative variables of the system are $U = (\rho, \rho u, \rho v, \rho w, E)$, while the primitive ones are (ρ, u, v, w, p). Finally as stated in sections 1.3.2 and 1.3.3, we shall not consider the characteristic variables to design the scheme and the polynomial reconstructions are always performed on conservative variables.

Remark: The use of a perfect gas law is not restrictive in the sense that the scheme we propose is independent of the EOS.

The complete numerical scheme is defined by equation (1.6) and equation (1.8), where the numerical flux $\mathbb{F}$ is defined by determining the exact or an approximate solution to a Riemann problem associated to the augmented one dimensional system in the direction normal to the face under consideration. This technique is possible thanks to the rotational invariance of the Euler equations. For all details concerning the resolution of the Riemann problem, the reader is referred to chapter 4 of [78] for an exhaustive overview. In our study, we shall employ two popular numerical fluxes. Considering two states U_L and U_R that are the rotated versions of $U^n_{ij,r}$ and $U^n_{ji,r}$, the Rusanov flux is given by

$$\mathbb{F}(U^n_{ij,r}, U_{ji,r},^n, \mathbf{n}_{ij}) = \frac{F(U_L) + F(U_R) - S_*(U_R - U_L)}{2}$$

where the wave speed estimate S_* is computed by

$$S_* = \max(|u_L - a_L|, |u_L + a_L|, |u_R - a_R|, |u_R + a_R|),$$

using the speed of sound $a_K = \sqrt{\frac{\gamma p_K}{\rho_K}}$ where K stands for L or R.

And the HLLC flux is defined by

$$\mathbb{F}(U^n_{ij,r}, U^{;}_{ji,r} n, \mathbf{n}_{ij}) = \begin{cases} F(U_L), & \text{if } 0 \leq S_L \\ F_*(U_L), & \text{if } S_L \leq 0 \leq S_* \\ F_*(U_R), & \text{if } S_* \leq 0 \leq S_R \\ F(U_R), & \text{if } 0 \geq S_R \end{cases},$$

where for K representing L or R, we have

$$F_*(U_K) = F(U_K) + S(U_K)(U_*(U_K) - U_K),$$

with

$$U_*(U_K) = \rho_K \left(\frac{S(U_K) - u_K}{S(U_K) - S_*} \right) \begin{pmatrix} 1 \\ S_* \\ v_K \\ w_K \\ \frac{E_K}{\rho_K} + (S_* - u_K)\left(S_* + \frac{p_K}{\rho_K (S_K - u_K)}\right) \end{pmatrix},$$

and the wave speed estimates are given by

$$\begin{aligned} S_L &= \min(u_L - a_L, u_R - a_R), \\ S_R &= \max(u_L + a_L, u_R + a_R), \\ S_* &= \frac{p_R - p_L + \rho_L u_L (S(U_L) - u_L) - \rho_R u_R (S(U_R) - u_R)}{\rho_L (S(U_L) - u_L) - \rho_R (S(U_R) - u_R)}. \end{aligned}$$

The boundary conditions are treated in a classical way. First let us remark that the FacePD of a boundary face is directly given by the CellPD since only one cell is connected to the face. Then for each quadrature point on a boundary face, we consider an inside evaluation computed from the polynomial on the connected cell and an outside value computed according to the boundary condition. More precisely:

- for inflow boundary conditions: we prescribe the imposed values on the outside;
- for reflective boundary conditions: we duplicate values of density, tangential momentum components and total energy while we take the opposite for normal momentum component;
- for outflow boundary conditions: we simply duplicate the inside values;
- for periodic boundary conditions: we refer the reader to section 2.2 for treatment details.

Finally we choose to consider only one CellPD per cell and two FacePD per face, that is to say all conservative variables are evaluated with the same polynomial degree. Though more refined adaptations could be envisaged, this proposition already provides good results for our applications.

2.3.2 Detection process

Contrary to the convection equation, numerous and stronger constraints apply to the variables of this system. In particular the density ρ, pressure p and internal energy ϵ must be positive.

The *a posteriori* essence of the method radically simplifies the way to enforce positivity of these variables. Indeed while most of existing multidimensional (*a priori*) higher-order methods do not guarantee the positivity-preserving property, we only have to put into the set of detection criteria $\mathcal{A}$ the conditions $\rho_i^\star > 0$ and $p_i^\star > 0$. In the sequel, such a detection process is called Physical Admissibility Detection (PAD) and the following theorem holds.

Theorem 2.8 *Let* $\{U_i^n\}_{i\in\mathcal{E}_{el}} = \left\{\rho_i^n, (\rho u)_i^n, (\rho v)_i^n, (\rho w)_i^n, E_i^n\right\}_{i\in\mathcal{E}_{el}}$ *be a positivity-preserving solution at time* t^n *and let* $\mathcal{A}$ *corresponds to the PAD, namely* $\rho_i^\star > 0$ *and* $p_i^\star > 0$. *If the FacePD strategy is upper-limiting and the first-order Godunov method is positivity-preserving, then the MOOD algorithm always converges and provides a solution at time* t^{n+1} *which satisfies* $\rho_i^{n+1} > 0$ *and* $p_i^{n+1} > 0$ *for all* $i \in \mathcal{E}_{el}$ *under the CFL condition of the first-order scheme (1.4).*

Proof: *We first recall that most of first-order Godunov schemes (1.4) provide a positivity-preserving solution under a CFL stability condition, thus this condition is not restrictive. Then theorem 2.5 implies that the candidate solution is always acceptable. Either the solution is* $\mathcal{A}$*-eligible,* i.e. *positivity-preserving, or the solution is computed from the first-order scheme which also provides a positivity-preserving solution. Consequently each candidate solution of the three steps of the RK3-TVD is positivity-preserving under the CFL of the first-order scheme and the final solution is positivity-preserving as a convex combination of the three candidate solutions.* □

Remark: Note that in the case of a perfect gas law the positivity of density and pressure implies the positivity of energy. In the case of other gas laws, we would simply include the positivity of energy in the set of criteria $\mathcal{A}$.

Remark: We emphasize that since we do not limit the reconstructed values *a priori*, they may be non-physical (*e.g.* negative density or total energy) at quadrature points. Consequently from a programming point of view, the calculated fluxes will be returned as a Not-a-Number (NaN) but we do treat this situation *a posteriori*. This is easily done by also checking if density or pressure is a NaN (*Not-a-Number*) via the function `ISNAN` in Fortran.

This important result highlights the potentiality of such an *a posteriori* concept. Nevertheless the PAD does not ensure a solution free of spurious oscillations and a complementary filter has to be designed. To this end, we remark that numerical non-physical oscillations always violate a Discrete Maximum Principle. Consequently the DMP may be relevant as second detection filter although the continuous solution of the Euler equations does not fulfill a maximum principle. Furthermore, we recall that the Euler system with constant pressure and velocity corresponds to a convection equation on the density. Therefore the $u2$ detection criterion is applied as a third filter to relax the DMP and recover the results of previous section.

Finally numerous numerical results have shown that the detection process [PAD $\rightarrow$ DMP $\rightarrow u2$] provides good results when the [DMP $\rightarrow u2$] part is only performed on the density. Indeed the detection and decrementing procedures of the MOOD method are decoupled, and it is relevant to perform the detection on only one variable that is affected by all types of waves, like density, since decrementing affects all variables.

0. Initialization of CellPD $\mathsf{d}_i = \mathsf{d}_{max},\ \forall i \in \mathcal{E}_{el}$
Do while (solution is not acceptable)
  1. Polynomial reconstruction of degree $\mathsf{d}_i,\ \forall i \in \mathcal{E}_{el}$
  2. Computation of FacePD d_{ij} and Evaluation of high-order approximations $U^n_{ij,r}$ at quadrature points $q_{ij,r}$ with degree $\mathsf{d}_{ij},\ \forall i \in \mathcal{E}_{el},\ \forall j \in \underline{\nu}(i),\ \forall r = 1, .., R_{ij}$
  3. Solution mean values update using equation (1.5)
  4. Detection process [PAD $\rightarrow$ DMP $\rightarrow u2$]:
    If candidate fulfills the Physical Admissibility Detection **then**
      If candidate $\rho_i^\star$ does not fulfill the DMP of equation (2.3) **then**
        If it is not a smooth extrema according to the $u2$ detection criterion **then**
          Decrement the CellPD d_i
        end if
      end if
    else
      Decrement the CellPD d_i
    end if
end do

Figure 2.5: The detection process for the Euler system [PAD $\rightarrow$ DMP $\rightarrow u2$].

An algorithm of the [PAD $\rightarrow$ DMP $\rightarrow u2$] detection process is given in Figure 2.5.

To conclude this section, we would like to underline some implementation details about the detection process. Actually in the above algorithm, the [PAD $\rightarrow$ DMP $\rightarrow u2$] performs well but does not completely provide very high-order on smooth solutions (say, the isentropic vortex in motion for instance, see next section for details). Deeper investigations show that the vortex profile is appropriately detected as smooth but that some decrementings occur on the flat part, *i.e.* plateaus, of the solution. This problem is due to the treatment of small *curvatures* in the $u2$ detection criterion which detects some spurious micro-oscillations as non smooth. More precisely, the *same sign* condition is not relevant when both *curvatures* are of the size of an epsilon. The following corrected definition of the $u2$ detection criterion fixes the problem.

Definition 2.9 (u2 detection criterion) *A candidate solution $U_i^\star$ in cell K_i for which the density $\rho_i^\star$ violates the DMP is nonetheless eligible if*

$$\mathcal{X}_i^{max}\mathcal{X}_i^{min} > -\delta \quad and \quad \left(\max\left(|\mathcal{X}_i^{max}|, |\mathcal{X}_i^{min}|\right) < \delta \quad or \quad \left|\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}}\right| \geq 1/2\right),$$

$$and \quad \mathcal{Y}_i^{max}\mathcal{Y}_i^{min} > -\delta \quad and \quad \left(\max\left(|\mathcal{Y}_i^{max}|, |\mathcal{Y}_i^{min}|\right) < \delta \quad or \quad \left|\frac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}}\right| \geq 1/2\right),$$

$$and \quad \mathcal{Z}_i^{max}\mathcal{Z}_i^{min} > -\delta \quad and \quad \left(\max\left(|\mathcal{Z}_i^{max}|, |\mathcal{Z}_i^{min}|\right) < \delta \quad or \quad \left|\frac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}}\right| \geq 1/2\right),$$

where δ is the longest one dimensional geometrical entity of cell K_i.

We would like to emphasize that the value of δ has been set following numerous test cases, and does not alter the capacity of the method to capture discontinuous profiles, while it is a necessary improvement to fully reach very high-order for the Euler system. It is worth noticing that this correction has even been tested for the convection equation without any loss in the quality of results. Furthermore we would like to point out that this modification only relaxes small oscillations. Indeed in the case when the product of minimal and maximal curvatures is satisfying left condition and maximal curvatures is big compared to δ (say $\mathcal{O}(1)$) the condition on the ratios of curvatures will ensure that the underlying function is considered as non regular.

In the same way, we slightly relax the DMP criteria to reduce the computational effort to avoid the waste of resources when performing the $u2$ detection criterion on plateaus. We consider that a DMP violation is not relevant if

$$\max_{j\in\overline{\nu}(i)}(\rho_i^{RK}, U_j^{RK}) - \min_{j\in\overline{\nu}(i)}(\rho_i^{RK}, U_j^{RK}) < \delta^3$$

where we use notation ^{RK} as in equation (2.3) and δ is set as in definition 2.9. Note that the purpose of this relaxation is to improve efficiency while maintaining the quality of results. It has been numerically verified for both the convection equations and the Euler system.

2.3.3 Numerical results

A substantial number of relevant numerical tests has been published in [18, 27] and shall be in [28]. Therefore we have chosen to reference all the numerical tests reproduced in chapter 3. Each reference is thus commented to emphasize the interest of the test case.

Results are presented in three complementary paragraphs. We refer to convergence test cases in the first paragraph while the second and third ones are dedicated to numerical results obtained with the [PAD] and [PAD $\rightarrow$ DMP $\rightarrow u2$] detection processes respectively. Note that comparisons with the [PAD $\rightarrow$ DMP $\rightarrow u2$] results always accompany the [PAD] ones. It thus supports the non-oscillatory behavior of the MOOD method using the [PAD $\rightarrow$ DMP $\rightarrow u2$].

$\star$ Convergence studies

There exist at least three popular convergence test cases that employ an exact solution to the Euler equations, namely the Ringleb flow, the steady isentropic vortex and the isentropic vortex in uniform motion. We do not consider the former since curved boundaries are mandatory, and we only run the vortex in motion since the second-order error generated by non-linear combinations of mean values (see section 1.3.2) does not affect the steady version. Details of the initialization can be found in [27] or [28].

Section 3.2 or [27]: 2D Polygonal Meshes

Tables with L^1 and L^∞ errors and rates along with corresponding convergence curves figures are given for the MOOD-$\mathbb{P}2$, MOOD-$\mathbb{P}3$ and MOOD-$\mathbb{P}5$ methods. Only the [PAD] detection processes is considered and effective higher-order is reached up to 6^{th}-order while a comparison is drawn between performing the reconstruction on the conservative variables and the primitive

ones in order to prove the claim of section 1.3.2. Moreover since the correction proposed in definition 2.9 was not investigated at that time, the [PAD] and [PAD $\rightarrow$ DMP $\rightarrow u2$] did not reach the optimal order of convergence.

$\rightarrow$ Table 3.14 p.106 and Figure 3.22 p.107 for a series of regular triangular meshes.

Section 3.3 or [28]: 3D Mixed-element Meshes

Tables with L^1 and L^∞ errors and rates are given for the MOOD-$\mathbb{P}2$, MOOD-$\mathbb{P}3$ and MOOD-$\mathbb{P}5$ methods using the [PAD $\rightarrow$ DMP $\rightarrow u2$] detection process given by definition 2.9. The correction brought to the $u2$ detection criteria enables to reach a full order of convergence up to 6^{th}-order.

$\rightarrow$ Table 3.18 p.147 and Figure 3.42 p.146 for a series of hexahedral and pyramidal meshes.

$\star$ Euler with [PAD]

In this part we reference results computed with the MOOD method using the [PAD] detection process. We recall that the [PAD] is only intended to ensure a physical solution and do not prevent spurious oscillations. Consequently the results presented below numerically assess the theorem 2.8 about the positivity-preserving property.

Section 3.2 or [27]: 2D Polygonal Meshes

Result to the *Double Mach reflection of a strong shock* problem is given for the MOOD-$\mathbb{P}5$ method on a mesh of 102,720 cells obtained by refinement of a coarser Delaunay one.

$\rightarrow$ Figure 3.25 p.112 provides global and zoomed views.

Result to the Noh test case is given for the MOOD-$\mathbb{P}3$ method on a circular polygonal mesh. Note that our implementation of the MUSCL method creates negative pressures and crashes.

$\rightarrow$ Figure 3.26 p.114 provides a 2D density-colored view and density *vs* radius one.

Section 3.3 or [28]: 3D Mixed-element Meshes

Result to the 3D explosion problem is given for the MOOD-$\mathbb{P}5$ method on a mesh of $[0,1]^3$ with 48,000 regular mesh of pyramids.

$\rightarrow$ Figure 3.47 p.152 for a density *vs* radius view.

$\star$ Euler with [PAD $\rightarrow$ DMP $\rightarrow \boldsymbol{u2}$]

In this part we reference results computed with the MOOD method using the [PAD $\rightarrow$ DMP $\rightarrow$ $u2$] detection process and should be compared to the state-of-the-art results. Some references are duplicated from the previous part and thus correspond to results for which a comparison between [PAD] and [PAD $\rightarrow$ DMP $\rightarrow u2$] is proposed to support the non-oscillatory property of the latter.

Section 3.1 or [18]: 2D Cartesian Meshes

Note that all results referred herein are obtained with the [DMP] detection process alone.

Results to the Sod shock tube are given for the MUSCL, MOOD-$\mathbb{P}1$ and MOOD-$\mathbb{P}2$ methods.

→ Figure 3.8 p.75 for a uniform $100 - 10$ Cartesian mesh.
→ Figure 3.9 p.77 for a non-uniform $100 - 10$ Cartesian mesh.

Results to a four-states Riemann Problem are given for the MUSCL, MOOD-$\mathbb{P}1$ and MOOD-$\mathbb{P}2$ methods. The classical isoline view and a 3D elevation one are shown.

→ Figure 3.10 p.78 for a uniform $400 - 400$ Cartesian mesh.

Results to the *Mach 3 wind with a step* problem are given for the MUSCL, MOOD-$\mathbb{P}1$ and MOOD-$\mathbb{P}2$ methods.

→ Figure 3.11 p.80 for complete isoline views on a uniform $120 - 40$ Cartesian mesh.
→ Figure 3.12 p.81 for complete isoline views on a uniform $480 - 160$ Cartesian mesh.

Results to the *Double Mach reflection of a strong shock* problem are given for the MUSCL, MOOD-$\mathbb{P}1$ and MOOD-$\mathbb{P}2$ methods.

→ Figure 3.13 p.82 for complete isoline views on a uniform $480 - 120$ Cartesian mesh.
→ Figure 3.14 p.83 for zoomed isoline views on uniform finer Cartesian meshes.

Section 3.2 or [27]: 2D Polygonal Meshes

Results to the Lax shock tube are given for the MOOD-$\mathbb{P}3$ method. We propose a comparison with the WENO4 method and show very good non-oscillatory behavior for the MOOD method.

→ Figure 3.23 p.108 for a uniform $100 - 10 - 2$ triangular mesh.

Results to the *Double Mach reflection of a strong shock* problem are given for the MOOD-$\mathbb{P}2$, MOOD-$\mathbb{P}3$ and MOOD-$\mathbb{P}5$ methods on a mesh of 102,720 cells obtained by refinement of an original coarse Delaunay one. We plot usual isolines views and give a computational costs comparison in terms of CPU and memory storage which shows the efficiency of the MOOD method.

→ Figure 3.24 p.110 for complete isoline views.
→ Figure 3.25 p.112 for a zoomed isoline view of the MOOD-$\mathbb{P}5$ result.
→ Table 3.16 p.111 for a computational costs comparison.

Result to the Noh test case is given for the MOOD-$\mathbb{P}3$ method on a circular polygonal mesh. Note that our implementation of the MUSCL method creates negative pressures and crashes.

→ Figure 3.26 p.114 provides a 2D density-colored view and density *vs* radius one.

At last, we propose a realistic test case of the impact of a shock on a cylindrical cavity extracted from [72]. The simulation is carried out on a non-regular polygonal mesh that contains degenerated cells. The result shows that the MOOD method is able to capture complex physics even on a second-rate mesh.

→ Figure 3.29 p.117 for top view at different times.
→ Figure 3.30 p.118 for zoomed views on instabilities.

Section 3.3 or [28]: 3D Mixed-element Meshes

Results to the Sod and Lax shock tubes are given for the MOOD-$\mathbb{P}3$ method on a tetrahedral mesh of a tube of unit length. The results shows that the MOOD method is able to capture simple waves without oscillations.

→ Figure 3.40 p.143 for density *vs* radius views.

Results to the Blastwave and Shu-Osher problem are given for the MOOD-$\mathbb{P}3$ method on a regular pyramidal mesh of a tube of unit length. The results show that the MOOD method is able to catch complex structures while preventing spurious oscillations on shocks and contact discontinuities.

→ Figure 3.41 p.144 for density *vs* radius views.

Result to the impact of a shock wave on a cylindrical cavity is given the MOOD-$\mathbb{P}_2$ method on a mesh made of triangular and quadrangular prisms. This test case proves that the MOOD method is able to simulate complex realistic physics on a mesh of prisms.

→ Figure 3.44 p.149 for density gradient views at several times.
→ Figure 3.45 p.150 for a zoom on created instabilities.

Results to the 3D explosion problem is given for the MUSCL, MOOD-$\mathbb{P}2$, MOOD-$\mathbb{P}3$ and MOOD-$\mathbb{P}5$ methods on a mesh of $[0,1]^3$ with 48,000 regular mesh of pyramids. The first figure gathers results of all methods and show the improvement of using higher-order polynomials, while in the second figure a comparison is drawn between the MOOD-$\mathbb{P}5$ method with [PAD] and with [PAD $\rightarrow$ DMP $\rightarrow u2$] which highlights the non-oscillatory improvement. Finally a comparison of computational costs in terms of CPU and memory storage is given in the last reference and shows that the MOOD method is very efficient.

→ Figure 3.46 p.151 for a density *vs* radius view with all methods results.
→ Figure 3.47 p.152 for an *all points* density *vs* radius view.
→ Table 3.19 p.153 for a computational costs comparison.

At last, result to the interaction of a shock wave with a quarter of cone is given for the MOOD-$\mathbb{P}_3$ method on a mesh of 1.1 million of tetrahedra to support the efficiency of the MOOD method on simple workstations.

→ Figure 3.48 p.155 for numerical Schlieren-type images on the $Ox-Oy$ and $Ox-Oz$ planes.
→ Figure 3.49 p.156 for a 3D plot of the main density isosurfaces.

2.4 Key optimizations

In this section, we first present two simple but mandatory optimizations to radically improve the efficiency of the presented MOOD method. Then we discuss the parallelization of the MOOD method and complete this chapter by providing in Figure 2.6 a detailed flowchart of the MOOD method as it is currently implemented in our 3D code.

⋆ Local re-updating

At first sight, the MOOD method seems computationally expensive because of the iterative nature of the MOOD algorithm since we have to repeat the spatial high-order scheme several times while polynomial degrees have been modified on only some cells. However we remark that in the high-order scheme (1.5), the time update of the solution mean value on a cell only involves the informations on the neighbors by face. Therefore, inside the MOOD algorithm, it is only necessary to recompute cells that have been detected and their neighbors by face. This is a drastic optimization since in most cases solution is acceptable for more than 80–90% of cells, even when solution contains shocks.

Practically speaking, the simplest implementation of this optimization is to use a table of one *logical* element per cell that contains the information on the acceptability of the cell, coupled with a simple *if* test at the beginning of the loop over cells in the MOOD algorithm and the same kind of technique may be used for a loop over faces if needed. This implementation is described in Figure 2.6 by the use of two tables, namely DetCell and DetFace, and has been used in our sequential code where its efficiency has been confirmed. Nevertheless it may have non-desired behavior in a massively parallelized code. Therefore a more parallel-like implementation would be to use tables of detected cells and detected faces, initialized with all cells and faces as detected. On the one hand it would demand an extra computational cost to fill the table of detected cells but on the other hand it would improve the management of the charge balance between nodes of the cluster.

⋆ Reduced polynomial degree decrementing

In the above presentation of the MOOD method, we did not specified how the decrementing is performed. The natural decrementing is to drop one-by-one polynomial degrees until zero is reached. However this may be costly in CPU time and even more in memory since reconstruction matrices would have to be stored for all degrees (it would nonetheless still be less memory consuming than for (W)ENO methods in most cases). Moreover in all the numerical tests we have observed that it is useless to consider so many different degrees since most of the time if the highest-order scheme fails, a discontinuity is to be expected. However it is worth testing one more degree than the highest one, especially because the size of the reconstruction stencil is much smaller for lower degrees (1 or 2).

Hence we advise to decrement from highest degree to degree 2 and then to degree 0, for at least two reasons: the symmetry brought by a degree 2 reconstruction (diffusive error), and because the quadratic reconstruction is used in the $u2$ detection processes presented in section 2.2 and 2.3. At the end, we only store two reconstruction matrices per cell, one for the maximal degree and one for the degree two. It is then important to note that already in 2D the memory

cost of a matrix for degree 2 (about 10 times 5 elements) is much smaller compared to a matrix of greater degree (about 16 times 9 for $\mathbb{P}_3$ and about 28 times 20 for $\mathbb{P}_5$). And it is even more relevant in 3D where the $\mathbb{P}_2$ reconstruction matrix represents about 16 times 9 elements while it is about 38 times 19 for $\mathbb{P}_3$ and 110 times 55 for $\mathbb{P}_5$.

⋆ Perspectives for parallelization

Considering today capabilities of many high-performance computers, the next major optimization that should be investigated is the parallelization of the MOOD method. There are currently three major parallelizations techniques: the OpenMP [1] for workstation with shared memory, the MPI [2] for massively parallel cluster with distributive memory and the GPGPU [3] to perform parallel operations on graphics cards or accelerators. It is moreover possible to combine them to obtain a hybrid parallelization.
Whichever type of parallelization we choose, the major purpose is to speed-up the simulations by a factor *close* to the number of processors we use. If so, some large simulations can be carried out in a reasonable time whereas they are not achievable on sequential machines. For instance a one-year sequential simulation might run in one day with a 400 cores cluster, or even in one hour with 10000 cores one. Nevertheless there is another interesting purpose to the massive parallelization on clusters using MPI. Indeed the memory storage is limited on a workstation (about 512 Go nowadays) while using the distributive memory of a cluster it becomes almost unlimited.

In the context of the MOOD method, these observations are still valid but the important savings in memory storage (compared to (W)ENO methods as instance) change the importance of the above remark. Indeed, as mentioned in [28] it is possible to run a 1 million cells Euler simulation with the fourth-order MOOD-$\mathbb{P}$3 method with *only* 16 Go of memory. Therefore a computation up to about 30 millions cells is possible on a single workstation. We believe that this is an important improvement in regard to existing very high-order Finite Volume methods.
Two important consequences result: the OpenMP parallelization is still suitable for realistic simulations since a today workstation can contain up to 64 cores and 512 Go of memory and these capacities are always growing; it makes realistic higher-order computations available to individuals and small companies since a complete up-to-date workstation costs less than ten thousands euros which is negligible compared to the millions of euros that a cluster costs (without taking the maintaining into account).

We have implemented an OpenMP version of our 3D code with about ten OpenMP directives and obtain a speed-up factor of two-to-three on a four-cores processor. However as stated in the first paragraph of this section, the method to deal with detected cells is not optimized for parallelization and may ruin the scalability on more than 4 cores. Furthermore it is noteworthy that the structure of the code has to be thought in advance to ensure an efficient parallelization, and this was not the case of our code. Going further in the reasoning, we can remark that our method is purely an unlimited one until the detection process is reached, and so there is no reason for this part of the code to cause more problems than for existing parallelized methods. In regard to the detection process, all informations needed in the detection criteria we proposed

1. Open Multi-Processing 2. Message Passing Interface 3. General-Purpose Processing on Graphics Processing Units

are local, *i.e.* restricted to the neighbors by nodes, which is very suitable for parallelization.

Finally the reconstruction process has to be efficiently parallelized since it is one of the main costly part of the code. A first remark is that requiring the very same number of neighbors for each reconstruction stencil may be an important point for the charge balance. A second point is proposed in appendix B where we first write each polynomial coefficient as a linear combination of the stencil mean values and discuss the way to obtain the curvatures for the $u2$ detection process. Then as an extension of the reasoning, we write the higher-order evaluations at a quadrature points as linear combinations of the stencil mean values. This may be an interesting way to merge the two steps (polynomial coefficients computation and evaluation of polynomial at quadrature points) currently used and consequently improve the scalability.

In the near future, the parallelization of the MOOD method shall complete the proof of efficiency of the MOOD method.

Conclusion

In the first part of this chapter, we have developed a complete mathematical framework to handle the *a posteriori* essence of the MOOD method. Then, the application of the concept to the linear convection equation and the non-linear Euler equations has been carefully detailed and supported by a list of significant published (or to-be-published) numerical tests. At last, we have provided simple optimizations that make the MOOD method one of the most efficient very high-order Finite Volume method currently available.

Throughout the chapter, we have pointed out the novelty of the approach and particularly the simplicity of the method to enforce strong constraints independently of the mesh and the domain spatial dimension. For instance the solution can be constrained in prescribed bounds (positivity of density, mass or volume fraction bounded by 0 and 1, etc.) if the first-order scheme ensures this property. This is of crucial importance to ensure the robustness of the method when applied to complex realistic simulations.

Apart from the *a posteriori* approach, two other original ideas have contributed to guarantee the quality of results at the state-of-the-art standards level. The first one is the $u2$ detection criteria which provides a definition to a *numerically smooth* function in our context. It allows to overcome the second-order limitation of a strict application of the DMP on mean values, and thus to reach arbitrary high-order of convergence. Up to our knowledge, such a definition has never been proposed before. The second crucial idea, that applies in the case of vectorial problems, is to separate the variables on which the detection is performed from the ones that are limited. Basically, this is workable thanks to the *a posteriori* treatment which intrinsically separates the detection process from the decrementing one.

Finally we believe that the MOOD method has been proved to be a viable and efficient alternative to existing very high-order Finite Volume methods and moreover to ensure appreciable properties (DMP, positivity-preserving, etc.) with an outstanding easiness.

The next chapter is dedicated to a review of the MOOD method design for the hydrodynamics Euler equations through the three journal publications. We give a short summary along with a review of the problems we encountered and the ones we corrected in order for the reader to understand the final design of the *a posteriori* MOOD method.

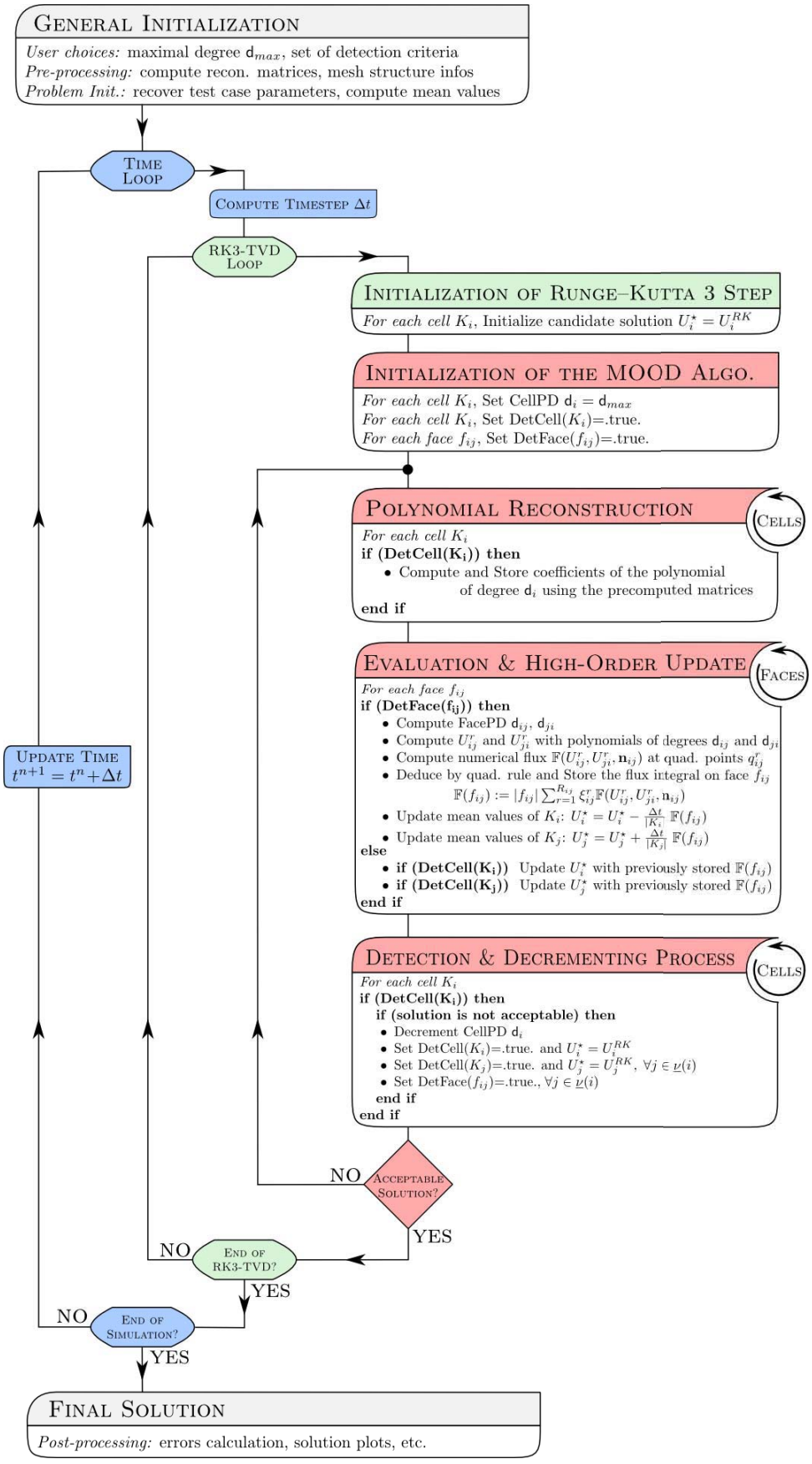


Figure 2.6: Complete flowchart of the MOOD method.

Chapter 3

Towards the MOOD method for the Hydrodynamics Euler System.

In chapters 1 and 2, we have presented an up-to-date version of the MOOD method for the hydrodynamics Euler equations. Naturally, the design has involved numerous stages during the three years of this doctorate leading to the suitable final ingredients. Through the history of the three main publications, we intend in this chapter to help the reader to figure out the details of the MOOD method.

This chapter is divided in three sections where we reproduce the two published articles [18, 27] and the submitted one [28] in order of appearance. Moreover, all papers reproduced hereafter are preceded by a *summary & review* paragraph in which we recap the main ideas contained in the paper and propose a retrospective review pointing out the problems that were not settled at the time of the paper. This way, we intend to enlighten the reasons for the final concepts of the MOOD method.

3.1 Part I: 3$^{\text{rd}}$-order accuracy on 2D Cartesian meshes

This section is dedicated to the first publication introducing the MOOD method. The reference is:

S. Clain, S. Diot, R. Loubère, *A high-order finite volume method for systems of conservation laws – Multi-dimensional Optimal Order Detection (MOOD)*, J. Comput. Phys. 230 (2011) 4028–4050.

In next paragraph, we sum up the content of the publication and highlight with hindsight the pros and cons of the MOOD method at that time. We then reproduce the paper from the abstract to the conclusion only correcting the misprints and modifying the references to fit the global bibliography.

Summary & Review

The article was the first step toward the *a posteriori* concept presented in this thesis. Most of the ingredients we developed in chapter 2 were introduced: genuinely multidimensional

polynomial reconstruction, CellPD, FacePD (EdgePD since it was 2D), *a posteriori* detection process, decrementing procedure. Although the study was restricted to Cartesian meshes (but not necessarily uniform) and to linear and quadratic reconstructions, the main characteristics of the MOOD method were already present in the numerical tests. Actually, the second-order MOOD-$\mathbb{P}1$ method performed better than the classical MUSCL method on non-uniform meshes for an equivalent cost when convection equation and Euler system are considered. Furthermore the MOOD-$\mathbb{P}2$ results showed promising improvements in the quality of the solution, and a very good aptitude of the MOOD method to capture discontinuities without spurious oscillations for the hydrodynamics Euler equations. At last, a brief cost comparison between the MUSCL method and the MOOD-$\mathbb{P}1$ and MOOD-$\mathbb{P}2$ ones demonstrated that the proposed method had strong basis to be efficient.

These investigations led this paper to be a first proof of efficient feasibility of an *a posteriori* concept for limitation. Nevertheless we would like to highlight four points in this paper that needed to be treated.

The first one deals with the polynomial reconstruction. Contrary to what we have proposed in chapter 2, we did not use one polynomial reconstruction per degree but a truncation of the maximal degree one in order to obtain the lower degree reconstructions. This technique is valid and ensures the right order of convergence, but always involves the large stencil of the maximal polynomial degree. We discovered that using a lower-degree reconstruction with a smaller stencil produces better approximations of discontinuities.

The second remark comes from a question that has arisen at the HONOM2011 conference. Actually in this first publication, when solving the Euler equations, we checked if the quadrature points evaluations of the polynomials were physical or not, and replaced the high-order approximations by the mean value if not. As suggested during the after-talk questions session, this may be seen as an *a priori* limitation. Consequently a simple modification based on the NaN convention fixed this problem and the method has become totally *a posteriori*.

The next point concerns the use of the strict DMP on mean values alone in the detection process, that locks the scheme to second-order in L^∞ norm (and third-order in L^1 norm). We have observed this limitation when carrying out the Double Sine Translation test case and knew that the DMP was responsible for this accuracy discrepancy. Actually even in the second-order MUSCL method, a full second L^∞ rate of convergence is never reached because the limiting principle is to ensure a DMP on mean values. However in the case of the hydrodynamics Euler equations, the second-order lock had an additional cause that was sharper and is the purpose of the last point.

At the time of the paper, we were new comers in the field of very high-order Finite Volume methods and discovered that performing the polynomial reconstruction from primitive variables mean values (in the Euler context), also creates a second-order error. We would like to emphasize that this point is mostly not stated in the literature and that there exist *very high-order* Finite Volume schemes based on primitive variables. It is even more subtle since this second-order error does not occur for at least two classical test cases for convergence, namely the Ringleb flow problem and the steady isentropic vortex. Note that these two last points appear in the section 1.3.2.

Abstract

In this paper, we investigate an original way to deal with the problems generated by the limitation process of high-order finite volume methods based on polynomial reconstructions. Multi-dimensional Optimal Order Detection (MOOD) breaks away from classical limitations employed in high-order methods. The proposed method consists of detecting problematic situations after each time update of the solution and of reducing the local polynomial degree before recomputing the solution. As multi-dimensional MUSCL methods, the concept is simple and independent of mesh structure. Moreover MOOD is able to take physical constraints such as density and pressure positivity into account through an "a posteriori" detection. Numerical results on classical and demanding test cases for advection and Euler system are presented on quadrangular meshes to support the promising potential of this approach.

3.1.1 Introduction

High-order methods for systems of nonlinear conservation laws are an important challenging question with a wide range of applications. Furthermore in an engineering context such methods may deal with complex multi-dimensional domains requiring unstructured, heterogeneous or even non-conformal meshes. To handle highly stretched unstructured meshes made with different cell shapes, one has to design genuinely multi-dimensional numerical methods which exclude dimensional splitting techniques.

Due to its simplicity (one unknown mean value per cell) and built-in conservativity property, first-order finite volume method is very popular in today's engineering applications or commercial codes. However, it suffers from a major drawback, namely the presence of a large amount of numerical diffusion leading to a poor accuracy and over smoothed discontinuities. High-order space and time finite volume methods based on local polynomial reconstructions and Runge-Kunta algorithm have been developed to improve the approximation accuracy. MUSCL methods are probably the most popular second-order finite volume schemes. First developed in the one-dimensional situation with linear reconstructions [48, 84, 49, 50], the technique has been extended to genuinely multi-dimensional case using structured or unstructured meshes [8, 7, 42, 94, 20, 62, 13]. Stability is achieved using a limiting procedure based on the Maximum Principle. In the present study, the Multi-dimensional Limiting Process (MLP) of [94, 62] is employed since it is one of the most up-to-date MUSCL methods. Besides, (Weighted) Essentially Non Oscillatory polynomial reconstruction procedures (ENO/WENO) were designed to reach higher-order of accuracy [39, 40, 1, 70, 69] using less restrictive conditions for the limitation which do not guarantee a strict Maximum Principle for scalar problems. Moreover, although ENO/WENO schemes can retain high-order spatial accuracy even at points of extrema, extra difficulties and complexities have to be faced for the implementation on multi-dimensional unstructured grids (see [1, 92]) as a large number of stencils for the polynomial reconstructions must be proceeded. Such drawbacks lead us to put ENO/WENO methods aside from the present study.

In this work we propose a genuinely multi-dimensional high-order method within a finite volume Eulerian framework on non-uniform meshes, the Multi-dimensional Optimal Order Detection (MOOD) method. In contrast to the traditional methods which use an *a priori* limitation procedure, the MOOD technique is based on an *a posteriori* detection of problematic cells.

In each cell optimal polynomial degrees are determined to build approximated states leading to a discrete maximum principle preserving solution. In an hydrodynamics context, physical properties such as the density and the pressure positivity are considered. Roughly speaking, the polynomial degree may drop to zero in the vicinity of discontinuities leading to a local stable first-order finite volume scheme whereas high-order scheme is achieved in smooth regions. As for other methods, the MOOD method is embedded into the sub-steps of a high-order Runge-Kutta time discretization.

The paper is organized as follows. Section 3.1.2 is dedicated to the generic framework used to describe the MOOD method. Section 3.1.3 is devoted to the linear reconstruction and to a short presentation of the MLP method [94, 62]. The MOOD method for scalar problems is detailed in the fourth section while section 3.1.5 is dedicated to an extension of MOOD method to the Euler equations. At last, the numerical results for the advection and the Euler equations problems are respectively gathered in sections 3.1.6 and 3.1.7. Classical tests are carried out and comparisons to the results of MLP method are provided. Several numerical examples prove the efficiency of the MOOD method in its second- and third-order version. The last section finally gathers conclusion and perspectives.

3.1.2 General framework

We consider the generic scalar hyperbolic equation defined on a domain $\Omega \subset \mathbb{R}^2$, $t > 0$ cast in the conservative form

$$\begin{aligned} \partial_t u + \nabla \cdot F(u) &= 0, && \text{(3.1a)} \\ u(\cdot,0) &= u_0, && \text{(3.1b)} \end{aligned}$$

where $u = u(\mathbf{x}, t)$ is the unknown function, $\mathbf{x} = (x_1, x_2)$ denotes a point of Ω and t the time. F is the physical flux and u_0 is the initial condition. Boundary conditions shall be prescribed in the following.

To elaborate the discretization in space and time, we introduce the following ingredients. We assume that the computation domain Ω is a polygonal bounded set of $\mathbb{R}^2$ divided into quadrangles K_i, $i \in \mathcal{E}_{el}$ where $\mathcal{E}_{el}$ is the cell index set with $\mathbf{c_i}$ being the cell centroid. For each cell K_i, $\lambda(i)$ is the set of all the nodes P_m, $m \in \lambda(i)$ while e_{ij} denotes the common edge between K_i and K_j with $j \in \underline{\nu}(i)$, $\underline{\nu}(i)$ being the index set of all the elements which share a common side with K_i. Moreover, $\overline{\nu}(i)$ represents the index set of all K_j such that $K_i \cap K_j \neq \emptyset$ (see figure 3.1). At last, $|K_i|$ and $|e_{ij}|$ measure the surface of K_i and the length of e_{ij} respectively and $\mathbf{n}_{ij}$ is the unit outward normal vector of K_i.

To compute an approximation of the solution of equation (3.1), we recall the generic first-order explicit finite volume scheme

$$u_i^{n+1} = u_i^n - \Delta t \sum_{j \in \underline{\nu}(i)} \frac{|e_{ij}|}{|K_i|} \mathbb{F}(u_i^n, u_j^n, \mathbf{n}_{ij}), \tag{3.2}$$

where $\mathbb{F}(u_i^n, u_j^n, \mathbf{n}_{ij})$ is a numerical flux which satisfies the classical properties of consistency and monotonicity.

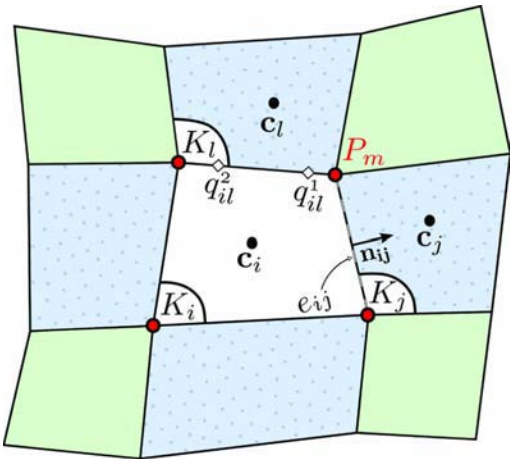


Figure 3.1: Mesh notation. K_i is a generic element with the centroid $\mathbf{c_i}$. Index set $\underline{\nu}(i)$ corresponds to blue cells with dots, $\overline{\nu}(i)$ corresponds to every non-white cells and $\lambda(i)$ is the set of red P_m node indexes. Edges are denoted by e_{ij} with $\mathbf{n}_{ij}$ the unit outward normal vector of element K_i. Numerical integration on edge e_{il} is performed with the two Gauss points q^1_{il}, q^2_{il}.

Unfortunately, such a scheme only provides first-order accuracy in space and higher-order reconstruction techniques are used to improve the solution approximation. To this end, we substitute in equation (3.2) the first-order approximation u_i^n and u_j^n with better approximations of u on the e_{ij} edge and consider the generic spatial high-order finite volume scheme

$$u_i^{n+1} = u_i^n - \Delta t \sum_{j \in \underline{\nu}(i)} \frac{|e_{ij}|}{|K_i|} \sum_{r=1}^{R} \xi_r \mathbb{F}(u_{ij,r}^n, u_{ji,r}^n, \mathbf{n}_{ij}), \tag{3.3}$$

where $u_{ij,r}^n$ and $u_{ji,r}^n$, $r = 1, ..., R$ are high-order representations of u on both sides of edge e_{ij} and ξ_r denote the quadrature weights for the numerical integration. In practice, $u_{ij,r}^n$ and $u_{ji,r}^n$ are two approximations of $u(q_{ij}^r, t^n)$ at quadrature points $q_{ij}^r \in e_{ij}$, $r = 1, ..., R$ (see figure 3.1).

For the sake of simplicity, let us write the scheme under the compact form

$$u_h^{n+1} = u_h^n + \Delta t \, \mathcal{H}^R(u_h^n), \tag{3.4}$$

with $u_h^n = \sum_{i \in \mathcal{E}_{el}} u_i^n \mathbb{1}_{K_i}$ the constant piecewise approximation of function u and operator $\mathcal{H}^R$ being defined as

$$\mathcal{H}^R(u_h^n) := \sum_{i \in \mathcal{E}_{el}} \left(- \sum_{j \in \underline{\nu}(i)} \frac{|e_{ij}|}{|K_i|} \sum_{r=1}^{R} \xi_r \mathbb{F}(u_{ij,r}^n, u_{ji,r}^n, \mathbf{n}_{ij}) \right) \mathbb{1}_{K_i}. \tag{3.5}$$

To provide a high-order method in time, we use the third-order TVD Runge-Kutta method

(see [70]) which corresponds to a convex combination of three explicit steps

$$u_h^{(1)} = u_h^n + \Delta t\ \mathcal{H}^R\left(u_h^n\right), \tag{3.6a}$$
$$u_h^{(2)} = u_h^{(1)} + \Delta t\ \mathcal{H}^R(u_h^{(1)}), \tag{3.6b}$$
$$u_h^{(3)} = \left(\frac{3u_h^n + u_h^{(2)}}{4}\right) + \Delta t\ \mathcal{H}^R\left(\frac{3u_h^n + u_h^{(2)}}{4}\right), \tag{3.6c}$$
$$u_h^{n+1} = \frac{u_h^n + 2u_h^{(3)}}{3}. \tag{3.6d}$$

Remark 3.1 *Note that a high-order scheme in space and time can be rewritten as convex combinations of the first-order scheme. From a practical point of view, implementation of the high-order scheme from an initial first-order scheme is then straightforward.* □

The main challenge is to build the approximations $u_{ij,r}^n$ and $u_{ji,r}^n$ on both sides of edge e_{ij} with $r = 1, ..., R$ to be plugged into relations (3.5) and (3.6). Polynomial reconstructions provide high-order approximations but unphysical oscillations arise in the vicinity of discontinuities. Indeed, the exact solution of an autonomous scalar conservation law (3.1) satisfies a local Maximum Principle and we intend to build the reconstructions such that this stability property is fulfilled at the numerical level (see [16, 17] and references herein). To this end, we state the following definition.

Definition 3.2 *A numerical scheme (3.4) satisfies the Discrete Maximum Principle (DMP) if for any cell index $i \in \mathcal{E}_{el}$ one has*

$$\min_{j\in\overline{\nu}(i)}\left(u_i^n, u_j^n\right) \leq u_i^{n+1} \leq \max_{j\in\overline{\nu}(i)}\left(u_i^n, u_j^n\right). \tag{3.7}$$

3.1.3 A short review on a multi-dimensional MUSCL method

All L^∞ stable second-order schemes are based on piecewise linear reconstructions equipped with a limiting procedure. The polynomial reconstruction provides the accuracy while the limitation algorithm ensures the physical relevancy of the numerical approximation. We briefly present the piecewise linear reconstruction step and recall the MLP method proposed in [62] which is used in the numerical part of this paper.

3.1.3.1 Linear reconstruction

Let $(u_i)_{i\in\mathcal{E}_{el}}$ be a set of cell centered mean values given on cells K_i. In order to simplify notations, let K be a generic cell with centroid $\mathbf{c} = (c_1, c_2)$. Considering mean values on a chosen neighborhood made of cells K_j, $j \in \nu$, we seek a polynomial function $\widetilde{u}(\mathbf{x})$ of degree $d = 1$. Let us define the notation for the mean value as

$$\left\langle \widetilde{u}(\mathbf{x})\right\rangle_K \overset{\text{def}}{=} \frac{1}{|K|}\int_K \widetilde{u}(\mathbf{x})d\mathbf{x}.$$

Usually we ask for the following criteria

Criterion 3.3 *The polynomial reconstruction $\widetilde{u}$ must fulfill*

1. $\left\langle \widetilde{u}(\mathbf{x}) \right\rangle_K = \bar{u}$ *where* $\bar{u}$ *is the mean value approximation of* u *on* K.
2. *The polynomial coefficients are the ones minimizing the functional*

$$E(\widetilde{u}) = \sum_{j \in \nu} \left(u_j - \left\langle \widetilde{u}(\mathbf{x}) \right\rangle_{K_j} \right)^2, \tag{3.8}$$

A classic way to write $\widetilde{u}$ is

$$\widetilde{u}(\mathbf{x}) = \bar{u} + G \cdot (\mathbf{x} - \mathbf{c}), \tag{3.9}$$

where $G = (G_1, G_2)$ is a constant approximation of ∇u on K. The first condition of criterion 3.3 is directly satisfied and classical techniques like least squares methods are used to determine vector G that minimizes the functional E in equation (3.8) .

3.1.3.2 Gradient limitation

As we mentioned above, a finite volume scheme only based on a local polynomial reconstruction without limiting procedure produces spurious oscillations. Initiated by the pioneer works of Kolgan and Van Leer [48, 49, 84, 50], the MUSCL technique deals with a local linear reconstruction like (3.9) on each cell K where the gradient G is reduced by a limiter coefficient $\phi \in [0, 1]$

$$\widetilde{u}(\mathbf{x}) = \bar{u} + \phi \ \left(G \cdot (\mathbf{x} - \mathbf{c}) \right). \tag{3.10}$$

such that any reconstructed values satisfy the Discrete Maximum Principle (see [7, 8, 42]). We choose to detail and use the MLP limiter instead of the classical Barth-Jespersen limiter because it provides more accurate results (see [62]). The MLP limiter applies the following procedure.

- Construction of an unlimited slope G using the neighbor cells K_j, $j \in \overline{\nu}$.
- Evaluation of the unlimited reconstruction (3.9) at the vertices P_m of K: $u_m = \widetilde{u}(P_m)$, $m \in \lambda$ the nodes index set of K.
- Evaluation of the bounds for each node P_m

$$\delta u_m^{\max} = \max_{j,\, P_m \in \lambda(j)} (u_j - \bar{u}), \quad \delta u_m^{\min} = \min_{j,\, P_m \in \lambda(j)} (u_j - \bar{u}).$$

- Evaluation of the vertex based limiter ϕ_m

$$\phi_m = \begin{cases} \min\left(1, \dfrac{\delta u_m^{\max}}{u_m - \bar{u}}\right) & \text{if } u_m - \bar{u} > 0, \\ \min\left(1, \dfrac{\delta u_m^{\min}}{u_m - \bar{u}}\right) & \text{if } u_m - \bar{u} < 0, \\ 1 & \text{if } u_m - \bar{u} = 0. \end{cases}$$

- Cell-centered limiter $\phi = \min_{m \in \lambda} \phi_m$.

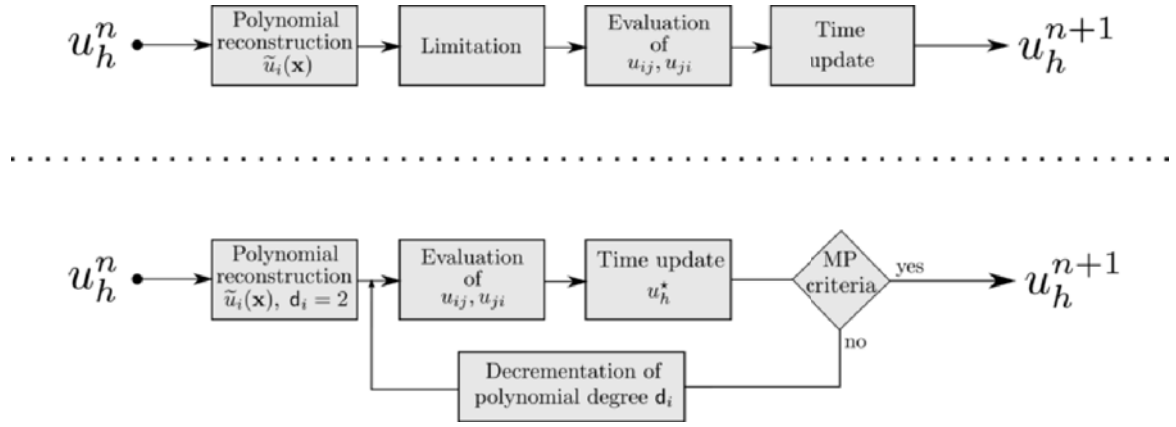


Figure 3.2: Classical high-order methods idea (top) and MOOD idea (bottom).

The MLP technique provides a second-order finite volume scheme which satisfies the Discrete Maximum Principle under a more restrictive CFL condition than the CFL condition of the first-order scheme.

Remark 3.4 *Although there exists a large literature about piecewise linear limitation, the extension of MUSCL type methods to piecewise quadratic or even higher degree polynomials in a multi-dimensional context is not yet achieved. An efficient limitation process is still an under-investigation field of research.*

□

3.1.4 The Multi-dimensional Optimal Order Detection method (MOOD)

Classical high-order methods are based on an *a priori* limitation of the reconstructed values which are plugged into a one time step generic finite volume scheme to update the mean values (see figure 3.2 top).
Unlike existing methods, the MOOD technique proceeds with an *a posteriori* limitation. Over each cell, an unlimited polynomial reconstruction is carried out to build a prediction $u_h^\star$ of the updated solution. Then the *a posteriori* limitation consists of reducing the polynomial degree and recomputing the predicted solution $u_h^\star$ until the DMP property (3.7) is achieved. To this end, a prescribed maximum degree $d_{\max}$ is introduced and used to perform an initial polynomial reconstruction on each cell. Through an iterative decremental procedure, we determine the *optimal degree* $\mathsf{d}_i \leq d_{\max}$ on each cell K_i such that each updated mean value $u_i^\star$ fulfills the DMP property (see figure 3.2 bottom).

In the following we focus on the quadratic polynomial case $d_{\max} = 2$ and first present the local quadratic reconstruction of [57]. Then the MOOD method is detailed and we prove that the numerical approximations satisfy the DMP property.

3.1.4.1 Quadratic reconstruction

Using the same framework as in section 3.1.3.1, the quadratic polynomial reconstruction is written

$$\widetilde{u}(\mathbf{x}) = \bar{u} + G \cdot(\mathbf{x}-\mathbf{c}) + \frac{1}{2}\Big((\mathbf{x}-\mathbf{c})^t H(\mathbf{x}-\mathbf{c}) - \bar{H}\Big), \tag{3.11}$$

with

$$\bar{H} = \Big\langle (\mathbf{x}-\mathbf{c})^t H(\mathbf{x}-\mathbf{c}) \Big\rangle_K, \qquad H = \begin{bmatrix} H_{11} & H_{12} \\ H_{12} & H_{22} \end{bmatrix},$$

where matrix H is an approximation of the Hessian matrix $\nabla^2 u$ on K. Note that by construction, the mean value of $\widetilde{u}$ on K is still equal to $\bar{u}$.
A minimization technique is used to compute G and H. To this end, for a cell K_j, let us define the integrals

$$\mathbf{x}_{K_j}^{\{\alpha,\beta\}} = \Big\langle (x-c_1)^\alpha (y-c_2)^\beta \Big\rangle_{K_j} - \Big\langle (x-c_1)^\alpha (y-c_2)^\beta \Big\rangle_K.$$

Algebraic manipulations yield the following expression for $\Big\langle \widetilde{u}(\mathbf{x}) \Big\rangle_{K_j}$

$$\Big\langle \widetilde{u}(\mathbf{x}) \Big\rangle_{K_j} = \bar{u} + \Big(G_1\ \mathbf{x}_{K_j}^{\{1,0\}} + G_2\ \mathbf{x}_{K_j}^{\{0,1\}}\Big) + \frac{1}{2}\Big(H_{11}\mathbf{x}_{K_j}^{\{2,0\}} + 2H_{12}\mathbf{x}_{K_j}^{\{1,1\}} + H_{22}\mathbf{x}_{K_j}^{\{0,2\}}\Big). \tag{3.12}$$

This expression is further derived for any cell K_j with $j \in \nu$ to form an over-determined linear system of the form $A\Lambda = B$ with

$$A = \begin{pmatrix} \mathbf{x}_{K_1}^{\{1,0\}} & \mathbf{x}_{K_1}^{\{0,1\}} & \mathbf{x}_{K_1}^{\{2,0\}} & \mathbf{x}_{K_1}^{\{1,1\}} & \mathbf{x}_{K_1}^{\{0,2\}} \\ \mathbf{x}_{K_2}^{\{1,0\}} & \mathbf{x}_{K_2}^{\{0,1\}} & \mathbf{x}_{K_2}^{\{2,0\}} & \mathbf{x}_{K_2}^{\{1,1\}} & \mathbf{x}_{K_2}^{\{0,2\}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{x}_{K_N}^{\{1,0\}} & \mathbf{x}_{K_N}^{\{0,1\}} & \mathbf{x}_{K_N}^{\{2,0\}} & \mathbf{x}_{K_N}^{\{1,1\}} & \mathbf{x}_{K_N}^{\{0,2\}} \end{pmatrix},\ \Lambda = \begin{pmatrix} G_1 \\ G_2 \\ \frac{1}{2}H_{11} \\ H_{12} \\ \frac{1}{2}H_{22} \end{pmatrix},\ B = \begin{pmatrix} u_1 - \bar{u} \\ u_2 - \bar{u} \\ \vdots \\ u_N - \bar{u} \end{pmatrix}, \tag{3.13}$$

with $N = \#\nu$. This system is solved with a QR decomposition of A using Householder transformations, such that $Q \in \mathcal{M}_{N-N}(\mathbb{R})$ is an orthogonal matrix and $R \in \mathcal{M}_{N-5}(\mathbb{R})$ an upper-triangular one. Finally back-substitution of $R\Lambda = Q^t B$ defines $\widetilde{u}$ (see [57]).

Remark 3.5 *A left preconditioner matrix can be applied to reduce the system sensitivity and improve the reconstruction quality. For example, in [57], the authors use a diagonal matrix whose coefficients $\omega_{jj} = \|\mathbf{c_j} - \mathbf{c}\|^{-2}$ $(j = 1, \ldots, N)$ correspond to geometrical weights in order to promote closest informations.* □

3.1.4.2 Description of the MOOD method

We now detail the MOOD technique considering the simple case where an explicit time discretization is employed. Moreover, without loss of generality, we present the method using only one quadrature point ($R = 1$) and skip the subscript r denoting u_{ij} in place of $u_{ij,r}$. Extension to several quadrature points ($R > 1$) is straightforward.

Assume that we have a given sequence $u_h^n = (u_i^n)_{i\in\mathcal{E}_{el}}$ of mean value approximations at time t^n, the goal is to build a relevant sequence $u_h^{n+1} = (u_i^{n+1})_{i\in\mathcal{E}_{el}}$ at time $t^{n+1} = t^n + \Delta t$. To this end, we define the following fundamental notions.

- d_i is the Cell Polynomial Degree (CellPD) which represents the degree of the polynomial reconstruction on cell K_i.
- d_{ij} and d_{ji} are the Edge Polynomial Degrees (EdgePD) which correspond to the effective degrees used to respectively build u_{ij} and u_{ji} on both sides of edge e_{ij}.

The MOOD method consists of the following iterative procedure.

---

1. **CellPD initialization.** Each CellPD is initialized to d_{max}.
2. **EdgePD evaluation.** Each EdgePD is set up as a function of the neighboring CellPD (see table 3.1).
3. **Quadrature points evaluation.** Each u_{ij} is evaluated with the polynomial reconstruction of degree d_{ij}.
4. **Mean values update.** The updated values $u_h^\star$ are computed using the finite volume scheme (3.3).
5. **DMP test.** The DMP criterion is checked on each cell K_i

$$\min_{j\in\overline{\nu}(i)}(u_i^n, u_j^n) \leq u_i^\star \leq \max_{j\in\overline{\nu}(i)}(u_i^n, u_j^n). \tag{3.14}$$

   If $u_i^\star$ does not satisfy (3.14) the CellPD is decremented, $\mathsf{d}_i := \max(0, \mathsf{d}_i - 1)$.

6. **Stopping criterion.** If all cells satisfy the DMP property, the iterative procedure stops with $u_h^{n+1} = u_h^\star$ else go to step 2.

We give in table 3.1 three possible strategies of EdgePD calculation. The simplest one named $\mathsf{EPD_0}$ consists of setting $\mathsf{d}_{ij} = \mathsf{d}_i$ and $\mathsf{d}_{ji} = \mathsf{d}_j$ whereas $\mathsf{EPD_1}$ chooses the minimal value between d_i and d_j for both d_{ij} and d_{ji}. At last, the smallest CellPD of all the direct neighbor cells is taken in the $\mathsf{EPD_2}$ strategy.

To conclude the section, there are two important remarks which dramatically reduce the computational cost.

Remark 3.6 *If $\mathsf{d}_{ij} < d_{max}$, there is no need to recompute a polynomial of degree d_{ij}, a simple truncation of the initial polynomial of degree d_{max} should be performed.* □

Remark 3.7 *Only cells K_i where CellPD has been decremented and their neighbors in a compact stencil have to be updated. Consequently only these cells have to be checked during next iterations of the MOOD procedure in the current time step. For instance the compact stencil for $\mathsf{EPD_0}$ and $\mathsf{EPD_1}$ is $\underline{\nu}(i)$ while for $\mathsf{EPD_2}$ it is $\big\{\underline{\nu}(i) \cup \{\underline{\nu}(j), j \in \underline{\nu}(i)\}\big\}$.* □

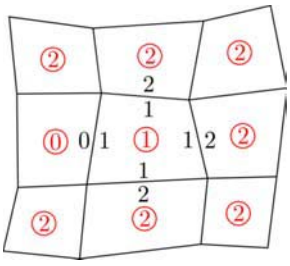
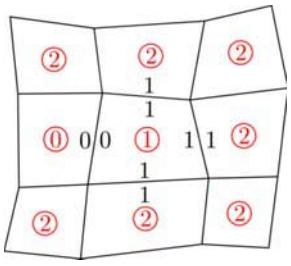
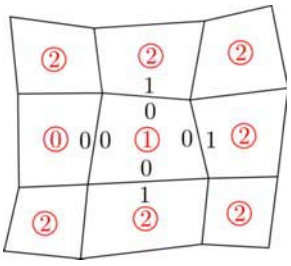
	EPD_0 **strategy**	EPD_1 **strategy**	EPD_2 **strategy**
EdgePD d_{ij}	d_i	$\min(\mathsf{d}_i, \mathsf{d}_j)$	$\min_{j \in \underline{\nu}(i)} (\mathsf{d}_i, \mathsf{d}_j)$
Example			

Table 3.1: Evaluation of the EdgePD d_{ij} using the CellPD of the two neighbor elements. Analytic formula on first line. Examples on the second line where CellPD are surrounded in red and EdgePD for internal edges are in black. Missing cells are assumed to have CellPD equal to 2.

3.1.4.3 Convergence of the MOOD method

We first recall the classical stability result (see [17] and references herein).

Proposition 3.8 *Let us consider the generic first-order finite volume scheme (3.2) with reflective boundary conditions. If the numerical flux is consistent and monotone, then the DMP property given by definition 3.2 is satisfied.*

It implies that if $u_{ij} = u_i$ and $u_{ji} = u_j$ for all $j \in \underline{\nu}(i)$ then relation (3.7) holds. To prove that the iterative MOOD method provides a solution which satisfies the DMP, we introduce the following definition.

Definition 3.9 *An* EPD *strategy is said upper-limiting (with respect to the CellPD) if for any* K_i

$$\mathsf{d}_i = \bar{\mathsf{d}} \implies \mathsf{d}_{ij} \leq \bar{\mathsf{d}} \ \text{ and } \ \mathsf{d}_{ji} \leq \bar{\mathsf{d}}, \quad \forall j \in \underline{\nu}(i). \tag{3.15}$$

We then have the following theorem.

Theorem 3.10 *Let us consider the generic high-order finite volume scheme with reflective boundary conditions and assume that the numerical flux is consistent and monotone. If the* EPD *strategy is upper-limiting then the MOOD method provides an updated solution* u_h^{n+1} *which satisfies the DMP property after a finite number of iterations.*

Proof. Let d_i be the CellPD of cell K_i. If $\mathsf{d}_i = 0$, then equation (3.15) implies that $\mathsf{d}_{ij} = \mathsf{d}_{ji} = 0$, hence $u_{ij}^n = u_i^n$ and $u_{ji}^n = u_j^n$, for all $j \in \underline{\nu}(i)$. We recover the first-order scheme (3.2) and proposition 3.8 yields that u_i^{n+1} satisfies the DMP property (3.7). Otherwise, if $\mathsf{d}_i > 0$ then two situations arise. Either the Maximum principle is satisfied and we do not modify d_i or we decrement d_i. Consequently if the maximum principle is not satisfied for all cells, then there is at least one cell having its *CellPD* positive which has to be decremented. Since we can not decrement more than $d_{\max} - \#(\mathcal{E}_{el})$ times, the iterative procedure stops after a finite number of iterations and the solution satisfies the DMP property. □

Remark 3.11 *Note that* EPD_1 *and* EPD_2 *are upper-limiting strategies whereas* EPD_0 *strategy does not satisfy condition (3.15). Thus* EPD_0 *cannot be used since MOOD iterative procedure may loop endlessly.* □

Remark 3.12 *To carry out a third-order Runge-Kutta time discretization (3.6) which provides a solution satisfying the DMP property, one has to perform the MOOD technique for each explicit sub-step since (3.6) can be written as a convex combination.* □

3.1.5 Extension to the Euler Equations

In this section, we propose an extension of the MOOD method to the Euler equations.

$$\partial_t \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ E \end{pmatrix} + \partial_{x_1} \begin{pmatrix} \rho u_1 \\ \rho u_1^2 + p \\ \rho u_1 u_2 \\ u_1(E+p) \end{pmatrix} + \partial_{x_2} \begin{pmatrix} \rho u_2 \\ \rho u_1 u_2 \\ \rho u_2^2 + p \\ u_2(E+p) \end{pmatrix} = 0, \tag{3.16}$$

where ρ, $\mathbf{V} = (u_1, u_2)$ and p are the density, velocity and pressure respectively while the total energy per unit volume E is given by

$$E = \rho\left(\frac{1}{2}\mathbf{V}^2 + e\right), \quad \mathbf{V}^2 = u_1^2 + u_2^2,$$

where e is the specific internal energy. For an ideal gas, this system is closed by the equation of state

$$e = \frac{p}{\rho(\gamma - 1)},$$

with γ the ratio of specific heats.

Despite that the physical variables do not have to respect the maximum principle, classical methods such as the MUSCL technique use a limiting procedure derived from the scalar case to keep the numerical solution from producing spurious oscillations. A popular choice consists of reconstructing and limiting the density, the velocity components and the pressure variables but other limitations can be carried out: the internal energy, the specific volume or the characteristic variables for instance.

Although applying the MOOD technique to each variable independently gives physically admissible solutions, an excessive diffusion is noticed. We thus propose a strategy to both have an accurate approximation where the solution is smooth and prevent the oscillations from appearing close to the discontinuities. In the following we consider ρ, u_1, u_2 and p as the variables to be reconstructed.

First we have to provide physically relevant reconstructed values at quadrature points, and since no limitation is used in the MOOD method, negative reconstructed values for pressure or density must be avoided (it would be the same for energy or specific volume). In that case, first-order values are substituted to the unphysical reconstructed values, for instance if the reconstructed value ρ_{ij} is negative on cell K_i, we replace it with the mean value ρ_i.

We now describe how we choose to use the two fundamental notions of the MOOD method (CellPD and EdgePD) in the Euler equations framework. Instead of using one CellPD per cell

and per variable, we choose to define only one CellPD per cell and to use it for all variables. Consequently only one EdgePD is defined per side of an edge and used for all variables.

As in the scalar case, we first build the local polynomial reconstruction of maximal degree $d_{\max}$ for each variable. Then we apply the MOOD algorithm of Section 3.1.4.2 where we substitute steps 5 and 6 with the following stages.

5. **Density DMP test.** The DMP criterion is checked on the density
$$\min_{j\in\overline{\nu}(i)}(\rho_i^n,\rho_j^n)\leq\rho_i^\star\leq\max_{j\in\overline{\nu}(i)}(\rho_i^n,\rho_j^n). \tag{3.17}$$
If $\rho_i^\star$ does not satisfy (3.17) the CellPD is decremented, $\mathsf{d}_i := \max(0,\mathsf{d}_i-1)$.
6. **Pressure positivity test.** The pressure positivity is checked and if $p_i^\star\leq 0$ and d_i has not been altered by step 5 then the CellPD is decremented, $\mathsf{d}_i := \max(0,\mathsf{d}_i-1)$.
7. **Stopping criterion.** If, for all $i\in\mathcal{E}_{el}$, d_i has not been altered by steps 5 and 6 then the iterative procedure stops and returns $(\rho,\rho u_1,\rho u_2,E)_h^{n+1}=(\rho,\rho u_1,\rho u_2,E)_h^\star$ else go to step 2.

Next section is dedicated to numerical experiments to assess the computational efficiency of the MOOD method.

3.1.6 Numerical results — the scalar case

Let Ω be the unit square $[0,1]-[0,1]$. We first consider the linear advection problem of a scalar quantity u with velocity $V(\mathbf{x})$:
$$\partial_t u+\nabla.(Vu)=0, \tag{3.18a}$$
$$u(.,0)=u^0, \tag{3.18b}$$
where $V(\mathbf{x})$ is a given continuous function on Ω and u^0 is the initial function we shall characterize in the following. In this section periodic boundary conditions are prescribed on $\partial\Omega$.
Comparisons are drawn between the simple first-order Finite Volume method (denoted FV with an abuse of terminology), the MUSCL method proposed in [62] (MLP) and the MOOD method with $d_{\max}=1$ (MOOD-P1) and $d_{\max}=2$ (MOOD-P2).
We use the following monotone upwind numerical flux (see equation (3.2))
$$\mathbb{F}(u_i^n,u_j^n,\mathbf{n}_{ij})=[V(\mathbf{x})\cdot\mathbf{n}_{ij}]^+\,u_i^n+[V(\mathbf{x})\cdot\mathbf{n}_{ij}]^-\,u_j^n,$$
where the velocity is evaluated at the quadrature point $\mathbf{x}$ and the positive and negative parts are respectively defined by
$$[\alpha]^+=\max(0,\alpha)\quad\text{and}\quad[\alpha]^-=\min(0,\alpha).$$
Notice that we use $\overline{\nu}(i)$ as the reconstruction stencil. Lastly two Gauss points are used on each edge to provide a third-order accurate spatial integration while time integration is performed with a forward Euler scheme for the FV method and with the RK3-TVD method given by system (3.6) for the MLP and MOOD methods.
Following remark 3.12, we simply apply the MOOD procedure detailed in section 3.1.4.2 to each sub-step of the RK3-TVD. The CellPD are thus reinitialized to $d_{\max}$ at the beginning of each time sub-step.

3.1.6.1 Test descriptions

The method accuracy is measured using L^1 and L^∞ errors which are computed with

$$err_1 = \sum_{i \in \mathcal{E}_{el}} |u_i^N - u_i^0||K_i| \quad \text{and} \quad err_\infty = \max_{i \in \mathcal{E}_{el}} |u_i^N - u_i^0|,$$

where $(u_i^0)_i$ and $(u_i^N)_i$ are respectively the cell mean values at initial time $t = 0$ and final time $t = t_f = N\Delta t$.
Two classical numerical experiments are carried out to demonstrate the ability of the method to provide effective third-order accuracy and to handle discontinuities with a very low numerical diffusion.
Double Sine Translation (DST)
We consider a constant velocity $V = (2, 1)$ and the initial condition is the C^∞ function

$$u^0(x_1, x_2) = \sin(2\pi x_1)\sin(2\pi x_2).$$

The final time is $t_f = 2.0$. Since we use periodic boundary conditions, the final time corresponds to a full revolution such that the exact solution coincides with the initial one.
Solid Body Rotation(SBR)
First introduced by R.J. Leveque in [52], this solid body rotation test uses three shapes which are a hump, a cone and a slotted cylinder. Each shape is located within a circle of radius $r^0 = 0.15$ and centered at (x_1^0, y_2^0)

Hump centered at $(x_1^0, x_2^0) = (0.25, 0.5)$

$$u^0(x_1, x_2) = \frac{1}{4}(1 + \cos(\pi \min(r(x_1, x_2), 1))).$$

Cone centered at $(x_1^0, x_2^0) = (0.5, 0.25)$

$$u^0(x_1, x_2) = 1 - r(x_1, x_2).$$

Slotted cylinder centered at $(x_1^0, x_2^0) = (0.5, 0.75)$

$$u^0(x_1, x_2) = \begin{cases} 1 & \text{if } |x_1 - 0.5| < 0.25, \text{ or } x_2 > 0.85, \\ 0 & \text{elsewhere,} \end{cases}$$

where $r(x_1, x_2) = \dfrac{1}{r^0}\sqrt{(x_1 - x_1^0)^2 + (x_2 - x_2^0)^2}$. To perform the rotation, we use the velocity $V(\mathbf{x}) = (-x_2 + 0.5, x_1 - 0.5)$ and the final time $t_f = 2\pi$ corresponds to one full rotation.

3.1.6.2 Numerical results

⋆ Comparison between EPD_1 and EPD_2 strategies

We consider the DST test case on uniform meshes from $20-20$ to $160-160$ cells and compare the L^1 and L^∞ errors and convergence rates displayed in table 3.2 using EPD_1 and EPD_2 strategies with the MOOD-P2 method. We obtain an almost effective third-order convergence

in L^1 norm and a 1.6 convergence rate in L^∞ norm for the two strategies. We observe in this case that the L^1 and L^∞ errors for $\mathsf{EPD_1}$ are slightly less important than for $\mathsf{EPD_2}$ and the convergence orders seem to indicate that the $\mathsf{EPD_1}$ strategy should be privileged. Moreover, from a practical point of view, the $\mathsf{EPD_1}$ implementation is performed with a more compact stencil than the $\mathsf{EPD_2}$ (see remark 3.7). In the sequel, only $\mathsf{EPD_1}$ strategy is used.

Nb of Cells	$\mathsf{EPD_1}$				$\mathsf{EPD_2}$			
	err_1		err_∞		err_1		err_∞	
20x20	9.469E-02	—	3.960E-01	—	1.104E-01	—	4.506E-01	—
40x40	1.113E-02	3.09	1.333E-01	1.57	1.382E-02	3.00	1.566E-01	1.52
80x80	1.768E-03	2.65	4.164E-02	1.68	2.309E-03	2.58	5.196E-02	1.59
160x160	2.481E-04	2.83	1.304E-02	1.68	3.262E-04	2.82	1.698E-02	1.61

Table 3.2: L^1 and L^∞ errors and convergence rates for DST problem with the MOOD-P2 method: $\mathsf{EPD_1}$ strategy (left) and $\mathsf{EPD_2}$ strategy (right).

⋆ Comparison between FV, MLP, MOOD-P1 and MOOD-P2 with $\mathsf{EPD_1}$ strategy on uniform meshes

Double Sine Translation. We report in table 3.3, 3.4 and 3.5 the L^1 and L^∞ errors and convergence rates for FV, MLP, MOOD-P1, MOOD-P2, unlimited P1 and P2 reconstruction methods respectively. At last, we plot in figure 3.3 the convergence curves for the four methods as well as the convergence curves for the unlimited versions.

Nb of Cells	FV				MLP			
	err_1		err_∞		err_1		err_∞	
20x20	3.924E-01	—	9.371E-01	—	1.417E-01	—	3.765E-01	—
40x40	3.480E-01	0.17	8.375E-01	0.16	3.038E-02	2.22	1.121E-01	1.75
80x80	2.663E-01	0.39	6.241E-01	0.42	6.904E-03	2.14	3.534E-02	1.67
160x160	1.734E-01	0.62	3.964E-01	0.65	1.693E-03	2.03	1.167E-02	1.60

Table 3.3: L^1 and L^∞ errors and convergence rates for the DST on uniform meshes with FV and MLP methods.

Nb of Cells	MOOD-P1				MOOD-P2			
	err_1		err_∞		err_1		err_∞	
20x20	1.502E-01	—	4.876E-01	—	9.469E-02	—	3.960E-01	—
40x40	3.141E-02	2.26	1.629E-01	1.58	1.113E-02	3.09	1.333E-01	1.57
80x80	7.438E-03	2.08	5.188E-02	1.65	1.768E-03	2.65	4.164E-02	1.68
160x160	1.787E-03	2.06	1.675E-02	1.63	2.481E-04	2.83	1.304E-02	1.68

Table 3.4: L^1 and L^∞ errors and convergence rates for the DST on uniform meshes with MOOD-P1 and MOOD-P2 methods.

Nb of Cells	P1				P2			
	err_1		err_∞		err_1		err_∞	
20x20	1.334E-01	—	3.227E-01	—	7.130E-02	—	1.729E-01	—
40x40	2.896E-02	2.20	6.593E-02	2.29	9.877E-03	2.85	2.427E-02	2.83
80x80	6.604E-03	2.13	1.408E-02	2.23	1.255E-03	2.98	3.091E-03	2.97
160x160	1.603E-03	2.04	3.310E-03	2.09	1.573E-04	3.00	3.876E-04	3.00

Table 3.5: L^1 and L^∞ errors and convergence orders for the DST on uniform meshes with P1 and P2 methods.

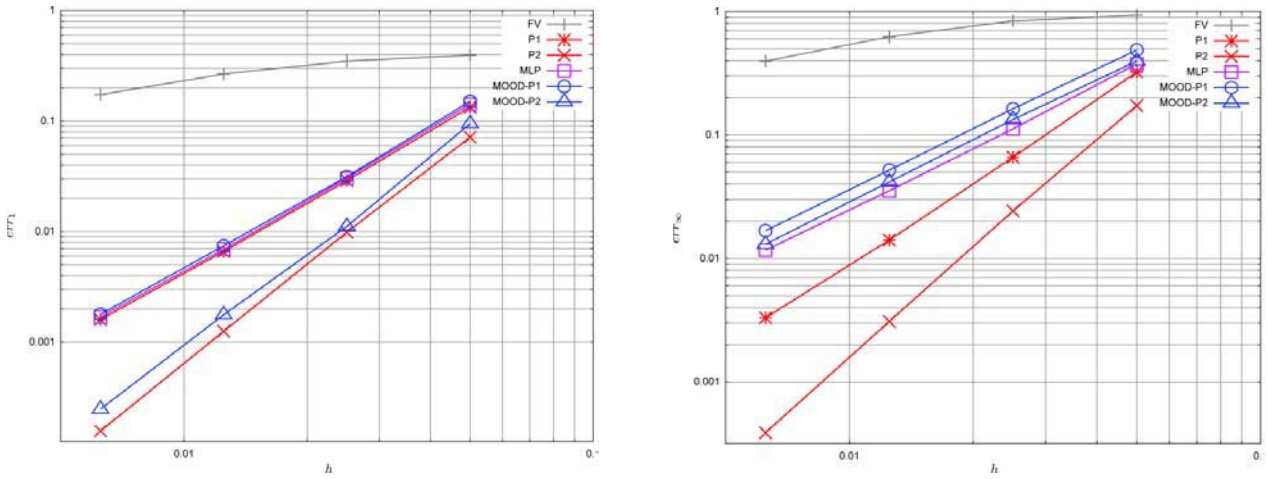


Figure 3.3: Convergence curves of err_1(left) and err_∞(right) for the DST on uniform meshes.

The high-order finite volume methods with the two Gauss points and the RK3 time scheme reach the optimal convergence rate for the unlimited P1 and P2 reconstructions hence the limiting procedure has to be blamed for the accuracy discrepancy.

Figure 3.3 shows that the optimal convergence rate in L^1 error for P1, MOOD-P1 and MLP methods is achieved since the curves fit very well. On the other hand, the P2 and MOOD-P2 curves are very close and parallel which confirms that MOOD-P2 is an effective third-order method for the L^1 norm. For the L^∞ norm, none of the limited methods is over the effective second-order while the unlimited P1 and P2 provide an effective second- and third-order respectively. Indeed the strict maximum principle application at extrema is responsible for the L^∞ error discrepancy and we can expect nothing more than a second-order scheme in L^∞ norm, whatever the polynomial degree is when the DMP condition is enforced.

Solid Body Rotation. We employ a $140 - 140$ uniform mesh of square elements in order to compare our results with $100 - 100 - 2$ triangular mesh in reference [62]. We display in the left panels of figure 3.4 three-dimensional elevations while top views of ten uniformly distributed isolines from 0 to 1 are printed in the right panels. We can measure the scheme accuracy by counting the number of isolines outside of the slot since the exact solution isolines would fit the slot shape. The smaller number of isolines outside of the slot is, the more accurate the scheme is. With the MLP reconstruction, we observe three isolines outside while we have only two

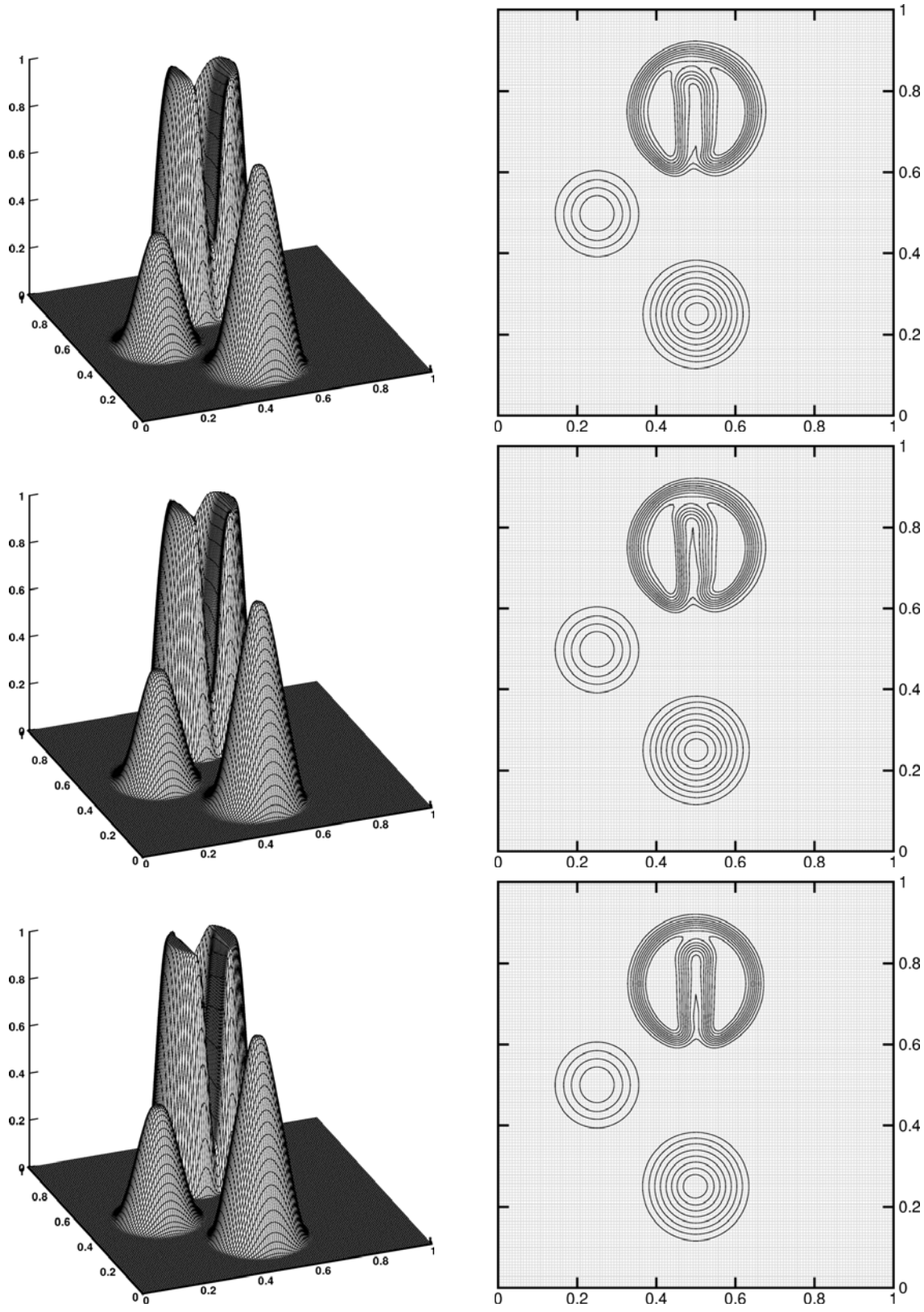


Figure 3.4: Results of SBR on a 140x140 uniform mesh. Isolines are from 0 to 1 by 0.1. Top: MLP method — Middle: MOOD-P1 method — Bottom: MOOD-P2 method.

with the MOOD-P1. At last, the outstanding result is that we have just one isoline outside of the slot with the MOOD-P2 method which proves the great ability of the technique to handle and preserve discontinuities.

⋆ Comparison between FV, MLP, MOOD-P1 and MOOD-P2 with EPD_1 strategy on non-uniform meshes

Approximation accuracy is reduced when one employs meshes with large deformations, *i.e.* the elements are no longer rectangular but quadrilateral with large aspect ratios. The present subsection investigates the MOOD method sensitivity to mesh distortion.
To obtain the distorted mesh for the DST, we proceed in two stages. First the following transformation is applied to an uniform mesh

$$x_1 \to \begin{cases} x_1(10x_1^2 + 5x_1 + 1), & \text{if } x_1 \leq 0.5, \\ (x_1 - 1)(10(x_1 - 1)^2 + 5(x_1 - 1)) + 1, & \text{elsewhere,} \end{cases}$$

and we operate in the same way with variable x_2.
Then we apply a second transformation

$$\begin{aligned} x_1 &\to x_1 + 0.1|x_1 - 0.5| \cos(6\pi(x_2 - 0.5)) \sin(4\pi(x_1 - 0.5)), \\ x_2 &\to x_2 + 0.1|x_2 - 0.5| \cos(4\pi(x_1 - 0.5)) \sin(6\pi(x_2 - 0.5)). \end{aligned}$$

As an example two non-uniform meshes are given in figure 3.5. Notice that the shape of domain Ω is preserved by the transformation.
Double Sine Translation. We report in tables 3.6, 3.7 and 3.8 the L^1 and L^∞ errors and convergence rates for FV, MLP, MOOD-P1, MOOD-P2, unlimited P1 and P2 reconstruction methods respectively. At last, we plot in figure 3.6 the convergence curves for the four methods as well as the convergence curves for the unlimited versions.

Nb of Cells	FV				MLP			
	err_1		err_∞		err_1		err_∞	
20x20	4.053E-01	—	9.032E-01	—	3.907E-01	—	8.752E-01	—
40x40	4.038E-01	0.01	9.822E-01	-0.12	1.893E-01	1.05	5.306E-01	0.72
80x80	3.834E-01	0.07	9.486E-01	0.05	4.370E-02	2.11	1.806E-01	1.55
160x160	3.144E-01	0.29	7.825E-01	0.28	9.846E-03	2.15	5.889E-02	1.62

Table 3.6: L^1 and L^∞ errors and convergence rates for the DST on non-uniform meshes with FV and MLP methods.

We first observe in table 3.8 an accuracy discrepancy with the unlimited reconstructions since the L^∞ errors are roughly ten times larger for the distorted mesh than for the uniform one given in table 3.5. Nevertheless, we obtain good effective rates of convergence both in L^1 and L^∞ norm for the P1 and P2 reconstructions. Optimal second-order scheme is achieved for the P1 method and convergence rate is around 2.9 for the P2 reconstruction.
For the L^1 norm, P1, MOOD-P1 and MLP convergence curves fit well hence we get the optimal accuracy with the three methods. In the same way, the P2 and MOOD-P2 are also superimposed

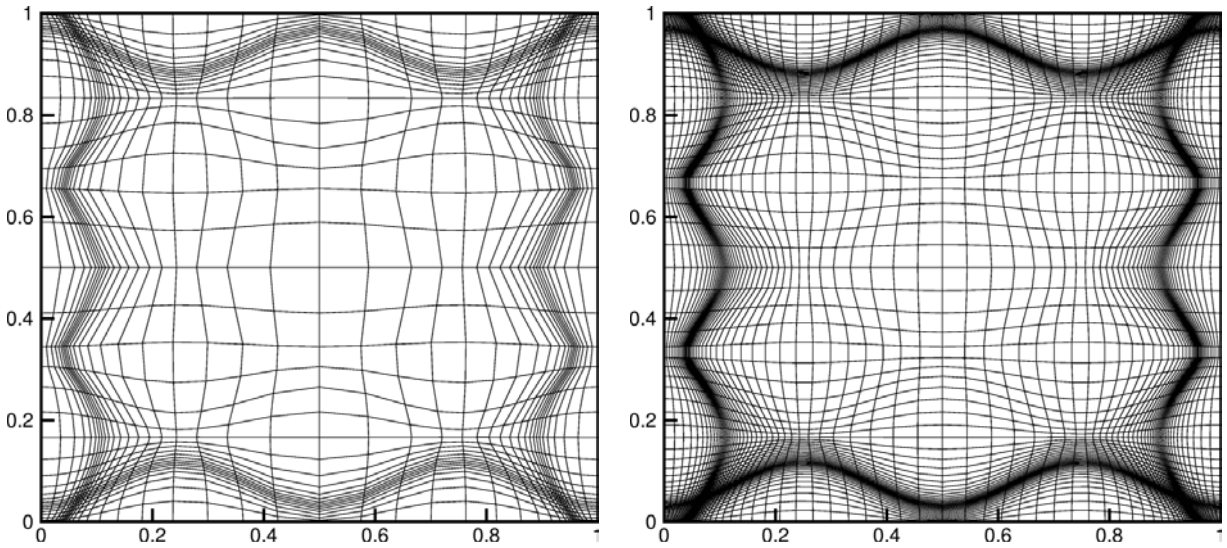


Figure 3.5: The $40 - 40$ and $80 - 80$ non-uniform meshes for the DST.

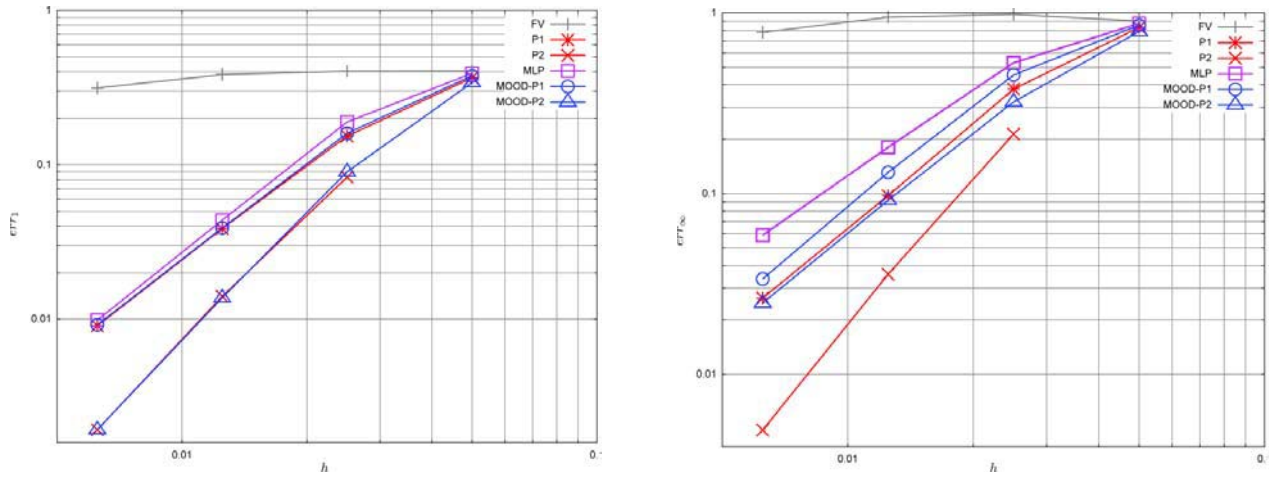


Figure 3.6: Convergence curves of err_1(left) and err_∞(right) for the DST on non-uniform meshes.

Nb of	MOOD-P1				MOOD-P2			
Cells	err_1		err_∞		err_1		err_∞	
20x20	3.770E-01	—	8.557E-01	—	3.408E-01	—	7.897E-01	—
40x40	1.599E-01	1.24	4.541E-01	0.91	8.992E-02	1.92	3.222E-01	1.29
80x80	3.892E-02	2.04	1.314E-01	1.79	1.375E-02	2.71	9.199E-02	1.81
160x160	9.170E-03	2.09	3.374E-02	1.96	1.922E-03	2.84	2.483E-02	1.89

Table 3.7: L^1 and L^∞ errors and convergence rates for the DST on non-uniform meshes with MOOD-P1 and MOOD-P2 methods.

Nb of	P1				P2			
Cells	err_1		err_∞		err_1		err_∞	
20x20	3.658E-01	—	8.312E-01	—	FAIL	—	FAIL	—
40x40	1.534E-01	1.25	3.793E-01	1.13	8.328E-02	—	2.135E-01	—
80x80	3.856E-02	1.99	9.760E-02	1.96	1.403E-02	2.57	3.582E-02	2.58
160x160	9.052E-03	2.09	2.643E-02	1.88	1.920E-03	2.87	4.917E-03	2.86

Table 3.8: L^1 and L^∞ errors and convergence rates for the DST on non-uniform meshes with P1 and P2 methods.

which means that MOOD-P2 is optimal with respect to the unlimited case. For the L^∞ norm, MLP method convergence rate is around 1.6 whereas the MOOD-P1, MOOD-P2 and P1 provide a 1.9 convergence rate. Notice that the MOOD-P2 produces more accurate results but does not reach the third-order convergence since it has to respect a strict DMP property. Finally, table 3.9 shows that the extrema are better approximated with respect to the exact solution with the MOOD methods than the MLP method, in particular when coarse meshes are employed.

Nb of	MLP		MOOD-P1		MOOD-P2	
Cells	Min	Max	Min	Max	Min	Max
20x20	-3.740E-02	3.479E-02	-7.168E-02	7.566E-02	-1.376E-01	1.516E-01
40x40	-4.634E-01	4.645E-01	-5.445E-01	5.458E-01	-6.738E-01	6.792E-01
80x80	-8.179E-01	8.204E-01	-8.747E-01	8.743E-01	-9.098E-01	9.079E-01
160x160	-9.433E-01	9.431E-01	-9.655E-01	9.668E-01	-9.752E-01	9.748E-01

Table 3.9: Min and Max for DST on non-uniform meshes with MLP, MOOD-P1 and MOOD-P2.

Solid Body Rotation. The mesh deformation presented above is not as relevant for the SBR as for the DST since the solid bodies rotate and do not go through the boundaries. A slight modification of the first step has been done

$$x_1 \to \begin{cases} x_1(5x_1^2 + 2.5x_1 + 1), & \text{if } x_1 \leq 0.5, \\ (x_1 - 1)(5(x_1 - 1)^2 + 2.5(x_1 - 1)) + 1, & \text{elsewhere.} \end{cases},$$

and we operate in the same way with variable x_2.
The $140 - 140$ non-uniform mesh is visible on the isolines top views. We display in the left

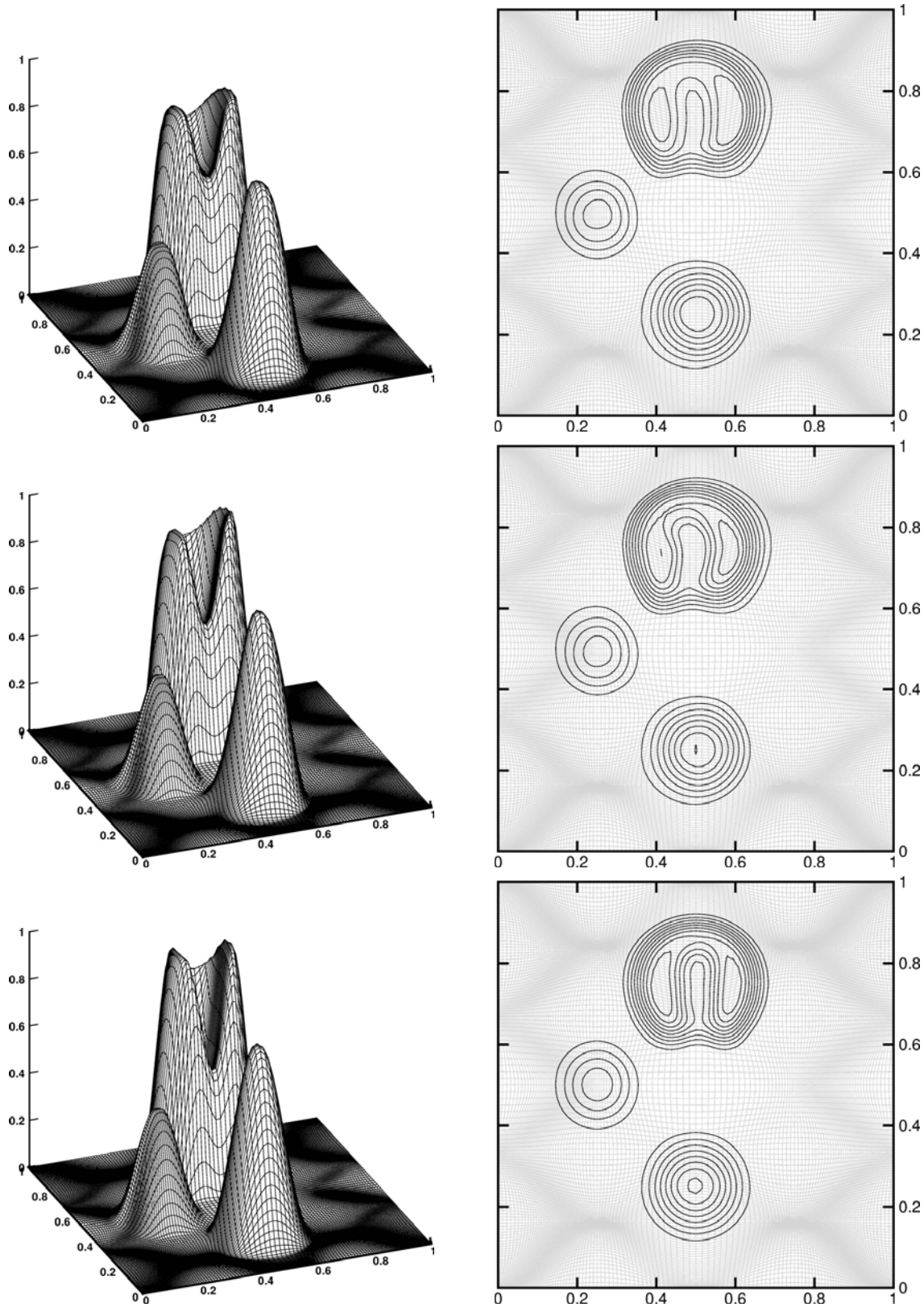


Figure 3.7: Results of SBR on a 140x140 non-uniform mesh. Isolines are from 0 to 1 by 0.1. Top: MLP method — Middle: MOOD-P1 method — Bottom: MOOD-P2 method.

panels of figure 3.7 three-dimensional elevations while top views of ten uniformly distributed isolines from 0 to 1 are in the right panels.

As in the smooth case, MOOD methods perform better than MLP on the distorted mesh. Although they are both second-order methods, we notice that MOOD-P1 gives a clearly better solution than the one computed with MLP, even on the smooth profiles. Moreover the MOOD-P2 result supports the usefulness of using a third-order method since an important gain in symmetry of the solution is obtained.

3.1.7 Numerical results — the Euler case

We now turn to the Euler equations (3.16) to test the MOOD method. Efficiency, accuracy and stability of the method are investigated on classical tests. In the present article, we use the HLL numerical flux detailed in [78]. Once again comparisons are drawn with the MLP technique proposed in [62]. We apply the MOOD method using the detection strategy presented in Section 3.1.5 to each sub-step of the RK3-TVD time discretization.

First the classical 1D Sod shock tube is used to test the ability of MOOD in reproducing simple waves. This test is first run on an uniform mesh and then on a non-uniform one to estimate the gain obtained when using MOOD method. Then we proceed with a 2D Riemann problem proposed by [71] (see also [53]). We conclude the series of tests with two classical references, the Mach 3 wind tunnel with a step problem [62, 91] and the double Mach problem [62, 91]. These two tests are run with MLP, MOOD-P1 and MOOD-P2 on uniform meshes for comparison purposes with classical results from literature.

3.1.7.1 Sod Shock Tube

The one dimensional Sod problem is used as a sanity check for the MOOD method. The computational domain is the rectangular domain $\Omega = [0, 1] - [0, 0.2]$. The exact solution is invariant in x_2-direction. The interface between the left state $(\rho, u_1, u_2, p) = (1, 0, 0, 1)$ and the right one $(0.125, 0, 0, 0.1)$ is located at $x_1 = 0.5$. Reflective boundary conditions are prescribed. The final time is $t_f = 0.2$.

Uniform mesh. The computational domain is uniformly meshed by 100 cells in the x_1 direction and 10 cells in the x_2 direction. We plot the density and the x_1-velocity at the final time with the exact solution using the MLP, MOOD-P1 and MOOD-P2 methods in figure 3.8. The curves show a very good agreement between the three methods. The plateau between the contact and the shock is wavy with the MLP method while MOOD produces better constant states. However we observe an undershoot (resp. overshoot) at the tail of the rarefaction with MOOD-P2 for the density (resp. velocity).

Non-uniform mesh. The same simulation is performed on the non-uniform mesh plotted in figure 3.9. The density and the x_1-velocity solutions at the final time using the MLP, MOOD-P1 and MOOD-P2 methods are also printed in figure 3.9. All cell values are represented so that the preservation of the 1D symmetry in the x_2 direction can be evaluated by the thickness of the points cloud. Clearly the MLP method provides the largest dispersion whereas the MOOD-P2 method manages to better preserve the x_2 invariance. Such a test case suggests that the MOOD method is less sensitive to mesh deformation. As in the uniform case an undershoot at the tail of the rarefaction wave appears for MOOD-P2 method but the solution is genuinely improved by

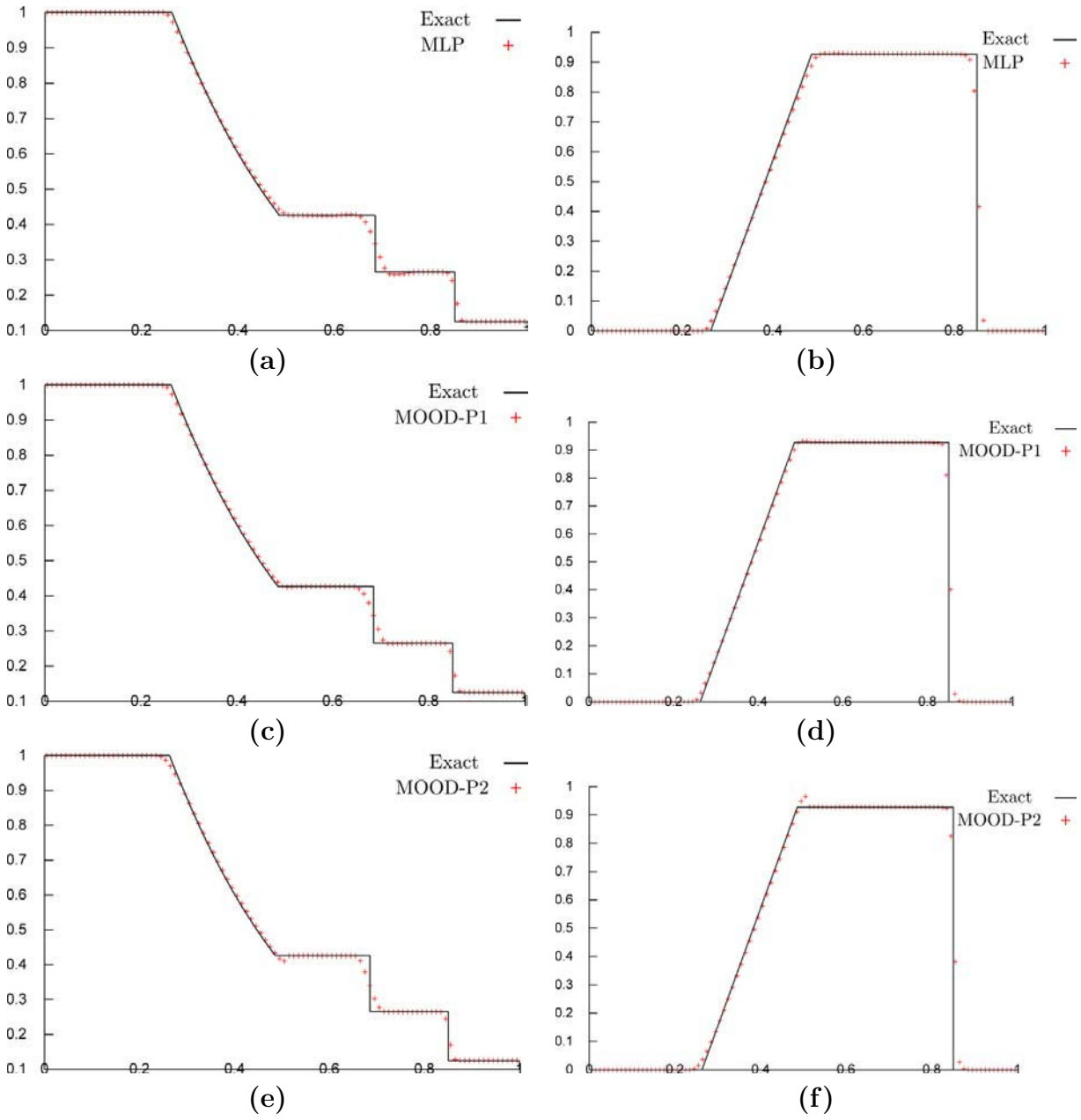


Figure 3.8: Sod shock tube problem: Density and x_1-velocity solutions on $100 - 10$ uniform mesh for **(a-b)**: MLP — **(c-d)**: MOOD-P1 — **(e-f)**: MOOD-P2.

comparison with MLP. The MOOD-P1 is an intermediate case where the dispersion is reduced in comparison with the MLP method but where the MOOD-P2 accuracy is not reached.

3.1.7.2 Four states Riemann problem

We now deal with one of the four states Riemann problem which corresponds to a truly 2D Riemann problem. The computational domain $\Omega = [0,1] - [0,1]$ is first uniformly meshed by a $100 - 100$ and then by a $400 - 400$ quadrangles grid. The four sub-domains correspond to four identical squares separated by the lines $x_1 = 0.5$ and $x_2 = 0.5$. Initial conditions on each sub-domains are

- for the lower-left domain Ω_{ll}, $(\rho, u_1, u_2, p) = (0.029, 0.138, 1.206, 1.206)$,
- for the lower-right domain Ω_{lr}, $(0.3, 0.5323, 0, 1.206)$,
- for the upper-right domain Ω_{ur}, $(1.5, 1.5, 0, 0)$,
- for the upper-left domain Ω_{ul}, $(0.3, 0.5323, 1.206, 0)$.

Each sub-domain is filled with a perfect gas of constant $\gamma = 1.4$. Outflow boundary conditions are prescribed and the computation is carried out till the final time $t_f = 0.3$. Density at the final time is presented for the three methods in figure 3.10. For each method on the left side one displays a three-dimensional elevation on the $100 - 100$ mesh while in the right panels 30 isolines are plotted between the minimal density, ρ_m, and maximal one, ρ_M of each method on the $400 - 400$ mesh. The 3D views clearly show that some artificial oscillations on the plateau are generated by the MLP method whereas the MOOD method better preserves the constant states. On the isoline view, we observe that the MOOD-P2 method gives thinner shocks and a finer resolved central peak at $x_1 = x_2 = 0.35$. As expected, this suggests that the MOOD-P2 method is more accurate.

3.1.7.3 Mach 3 wind tunnel with a step

The test was initially proposed in [91]. A uniform Mach 3 flow enters in a tunnel which contains a 0.2 unit length step leading to a flow with complex structures of interacting shocks. The wind tunnel is 1 length unit wide and 3 length units long and the step is located at 0.6 length unit from the left-hand side of the domain. At the initial time we consider a perfect gas ($\gamma = 1.4$) with constant density $\rho^0 = 1.4$, uniform pressure $p^0 = 1.0$ and constant velocity $\mathbf{V}^0 = (3, 0)$. Reflective boundary conditions are prescribed for the upper and lower sides as well as in front of the step. An inflow condition is set on the left boundary and an outflow condition on the right one. Numerical simulations are carried out till the final time $t_f = 4$.
We plot a series of figures presenting 30 density isolines for two different uniform meshes on which the three methods are tested. We first consider the situation with coarse mesh using $120 - 40$ cells. Figure 3.11 represents the density computed with the MLP, the MOOD-P1 and MOOD-P2 methods respectively on top, middle and bottom panels. It is noticeable that the MOOD method results are the most accurate. The shocks are less diffused and we can already observe the contact discontinuity formation of the upper slip line. With the MLP method, we remark that the formation of a triple point at $x_1 = 1.25$ above the step (at a distance of about 0.1) while the junction point should be exactly on the step interface. With the MOOD-P2 method, the triple point is closer to the interface (half the distance with respect to the MLP case).

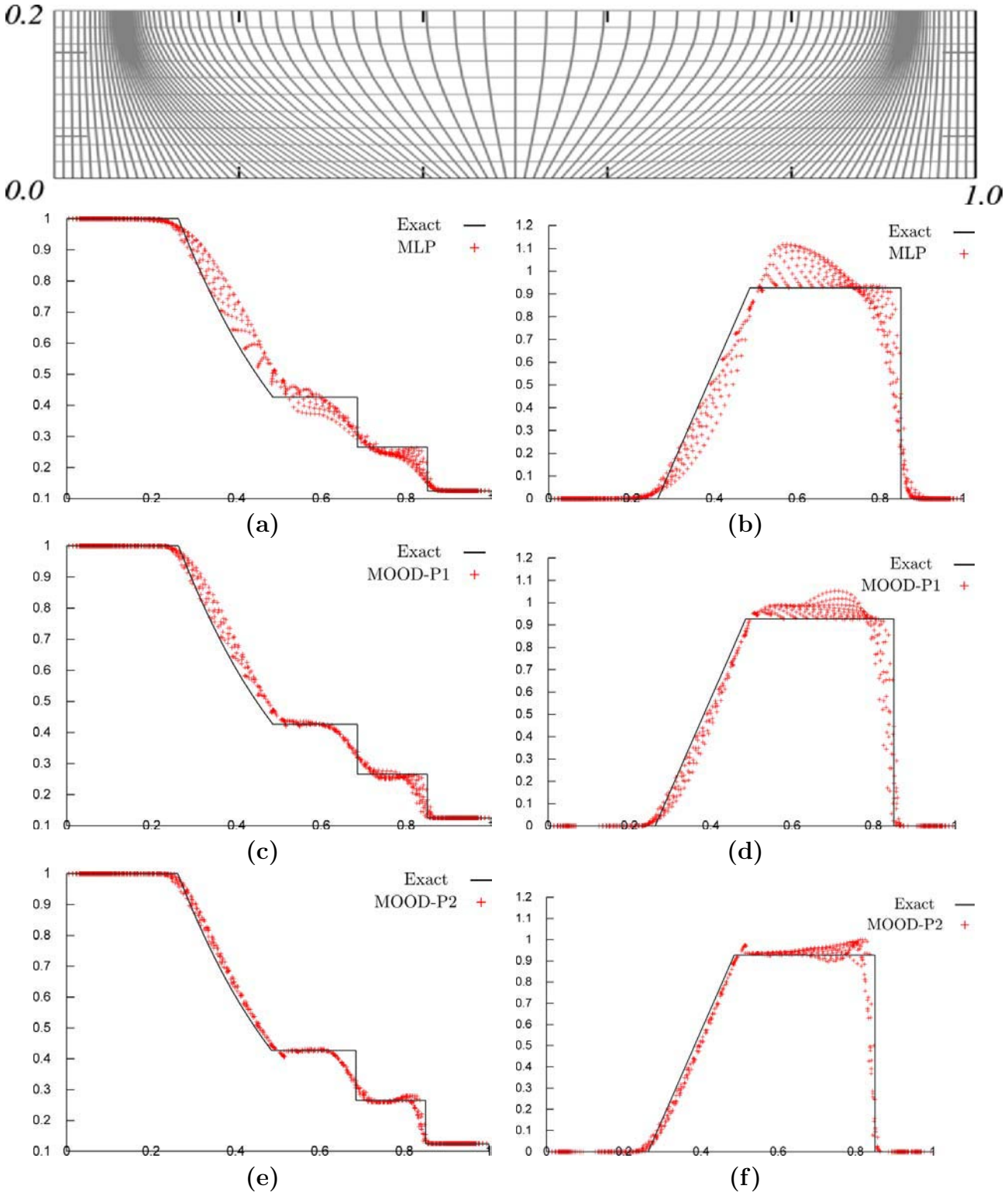


Figure 3.9: Sod shock tube problem: Non-uniform $100 - 10$ mesh (Top) — Density and x_1-velocity solutions on the above mesh for **(a-b)**: MLP — **(c-d)**: MOOD-P1 — **(e-f)**: MOOD-P2.

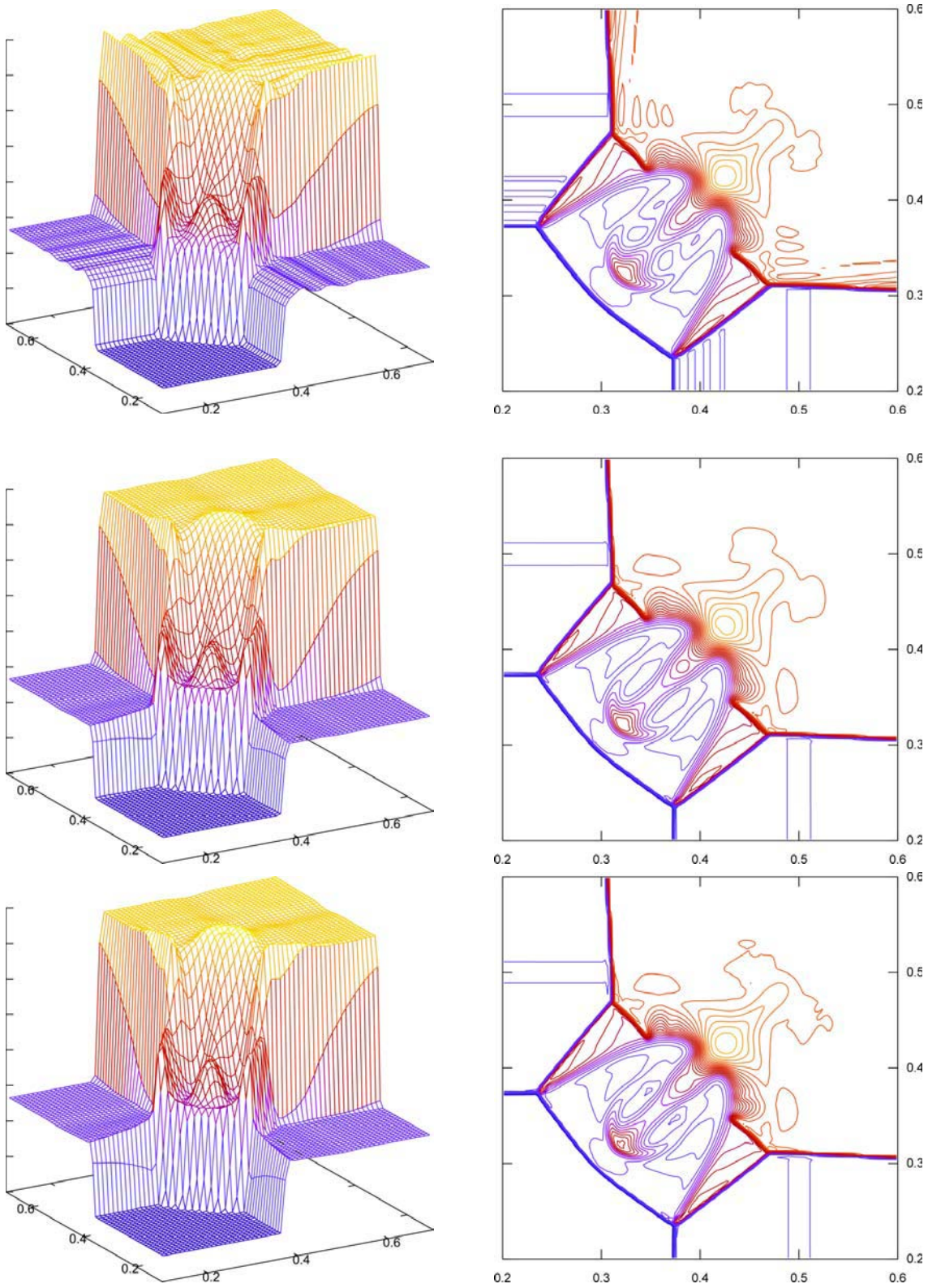


Figure 3.10: Density solution to the Four states Riemann problem. On the left 3D views on the $100-100$ mesh. On the right top views with 30 isolines between ρ_m and ρ_M on the $400-400$ mesh. Top: MLP method $\rho_m = 0.138$ $\rho_M = 1.821$ — Middle: MOOD-P1 method $\rho_m = 0.1377$ $\rho_M = 1.805$ — Bottom: MOOD-P2 method $\rho_m = 0.1379$ $\rho_M = 1.805$.

We plot the density obtained with a finer uniform mesh of $480 - 160$ cells in figure 3.12. The mesh refinement implies more accurate solutions for any method. Nevertheless MOOD methods still provide the best numerical approximations. However the method does not reveal the Kelvin-Helmholtz instabilities as in [24] as the strict DMP on the density reduces the scheme accuracy along the slip line and consequently increases the numerical dissipation.

3.1.7.4 Double Mach reflection of a strong shock

The last problem is the double mach reflection of a strong shock proposed in [91]. This test problem involves a Mach 10 shock which initially makes a 60° angle with a reflecting wall. The air ahead of the shock is at rest and has uniform initial density $\rho^0 = 1.4$ and pressure $p^0 = 1$. A perfect gas with $\gamma = 1.4$ is considered. The reflecting wall lies along the bottom of the domain, beginning at $x_1 = 1/6$. The shock makes a 60 degrees angle with the x_1 axis and extends to the top of the domain at $x_2 = 1$. The short region from $x_1 = 0$ to $x_1 = 1/6$ along the bottom boundary at $x_2 = 0$ is always assigned values for the initial post-shock flow. We prescribe a reflective condition on the bottom part for $x_1 > 1/6$, inflow boundary condition on the left side and outflow condition on the right side. At the top boundary, the boundary conditions are set to describe the exact motion of the Mach 10 flow (see [24]).
First for the three methods, a 30 density isolines top view on the $480 - 120$ uniform mesh using Lax-Friedrich's flux are plotted in figure 3.13. These results have to be compared to results of figure 12 in [33] and figure 13 in [45]. Then zoomed top views of 50 isolines — between minimal and maximal values, ρ_m and ρ_M respectively, taken over the results of the three methods on a same mesh — of the results obtained with the HLL flux are plotted in figure 3.14 for the $960 - 240$ uniform mesh on left and for the $1920 - 480$ one on right.
The first Mach stem M1 is connected to the main triple junction point with the incident shock wave and the reflected wave. A slip line is generated from the triple junction point behind the incident shock. A secondary Mach stem M2 also appear and interact with the slip line. As expected, the MOOD-P2 manages to better capture the Mach stem M1 (and M2 when we employ finer meshes) with respect to the two other methods. The slip line corresponds to a contact discontinuity where the jump of tangential velocity may generate Kelvin Helmholtz instabilities. Usually, the amount of instabilities measures the numerical diffusion influence [67]: large instabilities derive from small numerical diffusion and the number of plane vortexes in the slip line is a qualitative measure of the scheme diffusivity. In our test, even with the finest mesh, no instability is reported. Indeed, the application of a strict DMP reduces the accuracy of the scheme in the vicinity of the slip line maintaining a too large amount of diffusion. Nevertheless, other choices of detection variables could be investigated to reduce the numerical diffusion of contact discontinuities.

⋆ Computational cost comparison between MLP, MOOD-P1 and MOOD-P2

In this last section, we give in table 3.10 the ratios between MOOD methods computational times and MLP ones. For each test case, computational times are calculated on a given mesh. Numerical experiments show that the ratios are equivalent for finer or coarser meshes.
We recall that these ratios should only be taken as examples because computational times are strongly dependent of implementation and compilation and all runs are carried out on a

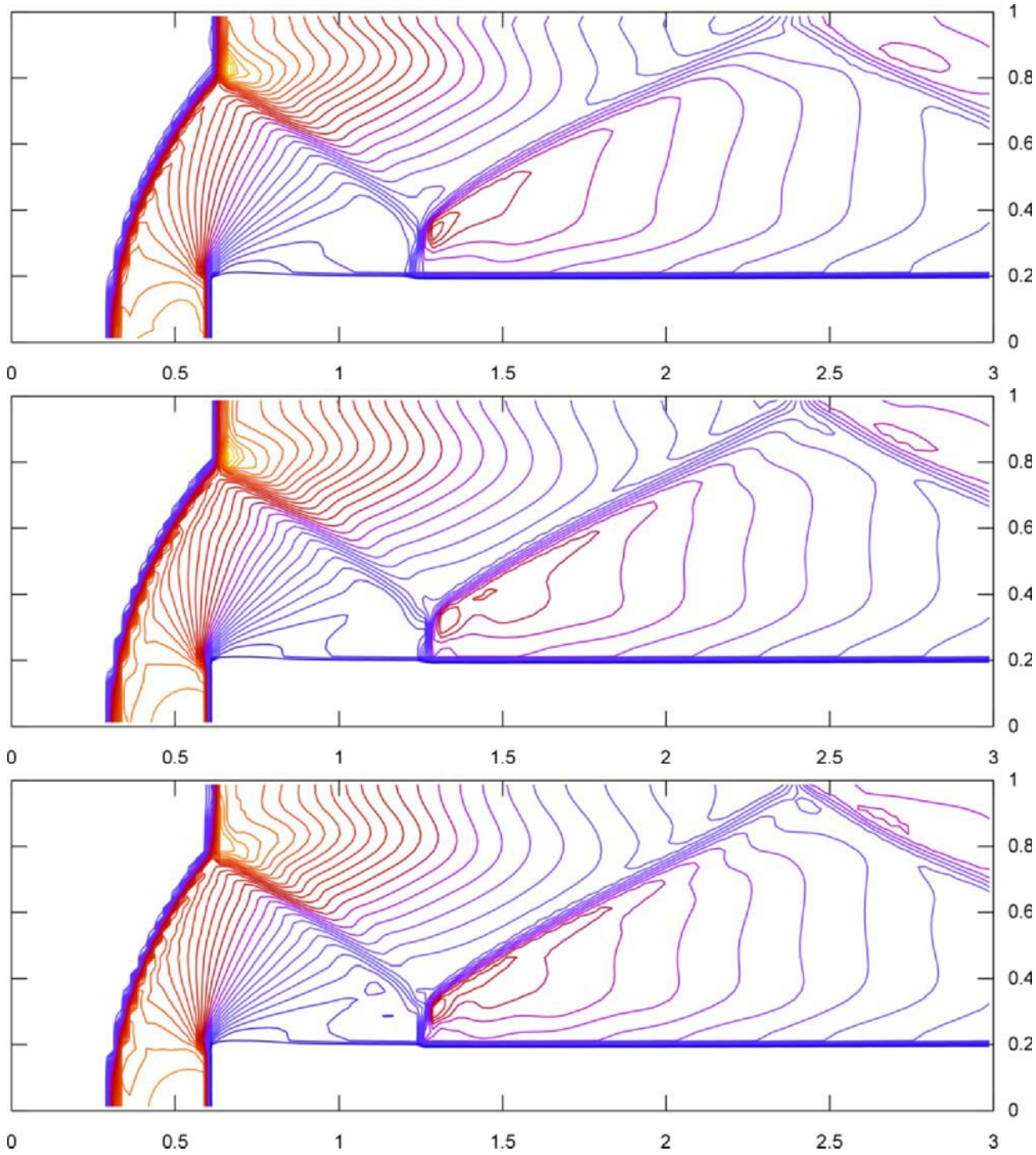


Figure 3.11: Mach 3 problem — Density solutions with 30 isolines between ρ_m and ρ_M on a $120 - 40$ uniform mesh. Top: MLP method $\rho_m = 0.5437$ $\rho_M = 6.75$ — Middle: MOOD-P1 method $\rho_m = 0.5589$ $\rho_M = 6.58$ — Bottom: MOOD-P2 method $\rho_m = 0.5358$ $\rho_M = 6.047$.

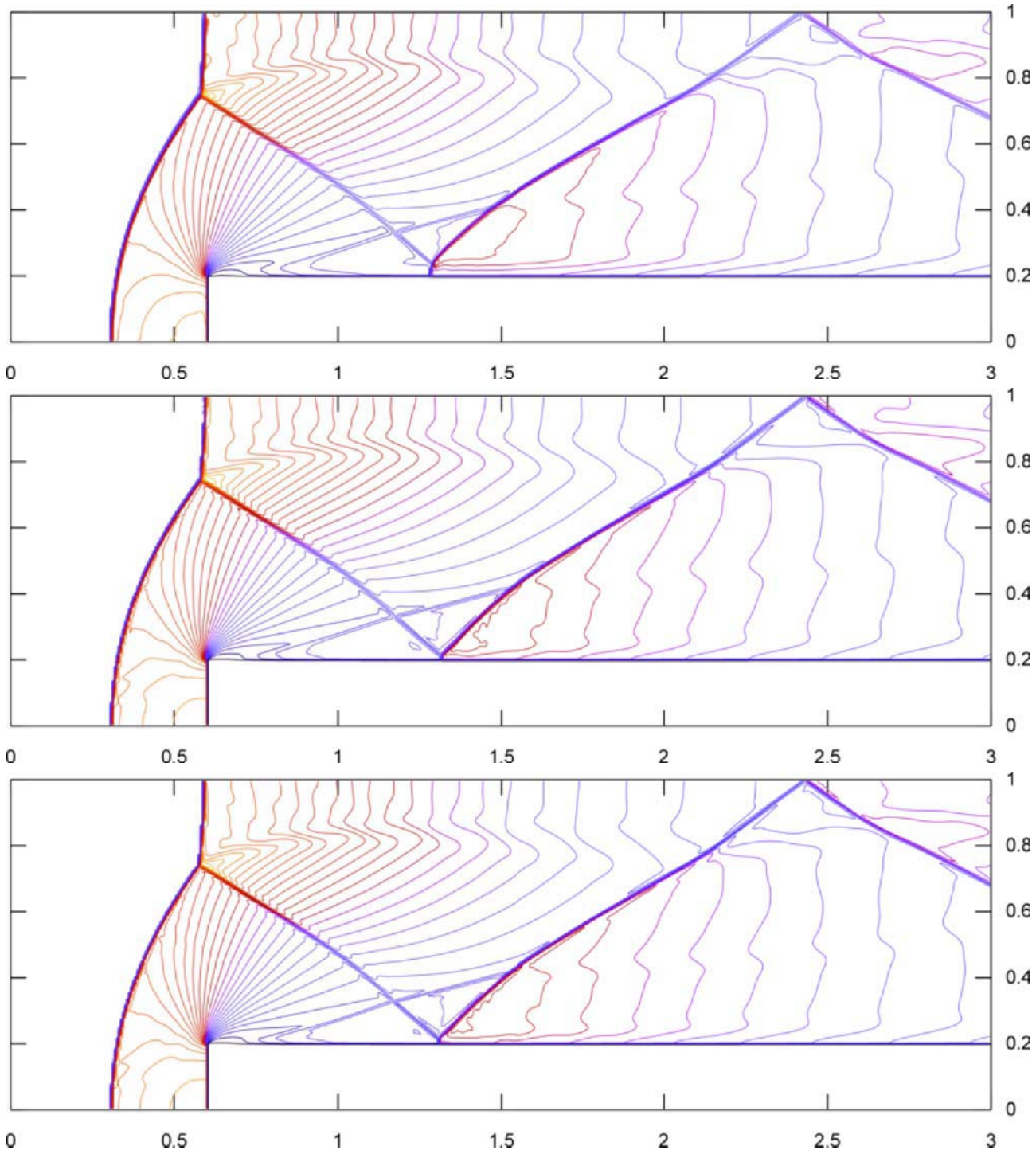


Figure 3.12: Mach 3 problem — Density solutions with 30 isolines between ρ_m and ρ_M on $480 - 160$ mesh. Top: MLP method $\rho_m = 0.176$ $\rho_M = 6.802$ — Middle: MOOD-P1 method $\rho_m = 0.150$ $\rho_M = 6.483$ — Bottom: MOOD-P2 method $\rho_m = 0.123$ $\rho_M = 6.257$.

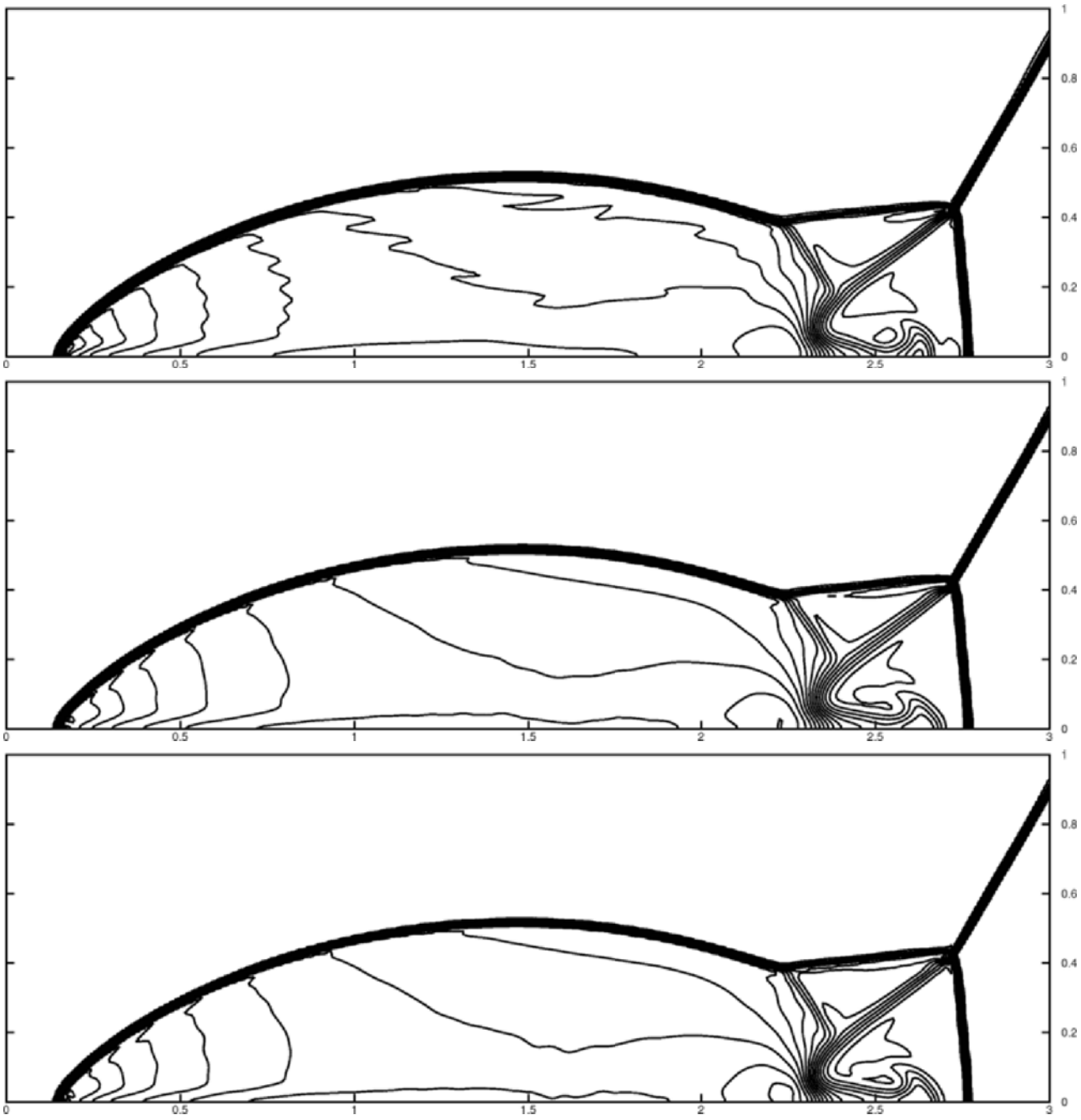


Figure 3.13: Double Mach problem on $480-120$ — Top: MLP method $\rho_m = 1.40$ $\rho_M = 22.21$ — Middle: MOOD-P1 method $\rho_m = 1.40$ $\rho_M = 20.05$ — Bottom: MOOD-P2 method $\rho_m = 1.40$ $\rho_M = 20.10$.

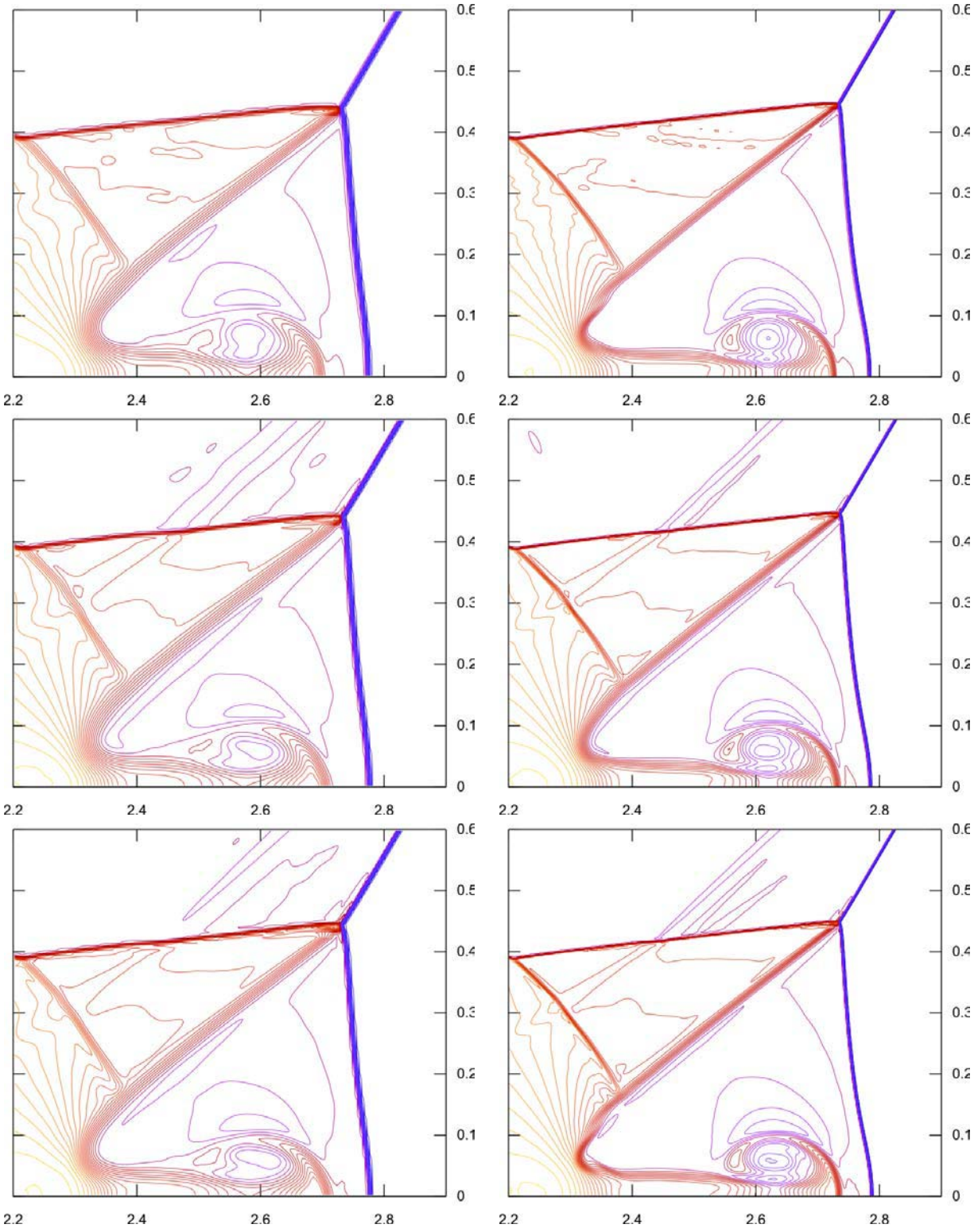


Figure 3.14: Double Mach problem on $960 - 240$ (left) and on $1920 - 480$ (right) — Zoom on the wave interaction zone — Top: MLP method $\rho_m = 1.400$ $\rho_M = 22.400$ on left and $\rho_m = 1.400$ $\rho_M = 22.68$ on right— Middle: MOOD-P1 method $\rho_m = 1.236$ $\rho_M = 22.550$ on left and $\rho_m = 1.216$ $\rho_M = 22.0$ on right — Bottom: MOOD-P2 method$\rho_m = 1.162$ $\rho_M = 22.800$ on left and $\rho_m = 1.146$ $\rho_M = 21.99$ on right.

single core. Table 3.10 shows that the MOOD-P1 method is slightly more expensive than MLP

Problem \ Method	MLP	MOOD-P1	MOOD-P2
DST	1	1.1	1.73
SBR	1	1.4	2.65
Sod Shock Tube	1	0.84	1.3
Mach 3 Wind	1	1.08	1.6
Double Mach	1	0.99	1.06
Average	*1*	*1.08*	*1.67*

Table 3.10: Computational time ratios between MOOD methods and MLP for different problems.

but gives better results on general meshes. In the scalar case, the difference between ratios of DST and SBR problem are explained by the fact that more iterations during the MOOD procedure, due to more DMP violations, are implied by non-smooth profiles. The MOOD-P2 computational cost is competitive (at most around 2.7 times more expensive than MLP on our numerical experiments) in regard to the observed accuracy improvement, see for instance figures 3.7 or 3.9.

3.1.8 Conclusion and perspectives

This paper presents a high-order polynomial finite volume method named Multi-dimensional Optimal Order Detection (MOOD) for conservation laws. Contrarily to classical high-order methods MOOD procedure is based on a test of the Discrete Maximum Principle (DMP) after an evaluation of the solution with unlimited polynomials. If the DMP property is not fulfilled then the polynomial degree is reduced and the solution is locally re-evaluated. This procedure is repeated up to satisfaction of the DMP which is always achieved after a finite number of iterations.
There are several important features of MOOD method which have to be compared with classical high-order methods, namely

- The MOOD method is an *a posteriori* limiting process, whereas classical limiting strategies perform an *a priori* limitation.
- The MOOD method computes one and only one high-order polynomial per cell and employs it without any limitation.
- Within the same cell the polynomial degree can be different on each edge.
- The MOOD method ensures the Discrete Maximum Principle (DMP) under the first-order CFL constraint.
- The MOOD method has no restriction to deal with higher polynomial degrees and polygonal meshes.

Two-dimensional numerical results are provided for advection and the Euler equations problems on regular and highly non-regular quadrangular meshes. They clearly show that MOOD method presents some promising good behaviors. The second-order MOOD method is at least equivalent to a second-order multi-dimensional MUSCL method on uniform grids but produces better results on non-uniform ones. A third-order version of MOOD has been shown to be effective

on regular and non-regular solutions for a small extra computational effort.
This paper is the first one presenting the MOOD concept and extensions are currently under investigations, as instance the behavior of the MOOD with polynomials of degree greater than two on polygonal meshes.

3.2 Part II: 6^{th}-order accuracy on 2D polygonal meshes

This section is dedicated to the second publication introducing important improvements of the MOOD method. The reference is:

S. Diot, S. Clain, R. Loubère, *Improved detection criteria for the Multi-dimensional Optimal Order Detection (MOOD) on unstructured meshes with very high-order polynomials*, Comput. Fluids 64 (2012) 43–63.

In next paragraph, we sum up the content of the publication and highlight with hindsight the pros and cons of the MOOD method at that time. We then reproduce the paper from the abstract to the conclusion only correcting the misprints and modifying the references to fit the global bibliography.

Summary & Review

The second publication has brought very important improvements to the MOOD method. First we have extended the concept to polygonal 2D meshes and to polynomials of degree up to five leading to the denomination of *very high-order* Finite Volume method. We have moreover refined the ingredients of the MOOD method that implied a clearer and more flexible framework, in particular by introducing the set of detection criteria $\mathcal{A}$.

Then concerning the DMP locking mentioned in previous section, we have noticed that the accuracy discrepancy only occurred at smooth extrema, and introduced a relaxation, namely the $u2$ detection criteria, which allows DMP violation only on these locations. As a result, we have obtained up to 6^{th}-order convergence for the convection equation of smooth solution while discontinuous solutions are prevented from spurious oscillations.

In the context of the hydrodynamics Euler equations, we have provided a comparison of convergence for the isentropic vortex in motion between the MOOD method using primitive and conservative variables for reconstructions. As stated in previous section and in section 1.3.2, the primitive variables version is only second-order. As a consequence all test cases for the Euler equations are carried out with polynomial reconstructions performed on the conservative variables.

Furthermore, the truncation of the maximal degree polynomial has been dropped and substituted by a lower-degree polynomial reconstruction when necessary. Also the *a priori* limitation of reconstructed values at quadrature points has been inserted in the *a posteriori* treatment by verifying the numerical validity of the solution.

Finally we have given precise measurements of the cost of the MOOD method on the so-called *Double Mach reflexion of strong shocks* problem using three computers with different architectures. According to these values, the MOOD method seems to be more efficient than the fast quadrature-free ADER method of [31].

All these improvements have made the MOOD method more robust and efficient but the $u2$ detection criteria was still to be investigated more deeply. Actually two issues were addressed later. First the ε parameter we have proposed, was dependent of the cell size and may be inefficient for large cells. This has been easily solved, and the correction is presented in section 2.2 and in next section. The second point was the extension to the Euler system since we were not able to reach the optimal order of convergence. The investigations led to observe that the decrementing were occurring on the plateaus so that the small curvatures were treated as irregular. An effective correction is presented in 2.3 and next section.

Abstract

This paper extends the MOOD method proposed by the authors in ["A high-order finite volume method for hyperbolic systems: Multi-dimensional Optimal Order Detection (MOOD)", J. Comput. Phys. 230, pp 4028-4050, (2011)], along two complementary axes: extension to very high-order polynomial reconstruction on non-conformal unstructured meshes and new Detection Criteria. The former is a natural extension of the previous cited work which confirms the good behavior of the MOOD method. The latter is a necessary brick to overcome limitations of the Discrete Maximum Principle used in the previous work. Numerical results on advection problems and hydrodynamics Euler equations are presented to show that the MOOD method is effectively high-order (up to sixth-order), intrinsically positivity-preserving on hydrodynamics test cases and computationally efficient.

3.2.1 Introduction

In a recent paper [18], an original high-order method, namely the Multidimensional Optimal Order Detection (MOOD) method, has been introduced to provide up to third-order approximations to hyperbolic scalar or vectorial solutions for two-dimensional geometry. The present article deals with new extensions of the method to general unstructured 2D meshes and to sixth-order convergence in space. Classical high-order reconstructions such as MUSCL or ENO/WENO methods are based on an *a priori* limiting procedure to achieve stability property. The MOOD method follows a fundamentally different way since the limiting procedure (polynomial degree reduction for instance) is achieved *a posteriori* and provides the optimal local polynomial reconstruction which satisfies given stability criteria.

The quest [83] of the (very) high-order schemes starts in the early 70's with the pioneer works of Van-Leer [84] and Kolgan [48, 49, 50]. Since this date, a large literature was dedicated to the limited reconstruction methods for structured and unstructured meshes. Several strategies became very popular due to their intrinsic simplicity such that the MUSCL method [7, 13, 17, 42, 52, 62] or their efficiency to achieve very high-order accuracy such that the ENO/WENO method [1, 45, 33, 57, 68, 70, 92, 40, 41, 66, 95, 82], the Discontinuous Galerkin method

[22, 21, 23, 24], the ADER method [30, 77, 81, 32, 31], the Residual Distribution Scheme [2, 25, 64] and the spectral method [37, 88, 89].

While second-order methods do not require particular cautions, dealing with higher-order methods leads to at least three specific difficulties which, up to our knowledge, are not always clearly identified. First point one should not consider the mean value of a function equivalent to the cell centroid value as it is often done in the MUSCL community. The point is straightforward to overcome but important to notice for newcomers in the field of higher-order numerical schemes. Second point, for vectorial problems the reconstruction process must be done on mean values of the conservative variables and not on non-linear combinations of them. This point is often implied in the classical ENO/WENO papers but is rarely clearly stated and this may mislead newcomers in the high-order community because the order of accuracy discrepancy can be missed depending on the numerical tests used. Contrarily one proposes the isentropic vortex in motion test case to numerically prove that if the primitive variables are used for the reconstruction process then very high-order of accuracy cannot be reached. Third point the Discrete Maximum Principle property on mean values should not be used anymore as a guide line for limitation. We propose in this paper to overcome this difficulty by a new limiting criteria (or Detection Criteria in the MOOD jargon) adapted to provide a full high-order method still maintaining robust stability. Simple examples are introduced within the text when some difficulties related to these points are to be expected.

The basic idea of the MOOD method consists of determining the higher polynomial degree of each local cell still satisfying some stability restrictions. To this end, an iterative process is developed. We perform a local polynomial reconstruction of degree d_i for each cell K_i at the current time t^n and compute a candidate solution at time t^{n+1} without any limiting features. Then a detecting procedure is carried out to check the cells which do not respect the stability criteria and we reduce the local polynomial degree to obtain a better stability. We state that the method is *a posteriori* since the limiting procedure (namely the polynomial degree reduction) is performed after the candidate solution computation. Such a situation is very useful to test the admissibility of the solution. Furthermore, one has to carry out the limiting algorithm if, and only if, it is necessary while the traditional *a priori* method performs unnecessary limitation.

In this paper we propose extensions of the MOOD method which take into account the three difficulties mentioned above. More precisely different detection processes both for the advection and hydrodynamics equations are developed. We numerically prove that these detection processes provide the effective higher-order of accuracy on smooth profiles (up to sixth-order). Moreover we show that for the hydrodynamics equations the method is positivity-preserving by construction and we numerically observe this behavior. The test case have been carried out on non-regular, polygonal and non-conformal meshes and the last test case of the paper show the ability of the MOOD method to simulate complex physics from an experimental set-up of the impact of a shock wave on a cylindrical cavity.

The paper is organized as follows. Section 3.2.2 is dedicated to the generic framework used to describe the MOOD method where the high-order finite volume scheme is presented. Several obstacles to achieve high-order reconstruction are pointed out and the polynomial reconstruction based on the mean value approximation is detailed. In Section 3.2.3, we introduce new criteria to obtain very high-order accurate schemes still preserving local stability. To show the MOOD method efficiency, numerical tests both for the scalar and the vectorial case are carried

out in Section 3.2.4. We mainly focus on the method accuracy and its robustness. We draw some remarks and future developments in the last section.

3.2.2 The MOOD method

3.2.2.1 General concept

The MOOD method is a generic procedure that solves multidimensional hyperbolic system of equations on an unstructured grid in the Eulerian framework. Given different numerical finite volume schemes the MOOD method provides an optimal choice for each computational cell by mitigating accuracy *vs* robustness. From an abstract point of view the MOOD algorithm involves two main ingredients: An ordered list of numerical schemes and a set of constraints with detection criteria which defines the desirable properties the numerical solution should have.

The over-topping numerical scheme represents the *best* scheme one would like to employ. Usually this scheme is the most accurate but less robust one. At the very end of the list lays the least accurate but more robust scheme which is assumed to be satisfactory in all possible situations due to the stabilization effect generated by its intrinsic numerical dissipation. In this paper the list is composed of a robust first-order scheme (an upwind or a Rusanov, HLL, HLLC scheme as instance) while several second or higher-order schemes using polynomial reconstructions compose an ordered list of desirable schemes (see Fig. 3.15 for instance). The second ingredient is the detecting procedure of a set of constraints which determine the local eligibility of the solution for each cell.

We recall that discontinuous solutions may not be handled with high-order reconstructions since local spurious and unphysical oscillations may take place. The low-order numerical scheme should be used to prevent the numerical approximations from oscillating and force to respect some constraints or mathematical properties that depend on equations under consideration. The numerical solution is considered as eligible if it fulfills given properties. As instance the positivity of certain variables such as density or pressure in hydrodynamics equations or the Discrete Maximum Principle for advection equation shall be considered.

In this paper the k-th numerical scheme of the list is a finite volume scheme using unlimited piecewise polynomial reconstruction of degree k. Ultimately this scheme has a $k+1$th-order of accuracy for smooth solutions. Consequently the LO scheme is the generic firs-order finite volume scheme and the HO-1 scheme corresponds to an unlimited MUSCL method.

The core of the MOOD method is a loop over the cells to determine the optimal polynomial degree one can safely use to produce an eligible numerical solution. It amounts to select a numerical method in the ordered list of Fig.3.15.

To this end, given a generic cell K_i and its neighbor cells K_j having edge e_{ij} in common, we first recall two definitions introduced in [18] and then give a new one to extend the MOOD concept:

- d_i is the Cell Polynomial Degree (CellPD) which represents the degree of the polynomial reconstruction on K_i.
- $\mathsf{d}_{ij} = \mathsf{d}_{ji} = \min(\mathsf{d}_i, \mathsf{d}_j)$ are the Edge Polynomial Degrees (EdgePD) corresponding to the degrees of the polynomial reconstructions used to compute approximations of the solution on edge e_{ij}.

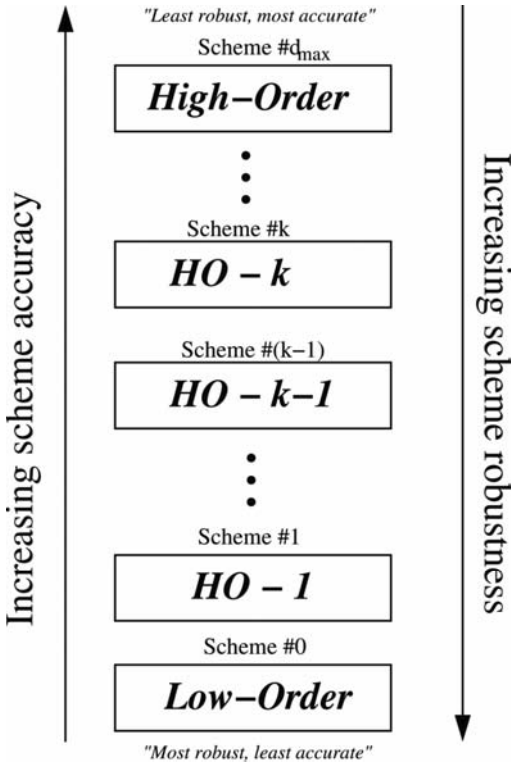


Figure 3.15: Schematic representation of an ordered list of numerical schemes used in the MOOD method. The bottom scheme is the most robust but least accurate one denoted "Low-Order". All over-topping schemes are successively more accurate but less robust. The MOOD method is designed to choose the more adapted scheme for each cell of the computational domain.

– $\mathcal{A}$ is a set of prescribed physical and/or stability constraints. If for each cell K_i the mean values of the numerical solution fulfill the constraints then the numerical solution is said to be $\mathcal{A}$-eligible.

The last item concerns the detecting procedure to distinguish if a candidate solution is eligible according to a set of constraints. In practice we decrement the d_i for any cell K_i which does not respect all the constraints. Such a cell is called *problematic*. Moreover since neighbor cells fluxes may be affected by this process, the decrementing is spread over the direct neighborhood. Such a polynomial degree decrementing for a problematic cell is repeated up to a $\mathsf{d}_i > 0$ for which the set of constraints is fulfilled or to $\mathsf{d}_i = 0$. At that ultimate step the robust and diffusive LO scheme is employed and its first-order solution is always taken as valid. In other words unlike traditional high-order schemes (using *a priori* limiting procedure), we introduce an *a posteriori* detecting procedure where the decision to alter the polynomial degree is carried out after computing the candidate solution.

We finally highlight that such a procedure may be interpreted as a *try and fail* algorithm. Such a generic strategy might be adapted to other classes of method such as the Discontinuous Galerkin method and detect the best polynomial degree in each cell or Finite Element method and detect the most appropriate finite element one can employ in a cell.

3.2.2.2 Framework

Let us consider a generic autonomous hyperbolic equation defined on a domain $\Omega \subset \mathbb{R}^2$, $t > 0$ which casts in the conservative form

$$\partial_t U + \nabla \cdot F(U) = 0, \tag{3.19a}$$
$$U(\cdot,0) = U_0, \tag{3.19b}$$

where $U = U(\mathbf{x}, t)$ is the vector of unknown functions, $\mathbf{x} = (x, y)$ denotes a point of Ω, t is the time, F is the physical flux function and U_0 is the initial condition. Boundary conditions shall be prescribed in the following.

We assume that the computational domain Ω is a polygonal bounded set of $\mathbb{R}^2$ divided into convex polygonal cells K_i, $i \in \mathcal{E}_{el}$, $\mathbf{c}_i$ being the cell centroid and $\mathcal{E}_{el}$ the cell index set. For each boundary edge, $K_i \cap \partial\Omega$, we introduce a virtual cell K_j with $j \notin \mathcal{E}_{el}$ which represents the exterior side of Ω and denote by $\mathcal{E}_{bd}$ the index set of all virtual cells. $\widetilde{\mathcal{E}_{el}} = \mathcal{E}_{el} \cup \mathcal{E}_{bd}$ is the index set of all cells. This notation avoids a special treatment for boundary edges in the scheme, and provides a natural notation for ghost cells should they exist or not.

For each cell K_i, one denotes by e_{ij} the common edge between K_i and K_j, with $j \in \underline{\nu}(i) \subset \widetilde{\mathcal{E}_{el}}$, $\underline{\nu}(i)$ being the index set of all the elements which share an edge with K_i. The extended neighborhood is represented by the index set $\overline{\nu}(i) \subset \widetilde{\mathcal{E}_{el}}$ of all K_j such that $K_i \cap K_j \neq \emptyset$ (see Fig. 3.16).

Moreover $|K_i|$ and $|e_{ij}|$ measure the surface of K_i and the length of e_{ij} respectively while $\mathbf{n}_{ij}$ is the unit outward normal vector to e_{ij} pointing from K_i to K_j. At last, q_{ij}^r, $r = 1, ..., R$ represent the Gaussian quadrature points employed for numerical integration on edge e_{ij}.

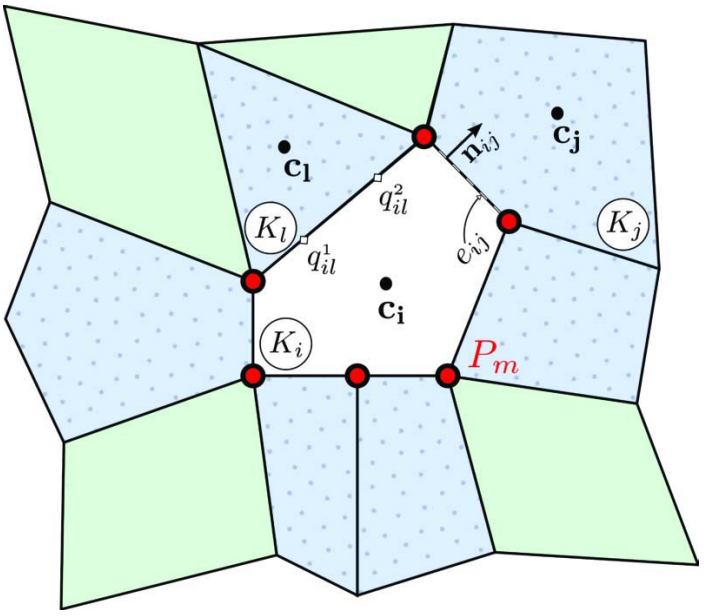


Figure 3.16: Mesh notation. Index set $\underline{\nu}(i)$ corresponds to blue cells with dots, $\overline{\nu}(i)$ corresponds to non-white cells.

The generic first-order explicit finite volume scheme is given by

$$U_i^{n+1} = U_i^n - \Delta t \sum_{j \in \underline{\nu}(i)} \frac{|e_{ij}|}{|K_i|} \mathbb{F}(U_i^n, U_j^n, \mathbf{n}_{ij}), \tag{3.20}$$

where $\mathbb{F}(U_i^n, U_j^n, \mathbf{n}_{ij})$ is a numerical flux which satisfies the classical properties of consistency and monotonicity. To provide higher-order accuracy, we substitute in equation (3.20) the first-order approximation U_i^n and U_j^n with better approximations of U at the quadrature points of edge e_{ij} leading to the generic spatial high-order finite volume scheme

$$U_i^{n+1} = U_i^n - \Delta t \sum_{j \in \underline{\nu}(i)} \frac{|e_{ij}|}{|K_i|} \sum_{r=1}^{R} \xi_r \mathbb{F}(U_{ij,r}^n, U_{ji,r}^n, \mathbf{n}_{ij}), \tag{3.21}$$

where $U_{ij,r}^n$ and $U_{ji,r}^n$, $r = 1, ..., R$ are high-order approximations of U at quadrature points $q_{ij}^r \in e_{ij}$, $r = 1, ..., R$ on both sides of edge e_{ij} and ξ_r denote the quadrature weights.

For the sake of simplicity, let us write the scheme under the compact form

$$U_h^{n+1} = U_h^n + \Delta t \, \mathcal{H}^R(U_h^n), \tag{3.22}$$

with $U_h^n = \sum_{i \in \mathcal{E}_{el}} U_i^n \mathbb{1}_{K_i}$ the constant piecewise approximation of function U and operator $\mathcal{H}^R$ being defined as

$$\mathcal{H}^R(U_h^n) := - \sum_{i \in \mathcal{E}_{el}} \left(\sum_{j \in \underline{\nu}(i)} \frac{|e_{ij}|}{|K_i|} \sum_{r=1}^{R} \xi_r \mathbb{F}(U_{ij,r}^n, U_{ji,r}^n, \mathbf{n}_{ij}) \right) \mathbb{1}_{K_i}. \tag{3.23}$$

Finally to provide a high-order method in time, we use the third-order TVD Runge-Kutta method (RK3, see [70]) which corresponds to a convex combination of three explicit steps

$$U_h^{n+1} = \frac{U_h^n + 2U_h^{(3)}}{3} \quad \text{with} \quad \left\{ \begin{array}{lcl} U_h^{(1)} & = & U_h^n + \Delta t \, \mathcal{H}^R(U_h^n) \\ U_h^{(2)} & = & U_h^{(1)} + \Delta t \, \mathcal{H}^R(U_h^{(1)}) \\ U_h^{(3)} & = & \widehat{U}_h^{(2)} + \Delta t \, \mathcal{H}^R(\widehat{U}_h^{(2)}) \end{array} \right. \tag{3.24}$$

where $\widehat{U}_h^{(2)}$ is the convex combination $(3U_h^n + U_h^{(2)})/4$.

3.2.2.3 Arbitrary degree polynomial reconstruction

In the introduction we have reminded one classical obstacle to reach higher-order of accuracy when polynomial reconstruction is to be used. It is well-known that the mean value U_i of a regular function U on K_i is approximated by the value of the solution at the cell centroid, $U(\mathbf{c}_i)$, with an error of $O(h^2)$ where h represents the characteristic length of the cell. It results that any reconstruction based on geometrical arguments using $U(\mathbf{c}_i)$ in place of U_i can only provide second-order approximation.

Therefore as classical higher-order finite volume methods the MOOD method is based on polynomial reconstructions from mean values on cells. Let us consider a generic reconstructed polynomial of degree d, given mean values U on a generic cell K, under the form

$$\widetilde{u}(\mathbf{x};\mathsf{d}) = U + \sum_{1\leq|\boldsymbol{\alpha}|\leq\mathsf{d}} \mathcal{R}_{\boldsymbol{\alpha}} \left((\mathbf{x}-\mathbf{c})^{\boldsymbol{\alpha}} - \frac{1}{|K|}\int_K (\mathbf{x}-\mathbf{c})^{\boldsymbol{\alpha}}\, d\mathbf{x} \right), \tag{3.25}$$

where $\mathbf{c}$ is the centroid of K, $\mathbf{x}$ a generic point in K and $\mathcal{R}_{\boldsymbol{\alpha}}$ are the unknowns polynomial coefficients where $\boldsymbol{\alpha} = (\alpha_x, \alpha_y) \in \mathbb{N}^2$ is a multi-index with $|\boldsymbol{\alpha}| = \alpha_x + \alpha_y$. Note that by construction, the mean value on K of the polynomial function is equal to U since the integral over K of the term between parenthesis in (3.25) vanishes. It thus fulfills the conservation property on K.

There exist several techniques [1, 33] to determine the coefficients $\mathcal{R}_{\boldsymbol{\alpha}}$. Here, we consider a least square approximation of neighbor mean values U_j where K_j belongs to a compact stencil $\mathcal{S}(K)$. It amounts to minimizing the functional

$$E(\widetilde{u}) = \sum_{j\in\mathcal{S}(K)} \omega_k \left[\frac{1}{|K_j|}\int_{K_j} \widetilde{u}\, d\mathbf{x} - U_j \right]^2, \tag{3.26}$$

where ω_k are positive weights used to provide a better condition number. In particular, the condition number of the associated linear system dependents on the spatial characteristic length thus we use the solution proposed in [33] to overcome this problem.

In practice, we do not directly solve the symmetric linear system associated with the minimization problem. Instead we use the technique from [9, 57] where an over-determined linear system is solved in a least-squares sense with a QR decomposition using Householder transformations. The reconstructed polynomial $\widetilde{u}$ is thus exact for any polynomial function of degree lower than d which provides the consistency of the reconstruction method and further the status of a $(\mathsf{d}+1)^{th}$-order numerical method.

Remark 3.13 *In 2D, at least* $\mathcal{N}(\mathsf{d}) = (\mathsf{d}+1)(\mathsf{d}+2)/2 - 1$ *neighbors are needed to provide the minimal number of equations. However for the sake of robustness more cells are involved. In details, we use at least 5 cells for* $\mathsf{d}=1$*, 8 cells for* $\mathsf{d}=2$*, 16 cells for* $\mathsf{d}=3$*, 20 cells for* $\mathsf{d}=4$ *and 26 cells for* $\mathsf{d}=5$*.*

Remark 3.14 *In the introduction we have stated that in the general case one should not identify the mean value of a non-linear combination with the non-linear combination of mean values. Let* ρ *and* ϕ *be two regular functions on cell* K_i *and* ρ_i*,* ϕ_i*,* $(\rho\phi)_i$*, denote their respective exact mean values. A Taylor expansion with respect to the centroid of the cell gives* $(\rho\phi)_i = \rho_i\phi_i + O(h^2)$*. For instance let us consider the one-dimensional variables* ρ, ϕ *and* $(\rho\phi)$ *and their mean values on cell* $K_1 = [0, h]$

$$\rho(x) = 1 + x, \qquad \phi(x) = 1 - x, \qquad (\rho\phi)(x) = 1 - x^2,$$
$$\rho_1 = 1 + \frac{h}{2}, \qquad \phi_1 = 1 - \tfrac{h}{2}, \qquad (\rho\phi)_1 = 1 - \frac{h^2}{3}.$$

We then deduce that $|(\rho\phi)_1 - \rho_1\phi_1| = h^2/12$ *leading to a second-order error. As instance it is well known that for Euler system of equations the non-linear transformation of the conservative mean values into primitive ones introduces a second-order error in the general case.*

3.2.2.4 Algorithm

Let us assume that we have access to a given sequence $U_h^n = (U_i^n)_{i\in\mathcal{E}_{el}}$ of mean value approximations at time t^n, the goal is to build an eligible sequence $U_h^{n+1} = (U_i^{n+1})_{i\in\mathcal{E}_{el}}$ at time $t^{n+1} = t^n + \Delta t^n$ in the sense that each approximation U_i^{n+1} respects a set of constraints $\mathcal{A}$. We only consider here a forward Euler time step without loss of generality. The MOOD method algorithm is the following:

1. *Initialization at t^n.* The MOOD procedure starts by initializing the CellPD to $\mathtt{d}_i = \mathtt{d}_{max}$ and by computing the coefficients of the polynomial reconstruction $\widetilde{U}_i(\mathbf{x};\mathtt{d}_i)$ on each cell.
2. *Evaluation of EdgePD and values at Gauss points.* We compute the EdgePD $\mathtt{d}_{ij}$ on each edge and use polynomial function $\widetilde{U}_i(\mathbf{x};\mathtt{d}_{ij})$ and $\widetilde{U}_j(\mathbf{x};\mathtt{d}_{ij})$ to compute approximations of U at Gaussian points on e_{ij}.
3. *Computation of candidate solution $U_h^\star$.* Numerical fluxes are computed using the reconstructed solution at Gauss points and one time step is carried out to provide a candidate $U_h^\star$ at time $t^{n+1} = t^n + \Delta t^n$.
4. *Check $U_i^\star$ for $\mathcal{A}$-eligibility.* If $\mathtt{d}_i \neq 0$ we check the $\mathcal{A}$-eligibility of each mean value $U_i^\star$ with respect to the constraints set $\mathcal{A}$. In the case $U_i^\star$ is not $\mathcal{A}$-eligible then CellPD $\mathtt{d}_i$ is decremented. If all cells are $\mathcal{A}$-eligible then the candidate solution is valid and we set $U_h^{n+1} = U_h^\star$ else the solution is recomputed following steps 2., 3. and 4.

Remark 3.15 *Only cells K_i where CellPD has been decremented and their neighbors in the compact stencil $\underline{\nu}(i)$ have to be re-updated. Consequently only these cells will have to be checked for the next iterations of the MOOD procedure within the current time step. This dramatically reduces computational cost.*

Remark 3.16 *Since polynomial reconstruction is costly in CPU time and memory, we proposed in [18] to truncate $\widetilde{U}_i(\div\mathtt{d}_{max})$ to obtain lower-order polynomials. However we found that for $\mathtt{d}_{max} > 2$ this technique implies non desirable behavior on discontinuous profiles as the reconstruction stencil remains large.*
Moreover numerical experiments show that a one-by-one degree decrementing leads to avoidable computational effort since the decrementing procedure is usually performed around discontinuities. We thus slightly modify the decrementing algorithm by jumping from $\mathtt{d} = \mathtt{d}_{max}$ to $\mathtt{d} = 2$ and then from $\mathtt{d} = 2$ to $\mathtt{d} = 0$ if needed. This also reduces the computational effort while providing equivalent results on a wide range of test cases compared to a one-by-one decrementing.

Remark 3.17 *Polynomial reconstruction on boundary cells are treated using ghost cells in order to to be consistent with the prescribed boundary conditions.*

The major difficulty remains to determine a list of constraints which both provides a very high accurate solution while avoids numerical artifacts such as spurious oscillations in the vicinity of discontinuity. This is the purpose of the next section.

3.2.3 Detection process

The list of constraints $\mathcal{A}$ corresponds to eligible criteria that the numerical approximation has to fulfill. To this end, detection process is necessary to list where the candidate numerical

solution fails to respect the constraints. Such process must be very carefully designed to preserve high accuracy for regular solutions whereas discontinuities should be treated with the lower order scheme to avoid non-physical oscillations. The first subsection deals with the advection problem and a new detection process called $u2$ and based on a smoothness detector. In the second subsection the Euler system is considered: Two detection processes are proposed and we show the positivity-preserving property of the MOOD method.

3.2.3.1 Advection problem: The $u2$ detection process

Solutions of autonomous scalar hyperbolic problems satisfy the Maximum Principle property. Such a property is also valid for advection problem with divergence free velocity. Therefore the Discrete Maximum Principle (DMP) seems to be a good candidate to detect problematic cells. Unfortunately, as mentioned in the introduction, the strict DMP applied to mean values reduces the order of accuracy to two (see the appendix for an example), and thus can not be used alone. Classical studies show that the accuracy discrepancy only occurs at extrema [65, 59, 46]. We will then mainly focus on extrema since the DMP detection process is still relevant where the solution is locally monotone. We propose the relaxation of the strict DMP at smooth extrema in order to avoid accuracy discrepancy. This leads to the introduction of an additional procedure to detect smooth extrema. Notice that in (W)ENO type of methods the DMP is not strictly enforced which implies that extrema are well approximated and consequently arbitrary high-order of accuracy is achieved.
The first detection criteria is the DMP: No polynomial degree decrementing is performed for cells where the DMP is satisfied. Let us now consider a cell K_i where $U_i^\star$ does not fulfill the DMP. Two situations may arise whether we deal with a discontinuity or a smooth extrema. The major difficulty is to give a concrete definition of the concept of a *smooth extrema* from a numerical point of view. Actually a function may be considered irregular for a coarse mesh but regular with a finer one. We try to overcome this difficulty by introducing the following definition.

Definition 3.18 *Let K_i be a cell and $\widetilde{u}_i = \widetilde{u}_i(.;2)$ a polynomial reconstruction of degree 2 for an underlying function U. We define the second derivatives in x and y directions by $\mathcal{X}_i = \partial_{xx}\widetilde{u}_i \in \mathbb{R}$ and $\mathcal{Y}_i = \partial_{yy}\widetilde{u}_i \in \mathbb{R}$. We will refer to these second derivatives as "curvatures".*
For all cell K_j, $j \in \underline{\nu}(i)$, we define the maximal and minimal curvatures as

$$\mathcal{X}_i^{min} = \min_{j\in\underline{\nu}(i)} (\mathcal{X}_i, \mathcal{X}_j) \quad and \quad \mathcal{X}_i^{max} = \max_{j\in\underline{\nu}(i)} (\mathcal{X}_i, \mathcal{X}_j)\,,$$

$$\mathcal{Y}_i^{min} = \min_{j\in\underline{\nu}(i)} (\mathcal{Y}_i, \mathcal{Y}_j) \quad and \quad \mathcal{Y}_i^{max} = \max_{j\in\underline{\nu}(i)} (\mathcal{Y}_i, \mathcal{Y}_j)\,.$$

We now introduce the new detection criterion to select smooth extrema.

Definition 3.19 *A numerical solution $U_i^\star$ in cell K_i which violates the DMP is nonetheless eligible if*

$$\mathcal{X}_i^{max}\mathcal{X}_i^{min} > 0 \quad and \quad \mathcal{Y}_i^{max}\mathcal{Y}_i^{min} > 0, \tag{3.27}$$

$$\frac{|\mathcal{X}_i^{min}|}{|\mathcal{X}_i^{max}|} \geq 1-\varepsilon_i \quad and \quad \frac{|\mathcal{Y}_i^{min}|}{|\mathcal{Y}_i^{max}|} \geq 1-\varepsilon_i, \tag{3.28}$$

where ε_i is a cell dependent parameter defined by

$$\varepsilon_i = (\Delta x_i)^{\frac{1}{2m}}, \ \textit{with} \ \ \Delta x_i = |K_i|^{\frac{1}{m}},$$

m being the spatial dimension ($m = 2$ here).

Such a detection criterion is motivated by the following considerations. For a given mesh, the solution is locally considered as non-oscillating if condition (3.27) is fulfilled meaning that, at the numerical level, the "curvatures" of the $\mathbb{P}_2$ approximation have the same sign.
Moreover for a given mesh, the solution is considered locally C^2 from a numerical point of view if condition (3.28) is fulfilled. The parameter ε is a mesh dependent coefficient which prescribes the tolerance. Such criteria verifies if the "curvatures" are almost identical in the vicinity of cell K_i with respect to the local characteristic space length Δx_i.
The choice of ε derives from numerous tests. In fact our numerical experiments have shown that ε scales like a cell dependent characteristics length to a power depending on the dimension of space (tests have been carried out in 1D and 2D). It seems to the authors to be the best compromise to gain a very high-order of convergence while maintaining reasonable monotonicity. Finally we remark that at the limit $\varepsilon = 0$ we recover the DMP.
The set of constraints $\mathcal{A}$ for advection equation is thus constituted by the DMP relaxed by the smooth extrema detector described above. The detection process is called $u2$ detection in reference to the second-order derivatives and is summarized in the sequel.
Being given a sequence $U_h^\star = (U_i^\star)_{i\in\mathcal{E}_{el}}$, the $u2$ detection procedure in the case of the advection problem is given by the following algorithm.

1. The DMP criterion is first checked on each cell K_i

$$\min_{j\in\overline{\nu}(i)}(U_i^n, U_j^n) \leq U_i^\star \leq \max_{j\in\overline{\nu}(i)}(U_i^n, U_j^n). \tag{3.29}$$

2. If $U_i^\star$ does not satisfy (3.29) then
   a- Compute $\mathcal{X}_k, \mathcal{Y}_k$ for $k \in \underline{\nu}(i)\bigcup\{i\}$ and coefficient ε_i,
   b- Check criteria (3.27) and (3.28). If cell i is not a smooth extrema then d_i is decremented, else $U_i^\star$ is eligible.

3.2.3.2 Euler system: Two detection processes and positivity-preserving

The compressible hydrodynamics Euler system of equations is the following hyperbolic unsteady non-linear system involving conservation of mass, momentum and total energy

$$\partial_t \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix} + \partial_x \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix} + \partial_y \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix} = 0. \tag{3.30}$$

The primitive variables are the density ρ, the velocity $\mathbf{U} = (u, v)$ and the pressure p. The pressure is linked to two thermodynamical variables such as density and specific internal energy ε through an Equation Of State (EOS) $p = p(\rho, \varepsilon)$. As instance the classical ideal gas law states

that $p = (\gamma - 1)\rho\varepsilon$ where γ is the ratio of specifics heats. Moreover the total energy E is such that $E = \rho(\varepsilon + 1/2\|\mathbf{U}\|^2)$.
Even if the DMP property is used in most of limiting procedures (MUSCL technique as instance), the DMP property does not make sense in the case of the Euler system, for the density or the total energy for instance, since the velocity is not divergence free. Consequently we can not rely only on DMP. We propose here two detecting procedures which we have been widely experimented and present in the next sections the pros and cons of such procedures.

⋆ Physical Admissible Detection (PAD)

The first and minimal detection criteria consists of ensuring the physical meaningfulness of the primitive variables, namely positivity of density and pressure.
Then the set of constraints $\mathcal{A}$ are used to test if the candidate solution satisfies $\rho_i^\star > 0$ and $p_i^\star > 0$. Note that $p_i^\star$ is not a conservative variable and derives from nonlinear combinations of conservative ones. The PAD algorithm is the following.

1. The Physical Admissibility criterion is first checked on each cell K_i

$$\rho_i^\star > 0, \quad p_i^\star > 0. \tag{3.31}$$

2. If the PAD criterion is not satisfied then d_i is decremented, else U_i^* is eligible.

The PAD procedure only consists of maintaining the physical meaningfulness of the numerical approximation. In other words, the high-order MOOD method coupled with the PAD Detection Process is positivity-preserving for density and pressure. This point is further discussed in section 3.2.3.2.

⋆ Extension of the $u2$ detection process

Physical admissibility of the solution is not enough to prevent oscillations in the vicinity of discontinuities. It is a precondition but we require an supplementary detection criterion to decide whether the numerical solution is locally smooth or not. To this end, we adapt the $u2$ criterion to the density variable using local $\mathbb{P}_2$ polynomial reconstruction $\widetilde{\rho}_i = \widetilde{\rho}_i(.;2)$ to evaluate $\mathcal{X}_i = \partial_{xx}\widetilde{\rho}_i$ and $\mathcal{Y}_i = \partial_{yy}\widetilde{\rho}_i$. The $u2$ detection algorithm for the Euler system is thus the following.

1. The PAD criterion is first checked on each cell K_i. If it is not satisfied then d_i is decremented and Steps 2. and 3. are skipped.
2. The DMP criterion of the density function is checked on each cell K_i

$$\min_{j\in\overline{\nu}(i)}(\rho_i^n, \rho_j^n) \leq \rho_i^\star \leq \max_{j\in\overline{\nu}(i)}(\rho_i^n, \rho_j^n). \tag{3.32}$$

3. If $\rho_i^\star$ does not satisfy (3.32) then
   a- Compute $\mathcal{X}_k, \mathcal{Y}_k$ for $k \in \underline{\nu}(i)\bigcup\{i\}$ and coefficient ε_i,
   b- Check criteria (3.27) and (3.28). If cell i is not a smooth extrema then d_i is decremented for any conservative variable, else U_i^* is eligible.

The set of constraints $\mathcal{A}$ consists of the PAD, and the $u2$ detection process on the density. Note that the density is thus the variable onto which the detection is performed. However there is a large number of possible choices of detection variables and decrementing procedures.

⋆ **Positivity-preserving property**

One important property a scheme must fulfill is to be positivity-preserving, that is given a set of physically admissible mean values the scheme provides another set of physically admissible ones. It is absolutely mandatory for the simulation to continue. In the case of the Euler equations density and pressure must be positive but this is not straightforwardly ensured by most of classical MUSCL or ENO/WENO schemes and most of simulation codes need a special treatment when the positivity is violated. Indeed designing a positivity-preserving scheme may be a difficult task and often leads to a more complex scheme because of the classical *a priori* limitation philosophy. This classical difficulty is stated by the authors in [96] page 2754 as " It is very difficult to design a conservative high-order accurate scheme preserving the positivity". However the *a posteriori* treatment implies that the MOOD method is intrinsically positivity-preserving assuming the three following points:

1. The lowest order scheme is positivity-preserving, in our case it is the first-order finite volume one.
2. The positivity of density and pressure are parts of the set of constraints $\mathcal{A}$.
3. The EdgePD strategy is upper-limiting see [18] definition 9 page 4033. This implies that if the CellPD of a given cell is 0 then this cell is fully updated with the first-order scheme.

The proof that the MOOD method is positivity-preserving is analogous to the one in theorem 10 page 4033 of [18]. In short, given a candidate solution one checks the positivity of density and pressure. If a cell is problematic that is to say density or pressure is negative then the CellPD is decremented. The next candidate solution is computed and checked again: Either this next candidate is positive or the decrementing process carries on until the CellPD is zero. In this latter case points 1. and 3. necessarily imply the positivity of the candidate solution. As this process is the same for any cell it leads to a positivity-preserving solution in a finite number of MOOD iterations.
In the numerical section we propose the Noh test case for which our implementation of the classical MUSCL scheme generates a negative pressure and fails to complete the simulation whereas the MOOD method always gives a physical meaningful solution.

3.2.4 Numerical tests

MOOD has been implemented into a 2D unstructured (polygonal) code which can deal with advection equation and hydrodynamics equations. The polynomial reconstruction ranges from piecewise constant up to piecewise polynomial of fifth degree. Following remark 3.16 one uses two decrementing sequences: $\mathbb{P}_5$-$\mathbb{P}_2$-$\mathbb{P}_0$ and $\mathbb{P}_3$-$\mathbb{P}_2$-$\mathbb{P}_0$. It implies that only two precomputed matrices for the reconstruction step per cell are only stored in memory for $\mathsf{d} = \mathsf{d}_{max}$ and $\mathsf{d} = 2$. The flux computation involves integrals which are approximated using Gaussian numerical integration. We use two Gaussian points on edges for $\mathbb{P}_2$ and $\mathbb{P}_3$ reconstructions and three

for $\mathbb{P}_5$ to reach the expected order of accuracy for numerical integrations. Time integration is performed with the RK3-TVD method given by system (3.24). We apply the MOOD procedure detailed in section 3.2.2 to each sub-step of the RK3-TVD. The CellPD are thus reinitialized to d_{max} at the beginning of each time sub-step. By default we use classical time step control with CFL=0.6. In the case of convergence study we use a fixed time step $\Delta t = \Delta x^{r/3}$ to reach r^{th}-order of accuracy. Given a variable φ the relative L^1 and L^∞ errors are measured by:

$$err_1 = \frac{\sum_{i\in\mathcal{E}_{el}} |\varphi_i^N - \varphi_i^0||K_i|}{\sum_{i\in\mathcal{E}_{el}} |\varphi_i^0||K_i|} \quad \text{and} \quad err_\infty = \frac{\max_{i\in\mathcal{E}_{el}} |\varphi_i^N - \varphi_i^0|}{\max_{i\in\mathcal{E}_{el}} |\varphi_i^0|},$$

where $(\varphi_i^0)_i$ and $(\varphi_i^N)_i$ are respectively the cell mean values at initial time $t = 0$ and final time $t = t_{\text{final}} = N\Delta t$.
The unstructured meshes used in this paper are of different kinds, logically rectangular, Delaunay triangulation, Voronoi tessellation and non-conformal polygonal mesh. Contrarily to what was done in [18] the whole detection is made *a posteriori*, namely we do not check if the reconstructed values at Gauss points are physically admissible or not. If they are not, the flux and the cell mean values are usually undefined therefore the cell is flagged as problematic.

3.2.4.1 Advection equation

Let us consider the scalar linear advection of a quantity u with velocity $V(\mathbf{x})$

$$\left\{ \begin{array}{rcl} \partial_t u + \nabla.(Vu) & = & 0, \\ u(.,t=0) & = & u^0, \end{array} \right. \tag{3.33}$$

where $V(\mathbf{x})$ is a continuous function on $\Omega \in \mathbb{R}^2$ and u^0 is the initial condition. Boundary conditions are prescribed as periodic ones on $\partial\Omega$.
The Double Sine Translation (DST) is first tested on Delaunay triangulations and Voronoi tessellations in order to prove that on smooth solution MOOD can actually maintain very high-order of accuracy with the $u2$ detection criteria. Only second-order of accuracy is reached when DMP detection criterion is used. The second test is the Solid Body Rotation (SBR) that is used to prove that MOOD-$u2$ can preserve smooth extrema but can still limit discontinuous profiles. This problem is further used to show the improvement obtained when polynomial reconstruction degree is increased, in other word when high-($\mathbb{P}_1$) and very high-order ($\mathbb{P}_3$, $\mathbb{P}_5$) numerical schemes are used.

⋆ Double Sine Translation (DST)

Let Ω be the unit square. We consider a constant velocity $V = (2, 1)$ and the C^∞ initial condition

$$u^0(x, y) = \sin(2\pi x)\sin(2\pi y).$$

The final time is $t_{\text{final}} = 2.0$. Periodic boundary conditions imply that the exact final solution coincides with the initial one. The solution is therefore always smooth during the computation. The computations are carried out on series of successively refined Delaunay triangulations (from 456 up to 29184 cells, see an example in Fig. 3.17 left panel) and polygonal Voronoi tessellations

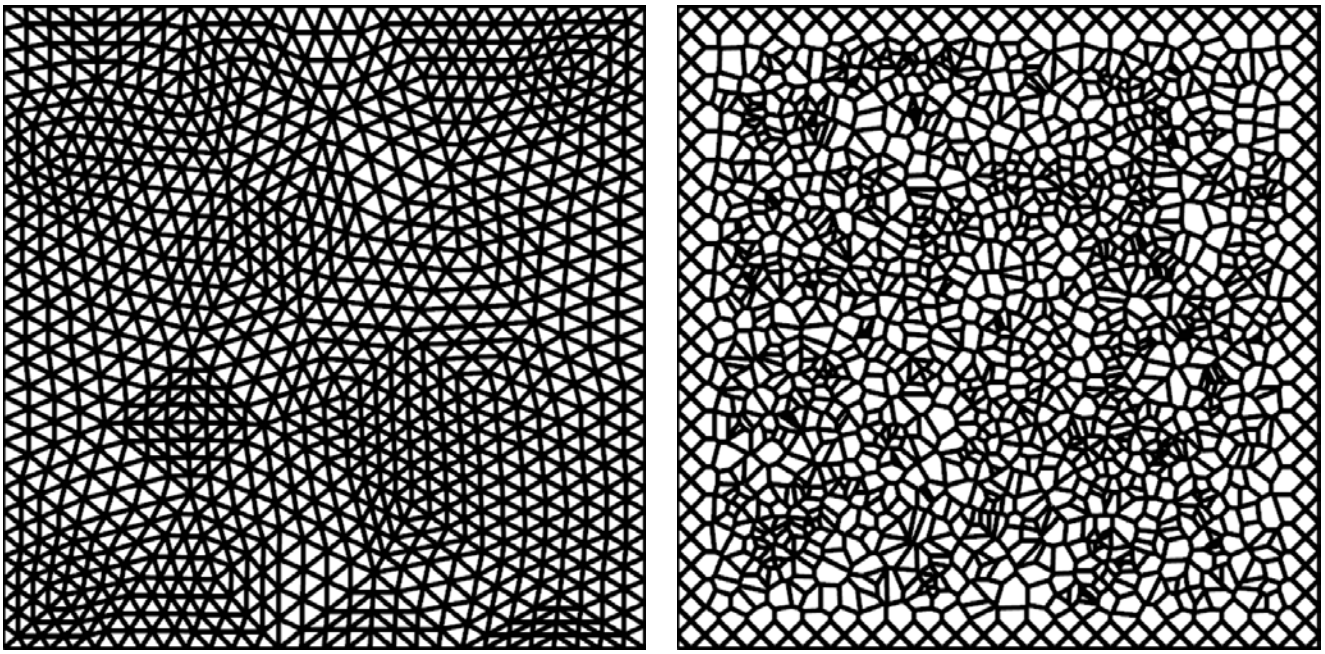


Figure 3.17: Example of Delaunay (left) and Voronoi (right) meshes for the DST problem.

(from 300 up to 19200 cells, see Fig. 3.17 right panel). Note that the meshes are far from being regular, see right panel of Fig. 3.17 fro instance. We plot in Fig. 3.18 the convergence curves obtained on the series of Delaunay triangulations and Voronoi tessellations. The MOOD method with the DMP detection process is displayed on top panels whereas the $u2$ Detection Process is on bottom panels. It clearly shows the strong limitation implied by the DMP since only 3^{rd}-order and 2^{nd}-order are reached in L^1 and L^∞ norms respectively independently of the polynomial degree. On the contrary the proposed $u2$ Detection Process reaches the expected order of convergence. This is actually explained by the fact that only polynomials of maximal degree are used during the whole computation, *i.e.* no CellPD decrementing is ever recorded. L^1 and L^∞ errors and rates are given in Table 3.11 for the DMP and the $u2$ detection criteria. One observes that the optimal order of convergence is reached for the $u2$ detection criterion whereas only second-order accurate results are obtained when the DMP is used.
This accuracy test on smooth functions is passed by the MOOD method with $u2$ Detection Process, the next section is thus dedicated to the study of its behavior on non-smooth profiles.

⋆ Solid Body Rotation (SBR)

First introduced by R.J. Leveque in [52], the Solid Body Rotation test on the unit domain consists of one rotation of three shapes: a hump, a cone and a slotted cylinder. Each shape is located within a circle of radius $r^0 = 0.15$

Hump centered at $(x^0, y^0) = (0.25, 0.5)$

$$u^0(x, y) = \frac{1}{4}(1 + \cos(\pi \min(r(x, y), 1))).$$

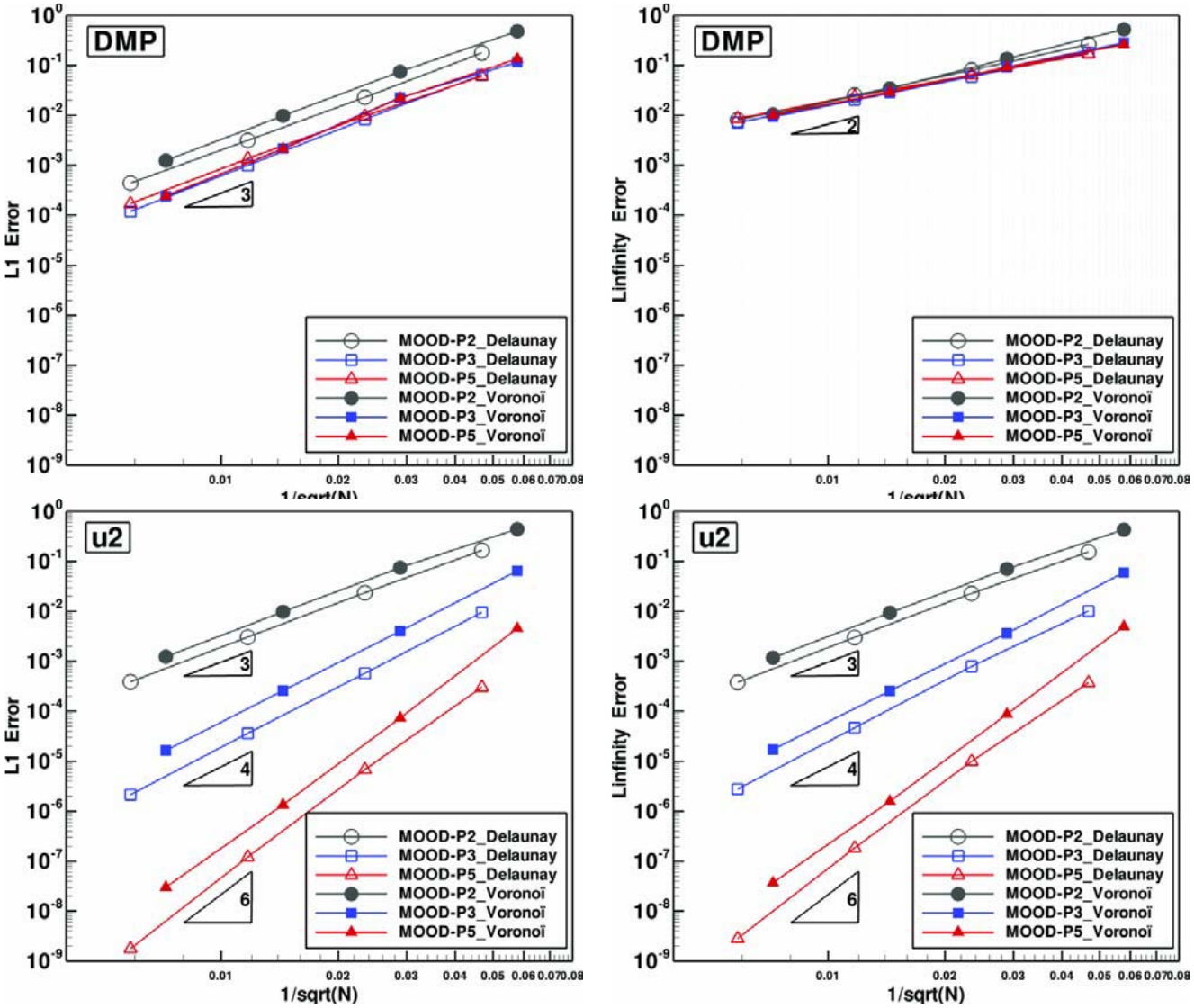


Figure 3.18: Error curves for the DST problem for series of Delaunay meshes (empty symbols) and of Voronoi meshes (filled symbols) for the DMP detection process (top) and the $u2$ one (bottom).

Cone centered at $(x^0, y^0) = (0.5, 0.25)$

$$u^0(x, y) = 1 - r(x, y).$$

Slotted cylinder centered at $(x^0, y^0) = (0.5, 0.75)$

$$u^0(x, y) = \begin{cases} 1 & \text{if } |x - 0.5| < 0.25, \text{ or } y > 0.85, \\ 0 & \text{elsewhere,} \end{cases}$$

where $r(x, y) = \dfrac{1}{r^0}\sqrt{(x - x^0)^2 + (y - y^0)^2}$. To perform the rotation, we use the velocity $V(\mathbf{x}) = (-y + 0.5, x - 0.5)$ and the final time $t_{\text{final}} = 2\pi$ corresponds to one full rotation.

		DMP detec. process				$u2$ **detec. process**			
Deg./Type	Cell Nb	L^1 error		L^∞ error		L^1 error		L^∞ error	
$\mathbb{P}_2$/Delaunay	456	1.775E-01	—	2.629E-01	—	1.656E-01	—	1.549E-01	—
	1824	2.303E-02	2.95	8.016E-02	1.71	2.351E-02	2.82	2.283E-02	2.76
	7296	3.142E-03	2.87	2.522E-02	1.67	3.049E-03	2.95	2.995E-03	2.93
	29184	4.391E-04	2.84	8.082E-03	1.64	3.870E-04	2.98	3.784E-04	2.98
$\mathbb{P}_2$/Voronoi	300	4.804E-01	—	5.278E-01	—	4.423E-01	—	4.339E-01	—
	1200	7.483E-02	2.68	1.359E-01	1.96	7.482E-02	2.56	7.070E-02	2.62
	4800	9.779E-03	2.94	3.432E-02	1.99	9.788E-03	2.93	9.348E-03	2.92
	19200	1.244E-03	2.97	1.039E-02	1.72	1.233E-03	2.99	1.176E-03	2.99
Expected order			3		3		3		3
$\mathbb{P}_3$/Delaunay	456	6.383E-02	—	1.801E-01	—	9.474E-03	—	1.007E-02	—
	1824	8.369E-03	2.93	5.920E-02	1.61	5.751E-04	4.04	7.916E-04	3.67
	7296	9.916E-04	3.08	2.057E-02	1.53	3.611E-05	3.99	4.664E-05	4.09
	29184	1.185E-04	3.06	7.146E-03	1.53	2.140E-06	4.08	2.774E-06	4.07
$\mathbb{P}_3$/Voronoi	300	1.158E-01	—	2.826E-01	—	6.431E-02	—	5.961E-02	—
	1200	2.263E-02	2.36	9.234E-02	1.61	4.017E-03	4.00	3.632E-03	4.04
	4800	2.157E-03	3.39	2.787E-02	1.73	2.583E-04	3.96	2.539E-04	3.84
	19200	2.393E-04	3.17	9.295E-03	1.58	1.649E-05	3.97	1.718E-05	3.89
Expected order			4		4		4		4
$\mathbb{P}_5$/Delaunay	456	6.098E-02	—	1.691E-01	—	3.034E-04	—	3.715E-04	—
	1824	9.660E-03	2.66	6.383E-02	1.41	6.796E-06	5.48	9.939E-06	5.22
	7296	1.359E-03	2.83	2.399E-02	1.41	1.207E-07	5.82	1.831E-07	5.76
	29184	1.704E-04	3.00	8.574E-03	1.48	1.767E-09	6.09	2.836E-09	6.01
$\mathbb{P}_5$/Voronoi	300	1.352E-01	—	2.610E-01	—	4.584E-03	—	4.955E-03	—
	1200	2.213E-02	2.61	9.116E-02	1.52	7.327E-05	5.97	8.740E-05	5.83
	4800	2.119E-03	3.38	2.914E-02	1.65	1.341E-06	5.77	1.573E-06	5.80
	19200	2.449E-04	3.11	1.005E-02	1.54	3.017E-08	5.47	3.703E-08	5.41
Expected order			6		6		6		6

Table 3.11: L^1 and L^∞ errors and convergence rate for the DST problem for the MOOD method with DMP and $u2$ detection process.

For this test case we use a genuinely unstructured and non-uniform mesh made of 5190 triangles see Fig. 3.19 where we also display the initial data in isolines view, see also Fig. 3.20 top-left panel where a side view of the initial data is provided. This mesh is refined around the slotted disk, the ratio between the largest and smallest edge length is approximately 7. The three shapes while rotating move across the refined and coarse zones. The purpose is to emphasize the effects on the numerical results of using a truly non-regular mesh.

We plot in Fig. 3.20 profile views of the solution obtained from three methods but all with a $\mathbb{P}_5$ polynomial reconstruction. First the MOOD method with the DMP Detection Process, then the MOOD method with the $u2$ Detection Process, and finally the unlimited version of the FV scheme. These results show on one hand that the solution with $u2$ Detection Process on the non-smooth slotted cylinder is almost the same as for the DMP. On the other hand it

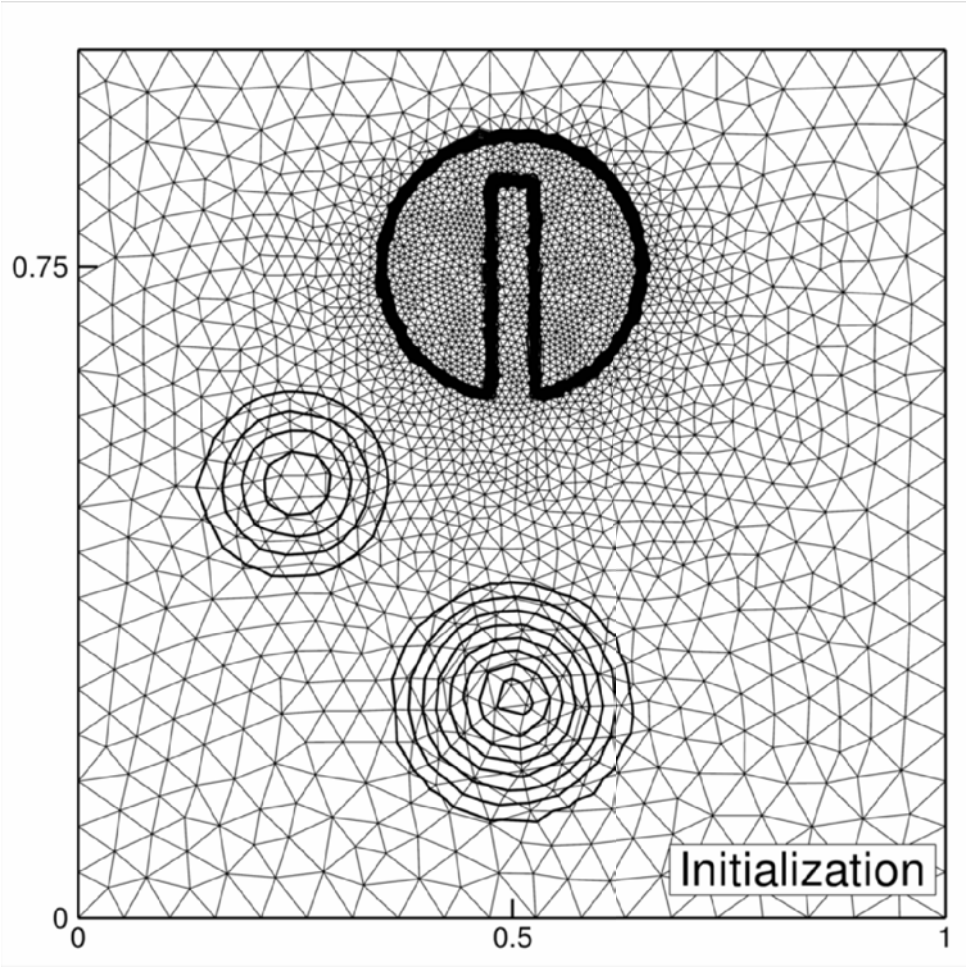
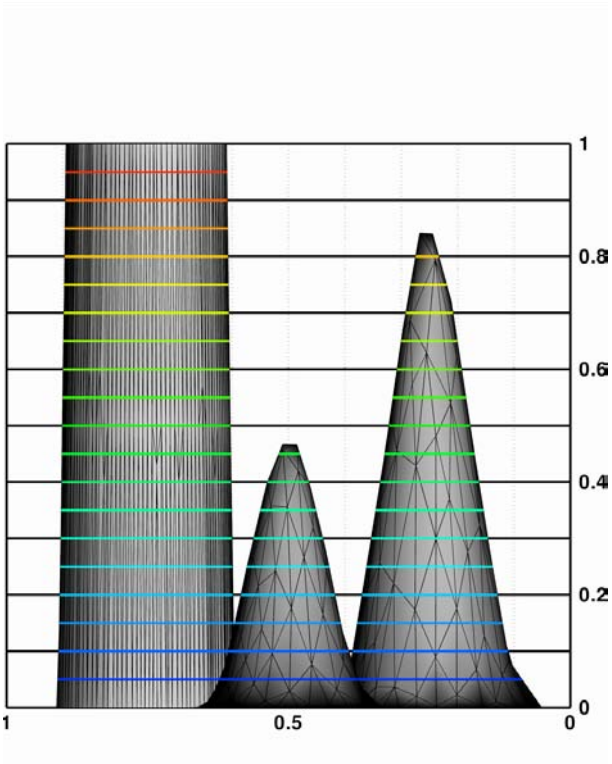


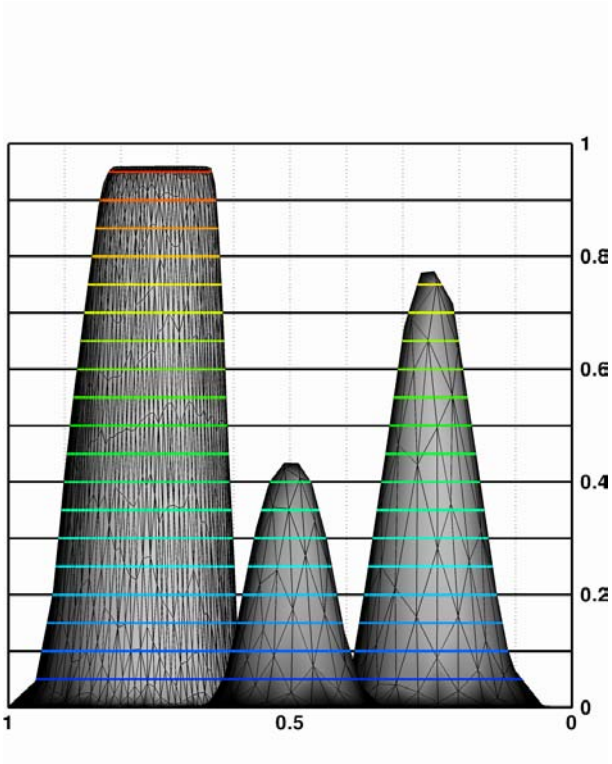
Figure 3.19: Initial mesh and initial data for the SBR problem. The mesh is composed of 5190 triangles refined around the slotted disk. The resulting mesh is genuinely non-uniform.

shows that the $u2$ solution on the two smooth profiles are exactly the ones obtained by the unlimited scheme. In other words, the $u2$ Detection Process maintains the same accuracy as an unlimited scheme on smooth profiles and almost the monotonicity of a limited scheme on non-smooth ones. The same conclusion applies for any other polynomial degrees tested hence we have skipped these figures. In Fig. 3.21 are displayed a zoom on the slotted disk at the final time for the initial/final, the limited MUSCL scheme (MLP [62]), MOOD-$\mathbb{P}_1$, MOOD-$\mathbb{P}_3$ and MOOD-$\mathbb{P}_5$ with $u2$ detection process.

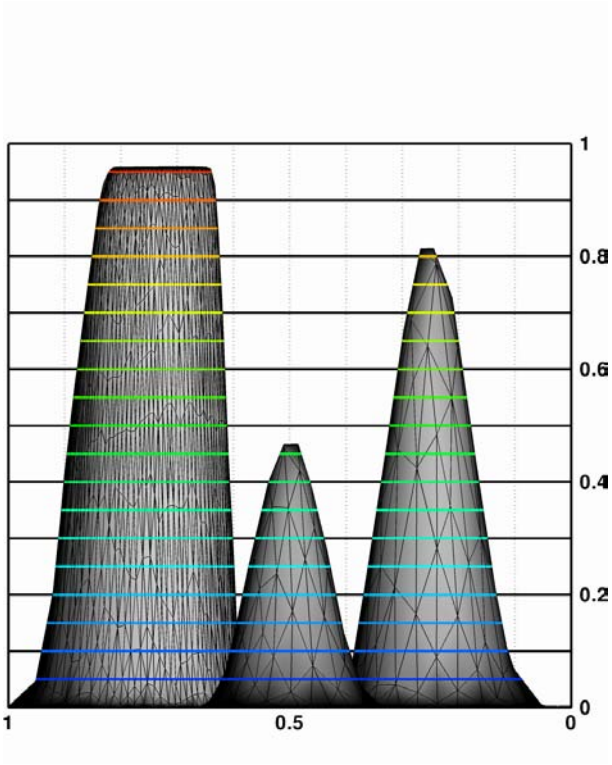
In Table 3.12 are gathered the errors for $\mathbb{P}_3$ and $\mathbb{P}_5$ in order to show that the $u2$ Detection Process provides a slightly better accuracy than the DMP detection process. Finally we display in Table 3.13 the min/max values of the final numerical solution for the limited MUSCL method (MLP), MOOD-$\mathbb{P}_1$, MOOD-$\mathbb{P}_3$ and MOOD-$\mathbb{P}_5$ all with DMP detection or $u2$ Detection Process. This table shows that the $u2$ detection process permits slight undershoots which is one of the reasons MOOD can reach high-order accuracy.



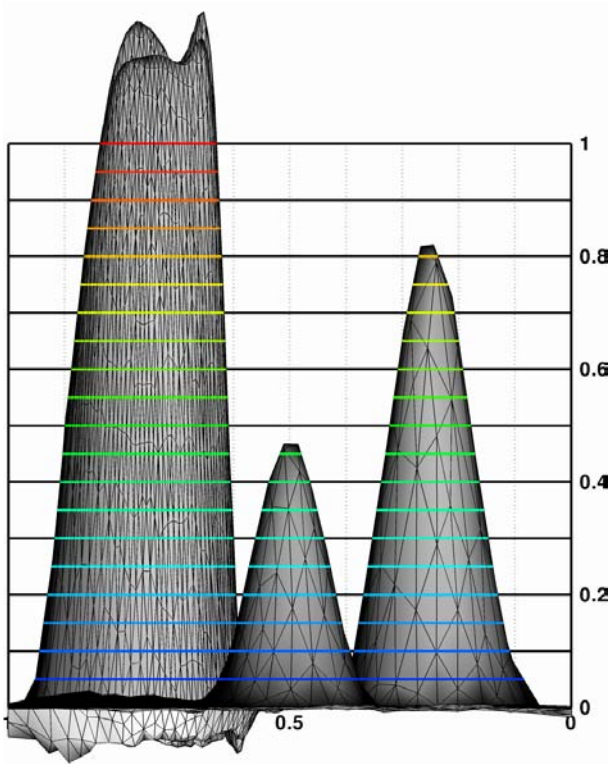
Initial≡Final



MOOD-$\mathbb{P}_5$ DMP



MOOD-$\mathbb{P}_5$ $u2$



UNLIMITED $\mathbb{P}_5$

Figure 3.20: Profiles of the SBR solution for the initial/final exact solution, MOOD-$\mathbb{P}_5$ with DMP detection process, MOOD-$\mathbb{P}_5$ with $u2$ detection process, MOOD-$\mathbb{P}_5$ without any limitation.

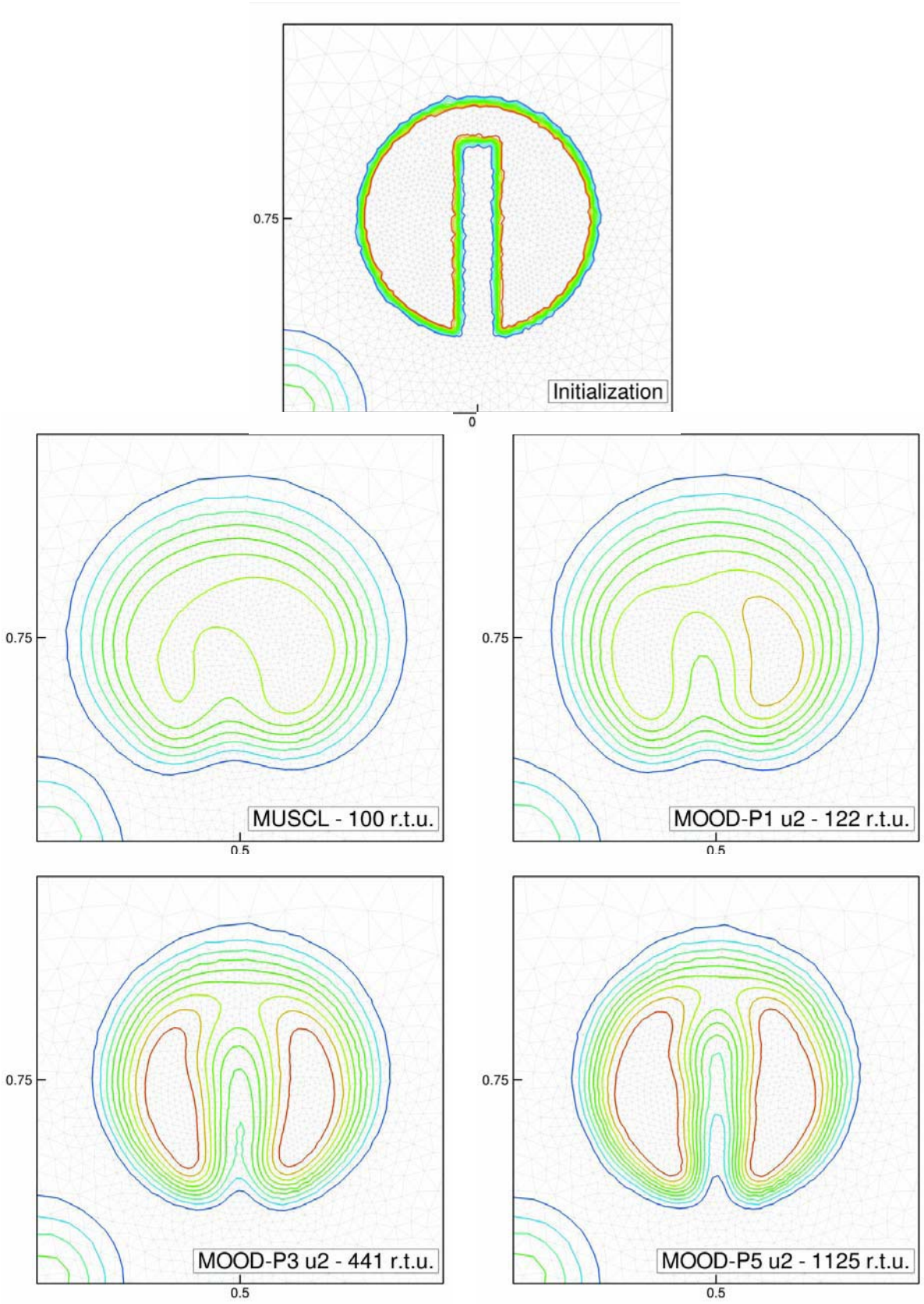


Figure 3.21: Profiles of the SBR solution for the initial/final exact solution (top), for a limited MUSCL method (MLP) and MOOD-$\mathbb{P}_1$, MOOD-$\mathbb{P}_3$, MOOD-$\mathbb{P}_5$ with $u2$ detection process.

L^1 Error	DMP	$u2$	UNLIMITED
$\mathbb{P}_3$	3.219E-1	3.171E-1	3.734E-1
$\mathbb{P}_5$	2.690E-1	2.621E-1	3.223E-1

Table 3.12: L^1 error for the SBR problem for different detection processes and polynomial degrees.

Method	MUSCL	MOOD-$\mathbb{P}_1$		MOOD-$\mathbb{P}_3$		MOOD-$\mathbb{P}_5$	
Detec.		DMP	$u2$	DMP	$u2$	DMP	$u2$
Min	5.58E-10	0.00E+00	-2.45E-03	3.27E-08	-1.31E-03	1.10E-08	-5.60E-05
Max	7.48E-01	8.53E-01	8.51E-01	9.49E-01	9.54E-01	9.61E-01	9.60E-01

Table 3.13: Minimal and maximal mean values for the SBR problem for different detection processes and polynomial degrees.

3.2.4.2 Euler system

In this section we test the MOOD method on unstructured meshes for hydrodynamics problems governed by the Euler system. First we need to assess the effective numerical accuracy of the method on a smooth problem for which an exact solution exists. We choose an isentropic vortex which presents a smooth profile during the entire simulation and, as such, permits the estimation of errors and convergence orders. In a second test we run the Lax shock tube to assess the essentially non-oscillatory behavior of MOOD compared to classical WENO results. Then we run the Double Mach reflection problem to highlight the good capacity of the MOOD method to capture strong shocks and contact discontinuities. Moreover we provide CPU cost and memory storage tables. Next the Noh problem is used to assess the positivity-preserving property of the MOOD method. Last we propose a genuine physical problem extracted from [72] for which experimental results are available.

⋆ Isentropic vortex

The isentropic vortex problem is detailed in [68] and [93], therefore we only mention the basic data for the sake of consistency. The simulation domain Ω is the square $[-5,5]-[-5,5]$ and we consider an initial gas flow given by the following condition (ambient gas) $\rho_\infty = 1.0$, $u_\infty = 1.0$, $v_\infty = 1.0$, $p_\infty = 1.0$, with a normalized ambient temperature $T^*_\infty = 1.0$ computed with the perfect gas equation of state and $\gamma = 1.4$.

A vortex centered at $\mathbf{x}_{\text{vortex}} = (x_{\text{vortex}}, y_{\text{vortex}}) = (0,0)$ is added to the ambient gas at the initial time $t = 0$ with the following conditions $u = u_\infty + \delta u$, $v = v_\infty + \delta v$, and $T^* = T^*_\infty + \delta T^*$

$$\delta u = -y' \tfrac{\beta}{2\pi} \exp\left(\tfrac{1-r^2}{2}\right), \quad \delta v = x' \tfrac{\beta}{2\pi} \exp\left(\tfrac{1-r^2}{2}\right),$$
$$\delta T^* = -\tfrac{(\gamma-1)\beta}{8\gamma\pi^2} \exp\left(1-r^2\right).$$

with $r = \sqrt{x'^2 + y'^2}$, $(x' = x - x_{\text{vortex}}, y' = y - y_{\text{vortex}})$ and vortex strength is given by $\beta = 5.0$.

Consequently, the initial density is given by

$$\rho = \rho_\infty \left(\frac{T^*}{T^*_\infty}\right)^{\frac{1}{\gamma-1}} = \left(1 - \frac{(\gamma-1)\beta}{8\gamma\pi^2}\exp\left(1-r^2\right)\right)^{\frac{1}{\gamma-1}} \tag{3.34}$$

We assume periodic condition on the boundary and the exact solution at any time t is the same vortex but translated.

The goal of the present test is to highlight the stagnation of the rate of accuracy when primitive variables are used for the polynomial reconstructions instead of conservative ones. As pointed out in the previous section, nonlinear operations on means values reduces the method order up to at most a second-order one. We have performed the numerical simulations of the isentropic vortex problem with the same mesh using the less restrictive Physical Admissible Detection (PAD) procedure to provide effective very high-order. A series of refined meshes (from 200 up to 51200 cells) are successively used to compute the numerical solution.

In Table 3.14 are gathered the L^1 and L^∞ errors and rates of convergence for MOOD-$\mathbb{P}_2$, MOOD-$\mathbb{P}_3$, MOOD-$\mathbb{P}_5$ using the Physical Admissible Detection Clearly, conservative variable reconstructions provide the optimal convergence rate whereas the reconstruction with primitive variables is systematically reduced to a second-order one. We also display in Fig. 3.22 the

		Conservative variables				**Primitive variables**			
Deg.	Cell Nb	L^1 error		L^∞ error		L^1 error		L^∞ error	
$\mathbb{P}_2$	200	1.850E-02	—	2.680E-01	—	2.002E-02	—	2.884E-01	—
	800	6.519E-03	1.50	1.255E-01	1.09	7.621E-03	1.39	1.771E-01	0.70
	3200	1.444E-03	2.17	2.208E-02	2.51	1.536E-03	2.31	4.054E-02	2.13
	12800	2.504E-04	2.53	3.631E-03	2.60	2.554E-04	2.59	6.060E-03	2.74
	51200	3.347E-05	2.90	4.923E-04	2.88	4.540E-05	2.49	8.756E-04	2.79
Expected order			3		3		3		3
$\mathbb{P}_3$	200	1.137E-02	—	1.880E-01	—	1.424E-02	—	2.384E-01	—
	800	2.504E-03	2.18	4.686E-02	2.00	3.530E-03	2.01	8.358E-02	1.51
	3200	3.524E-04	2.83	5.977E-03	2.97	5.666E-04	2.64	8.835E-03	3.24
	12800	1.947E-05	4.18	3.725E-04	4.00	1.377E-04	2.04	1.649E-03	2.42
	51200	1.069E-06	4.19	1.996E-05	4.22	3.460E-05	1.99	4.091E-04	2.01
Expected order			4		4		4		4
$\mathbb{P}_5$	200	8.193E-03	—	1.200E-01	—	1.161E-02	—	1.915E-01	—
	800	1.762E-03	2.22	3.433E-02	1.81	2.492E-03	2.22	3.740E-02	2.36
	3200	6.767E-05	4.70	1.133E-03	4.92	5.482E-04	2.18	6.112E-03	2.61
	12800	1.011E-06	6.06	2.237E-05	5.66	1.382E-04	1.99	1.598E-03	1.94
	51200	2.583E-08	5.29	4.809E-07	5.54	3.462E-05	2.00	4.039E-04	1.98
Expected order			6		6		6		6

Table 3.14: L^1 and L^∞ errors and convergence rates for the isentropic vortex problem with MOOD and the Physical Admissible Detection Process. Comparison between conservative and primitive variables polynomial reconstructions for different polynomial degrees.

convergence curves corresponding to the errors of Table 3.14. Finally we also mention that

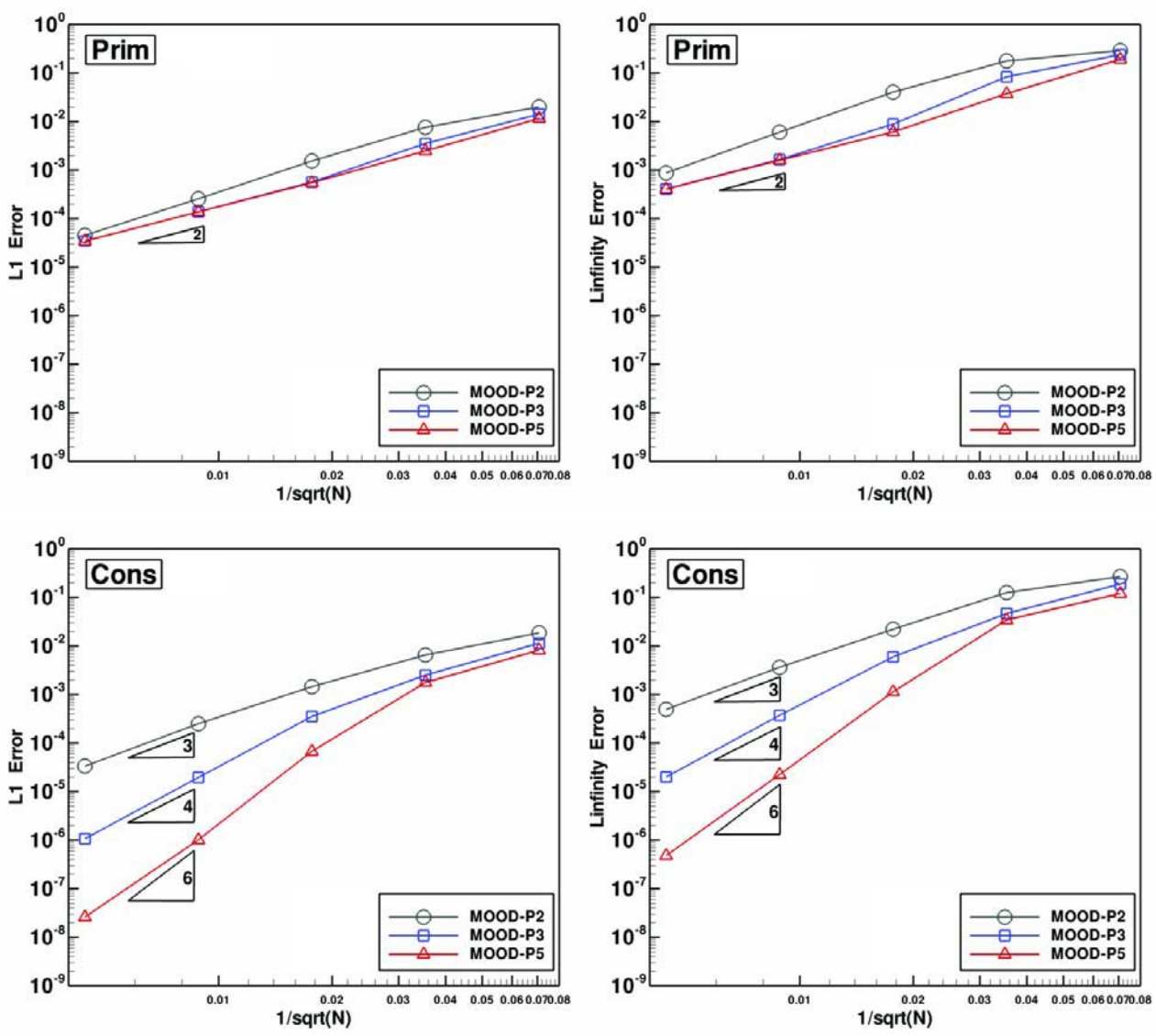


Figure 3.22: Convergence curves for the isentropic vortex. Top figures correspond to the reconstruction with primitive variables while bottom figures use reconstruction with conservative variables. The left column represents the L^1-norm error and the right column the L^∞-norm error. The PAD detection process has been used.

when the vortex is not in motion, *i.e.* $(u_\infty, v_\infty) = (0, 0)$, then the reconstruction using primitive variables does produce the correct order of convergence.

⋆ Lax shock tube

The 1D Lax shock tube consists of two states $(\rho_L, u_L, p_L) = (0.445, 0.698, 3.528)$ and $(\rho_R, u_R, p_R) = (0.5, 0, 0.571)$ separated by the interface $x = 0$. In order to compare with the finite volume multi-dimensional WENO results of [41], we run the problem on the domain $\Omega = [-1; 1] - [0, 0.2]$ until final time $t = 0.26$ using a mesh made of $100 - 10$ quadrangles split into two triangles

through the same diagonal for all cells (see Fig. 5.5 of [41]).
The goal of this test is to compare the essentially non-oscillatory behavior of the MOOD method using the $u2$ detection process with the classical genuinely multi-dimensional finite volume WENO results based on conservative or characteristic variables.
On the top panels of Fig. 3.23 we reproduce the density profiles from [41] obtained with the 4^{th}-order WENO method based on conservative (left) and characteristic (right) variables. We recall that this method uses combinations of $\mathbb{P}^2$ (3^{rd}-order) polynomials to reach 4^{th}-order. The bottom panel presents the MOOD-$\mathbb{P}_3$ density profiles on conservative variables with the $u2$+PAD detection process where only one line of triangles is displayed to comply with Fig. 5.6 of [41]. On the left we plot the result obtained with the Lax-Friedrichs numerical flux (which is used for the WENO results) and on the right the result using the HLLC flux.

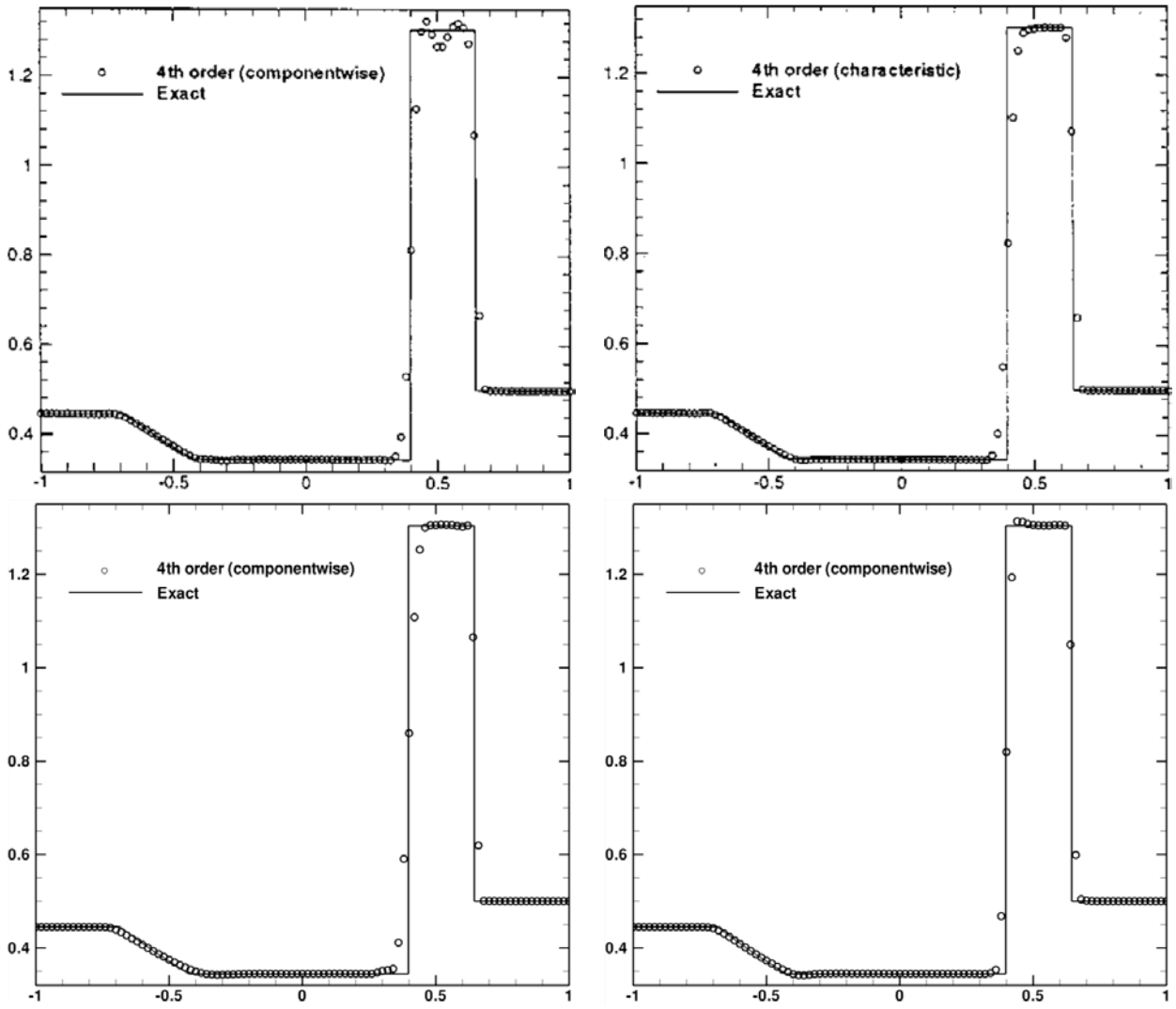


Figure 3.23: Comparison between WENO and MOOD methods on 100−10 quadrangles split into triangles — Top: results of the 4^{th}-order WENO method using Lax-Friedrichs flux from [41] on conservative variables (left) and on characteristic variables (right) — Bottom: results of the 4^{th}-order MOOD-$\mathbb{P}_3$ method with $u2$+PAD detection process using Lax-Friedrichs (left) and HLLC (right) fluxes.

We observe that the non-oscillatory behavior of the MOOD method with $u2$+PAD detection process is equivalent to the WENO on characteristic variables while it is clearly better than the WENO on conservative variables. Moreover we see that the use of HLLC gives better a result for a negligible additional cost, with only three points in the contact discontinuity instead of five and remains essentially non-oscillatory.

⋆ Double Mach reflection of a strong shock

The double mach reflection of a strong shock was first proposed in [91]. This test problem involves a Mach 10 shock in a perfect gas with $\gamma = 1.4$, which is initially positioned at $x = 1/6$, $y = 0$ and makes a 60° angle with the x-axis. The gas ahead of the shock is at rest and has uniform initial density $\rho^0 = 1.4$ and pressure $p^0 = 1$. The reflecting wall lies along the bottom of the domain, beginning at $x = 1/6$. The region from $x = 0$ to $x = 1/6$ along the bottom boundary at $y = 0$ is always assigned values for the initial post-shock flow. Inflow boundary condition on the left side and outflow condition on the right side are also set. At the top, the boundary conditions are set to describe the exact motion of the Mach 10 flow (see [24]).
The goal of the test is, on one hand, to quantitatively show the effect of the polynomial degree reconstruction when dealing with strong shock and, on the other hand, to observe the capacity of the method to reproduce the complex structure due to the contact discontinuities in the right part of the shock.
The mesh has been obtained using the free mesher Gmsh by a refinement of a coarser Delaunay ones, it is constituted of 102720 triangles (see Fig. 3.24 top.). Moreover for all figures 30 isolines between 1.39 and 23 have been drawn.

We depict in Fig. 3.24 the impact of the polynomial degree of the reconstruction on the numerical solution using the same mesh. The $u2$+PAD Detection Processes has been employed to control the oscillations in the vicinity of the shock. Clearly the degree of the reconstruction has a strong impact on the solution accuracy and improve the shock capture. Most relevant parts are the contact discontinuities in the right zone $x \in [2.3, 2.7]$ which show the capacity of the scheme to reduce numerical viscosity when employing higher-order reconstructions.

Figure 3.25 is a comparison between the Physical Admissible Detection (PAD) and the coupling $u2$+PAD. The $u2$ Detection Process reduces the oscillations but increases the numerical viscosity close to contact discontinuities. It is worth noting that even with a weak Detection Process, namely the PAD procedure, the MOOD method is still very robust and provides a solution resembling the classical ones from the literature [91]. The choice of the detecting procedure depends of the simulation goal: Less oscillations with the $u2$+PAD or less diffusive with the PAD alone.

To conclude with this test case, we provide in Tables 3.15 and 3.16 the cost of the MOOD method running on a single core of the three following machines (using -O3 flag for gfortran compiler)

M1 : A laptop with Intel Core2Duo P7550 (2 cores) @ 2.26GHz, 3MB of L2 Cache, 8GB of RAM.

M2 : A server with two Intel Xeon E5335 (4 cores) @ 2.00Ghz, 8MB of L2 Cache, 16GB of RAM.

M3 : A desktop with Intel Core i5 2500 (4 cores) @ 3.30GHz, 6MB of L2 Cache, 8GB of RAM.

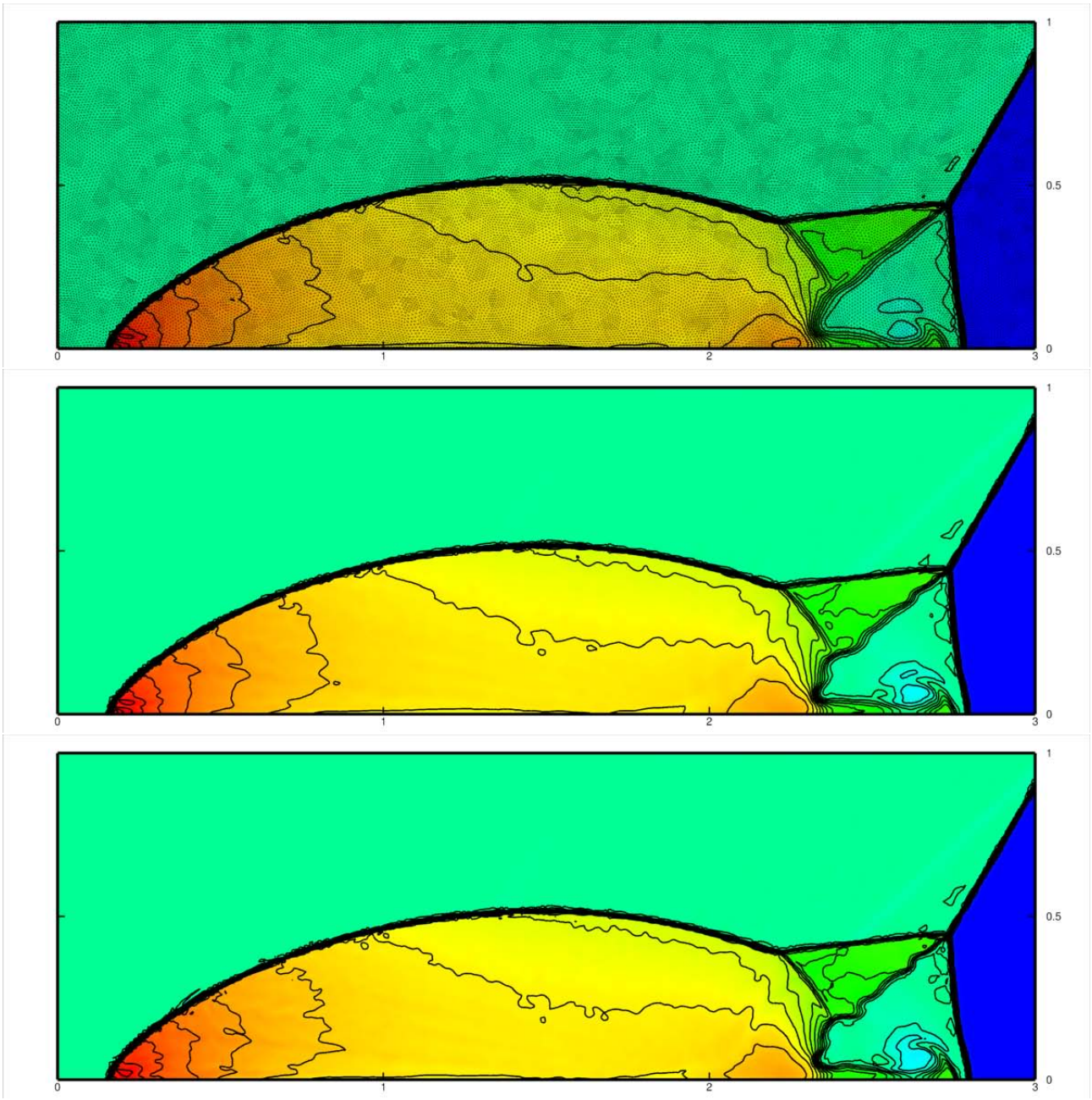


Figure 3.24: Comparison between the $\mathbb{P}_2$ (top), $\mathbb{P}_3$ (middle) and $\mathbb{P}_5$ (bottom) polynomial reconstructions with the conservative variables using the same mesh. Physical Admissible Detection (PAD) and $u2$ Detector have been both used to prevent numerical oscillations

This comparison is done on two different meshes, one made of 57600 uniform quadrilaterals and one Delaunay triangulation with 17624 cells. We compare MOOD-$\mathbb{P}2$, MOOD-$\mathbb{P}3$ and MOOD-$\mathbb{P}5$ for both the PAD and $u2$+PAD detection processes. We give in Table 3.15 the memory cost (in left column) and the total number of iterations (in right column) for all simulations, while we provide in Table 3.16 the total CPU time (in left column) and the time in micro-seconds

needed for one complete time step of a single cell (in right column) including reconstruction, flux computation and time integration (RK3) of all variables.
It is fairly difficult to compare the cost of two methods running on different machines, for instance the method is faster on triangles with $M2$ compared to $M1$ but it is the opposite for quadrilaterals. However according to reference [31] the MOOD method is competitive when compared to truly unstructured methods of the same order.

Mesh	$\mathbb{P}_2$ - 3^{rd}-order		$\mathbb{P}_3$ - 4^{th}-order		$\mathbb{P}_5$ - 6^{th}-order		Detection
	memory	iterations	memory	iterations	memory	iterations	
57600 qua.	240Mo	1012	385Mo	998	840Mo	1004	$u2$+PAD
	180Mo	1016	270Mo	1010	572Mo	1031	PAD
17624 tri.	60Mo	1264	105Mo	1265	250Mo	1265	$u2$+PAD
	50Mo	1268	67Mo	1272	165Mo	1275	PAD

Table 3.15: Memory storage and total number of iterations for the Double Mach problem according to the different configurations with the MOOD method.

Machine	Mesh	$\mathbb{P}_2$ - 3^{rd}-order		$\mathbb{P}_3$ - 4^{th}-order		$\mathbb{P}_5$ - 6^{th}-order		Detection
		total	per iter.	total	per iter.	total	per iter.	
M1	57600 qua.	2157s	37μs	3162s	55μs	8964s	155μs	$u2$+PAD
		1346s	23μs	2327s	40μs	8314s	140μs	PAD
	17624 tri.	601s	27μs	1003s	45μs	1650s	74μs	$u2$+PAD
		492s	22μs	762s	34μs	1573s	70μs	PAD
M2	57600 qua.	2228s	38μs	4785s	83μs	12371s	214μs	$u2$+PAD
		1629s	28μs	3830s	66μs	11629s	196μs	PAD
	17624 tri.	615s	27μs	922s	41μs	1292s	58μs	$u2$+PAD
		521s	23μs	707s	32μs	1079s	48μs	PAD
M3	57600 qua.	683s	12μs	1089s	19μs	3696s	66μs	$u2$+PAD
		490s	8μs	859s	15μs	3604s	61μs	PAD
	17624 tri.	265s	12μs	397s	18μs	594s	27μs	$u2$+PAD
		230s	10μs	308s	14μs	492s	22μs	PAD

Table 3.16: Total time and cost for one complete time step of a single cell for the Double Mach problem.

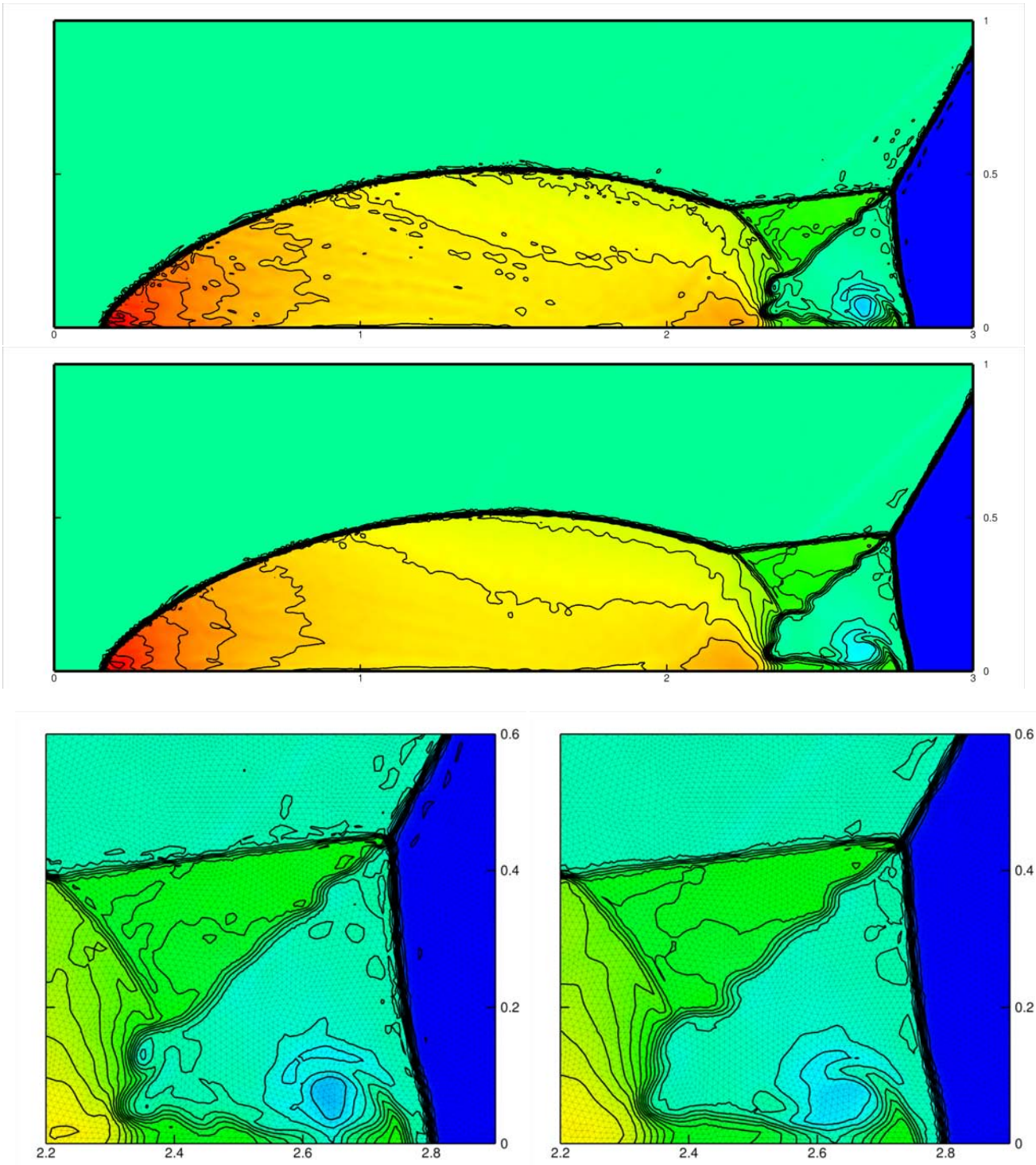


Figure 3.25: Results of the MOOD method with $\mathbb{P}5$. In the top figure, simulation has been carried out with the Physical Admissible Detection (PAD) Detection Process while we have both employed the PAD and $u2$ Detection in the middle figure. The left bottom and right bottom figure give a zoom of the solution with the PAD and $u2$+PAD Detection Process respectively.

⋆ **Noh problem as a positivity-preserving test case.**

The goal of the Noh problem in Cartesian geometry is to numerically prove that the MOOD method is positivity-preserving, see section 3.2.3.2 for a discussion on this point. It is a difficult problem well-known in the Lagrangian community, see as instance [56, 55]. It is noticeable that our implementation of the classical MUSCL scheme is not able to simulate this problem without creating negative pressures.
The problem is run in the disk of radius 1.2 centered at $(0,0)$. We initialize a perfect gas with $\gamma = 5/3$, density $\rho_0 = 1$, pressure $p_0 = 10^{-10}$ and velocity $U_0(x,y) = \big(-x,-y\big]/\sqrt{x^2+y^2}$ such that $||U_0(x,y)|| = 1$. A cylindrical shock wave generated at the origin further diverges until final time $t_{\text{final}} = 2.0$. The exact solution at t_{final} is thus given by

$$\{\rho, p, u_r\} = \begin{cases} \{16, \frac{16}{3}, 0\} & \text{if } r < r_s, \\ \{(1+\frac{2}{r}), 10^{-10}, -1\} & \text{if } r > r_s, \end{cases} \tag{3.35}$$

where r is the radius, u_r the radial velocity and $r_s = 2/3$ the shock wave position. This problem is simulated on a polygonal mesh made of 19756 cells with about 100 cells in the radial direction. Notice that the mesh is made of seven layers of quadrangles separated with degenerated polygons, see Fig. 3.26. We display the MOOD-$\mathbb{P}3$ results for the density maps (left panels) and the density as a function of cell radius (right panels) in Fig. 3.26. The top panels correspond to the PAD detection process whereas the bottom ones correspond to the $u2$+PAD process. One obverves that the symmetry is almost perfectly reproduces. Notice that the PAD detection process is only intended to ensure the physical meaningfulness of the solution but does not prevent oscillations to occur. Independently of the order of the scheme the PAD always provides a meaningful solution. As a consequence the $u2$+PAD not only provides a valid solution without negative pressure but also removes the oscillation after the shock wave.

⋆ **Impact of a shock on a cylindrical cavity**

We finally test the ability of MOOD method to capture physics in realistic conditions by simulating the experiment proposed in [72] where a planar shock impacts a cylindrical cavity. We consider the case of a nominal incident shock Mach number of 1.33 in ambient air (with $\gamma = 1.4$) at 0.95 bar pressure. Moreover we use the domain configuration A (following notation of [72]) we detail in Figure 3.27.
The variables initialization is split in two parts, the pre-shock values

$$(\rho, u, v, p) = (1.1175, 0.0, 0.0, 95000.0),$$

and the post-shock ones

$$(\rho, u, v, p) = (1.7522, 166.3435, 0.0, 180219.75),$$

leading to conditions of [72] at temperature $T = 296.15K$.
The simulation is only preformed in the lower half part of the domain for symmetry argument, namely from $y = 0\ mm$ to $y = 75\ mm$. The 193615 cells mesh is composed of triangles, quadrangles but also more general polygons with non-conformal elements (see Figure 3.28) to better suit with the complex geometry of the set-up. Notice that non-conformity is simply handled

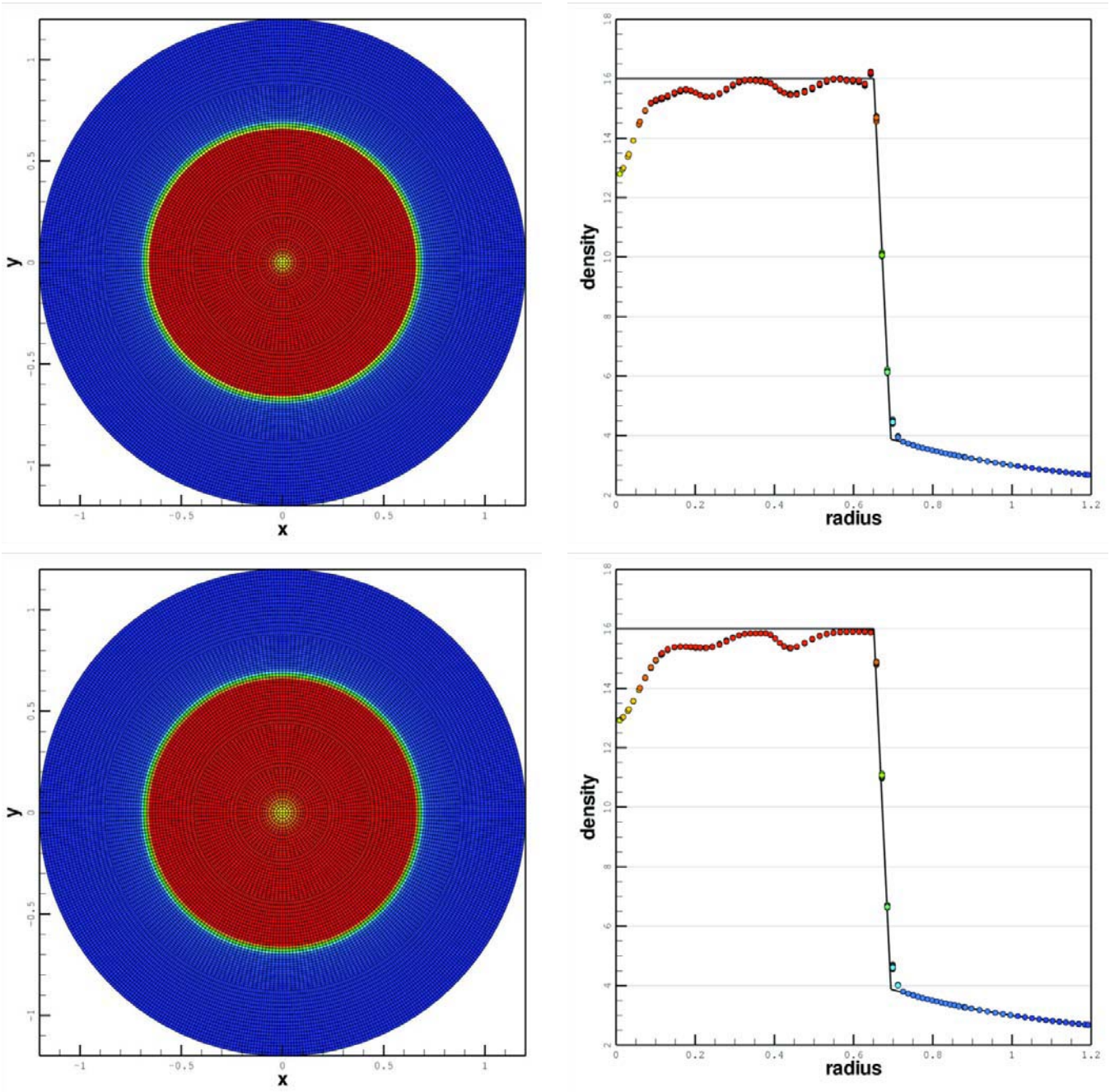


Figure 3.26: Noh problem at $t_{\text{final}} = 2.0$ on a polygonal grid — Left: Density map and mesh — Right: Cell density as a function of cell radius *vs* exact solution — Top panels correspond to the PAD detection process — Bottom panels correspond to the $u2$+PAD detection process.

using polygons, *i.e.* no special treatment is used. We also deliberately use a heterogeneous mesh to highlight that the MOOD method is not much affected by the quality of the mesh.

The simulation are carried out with the MOOD-P3 method (fourth-order) using the PAD and the $u2$ Detection Process. Pictures are rendered as a full mesh by symmetry even if the com-

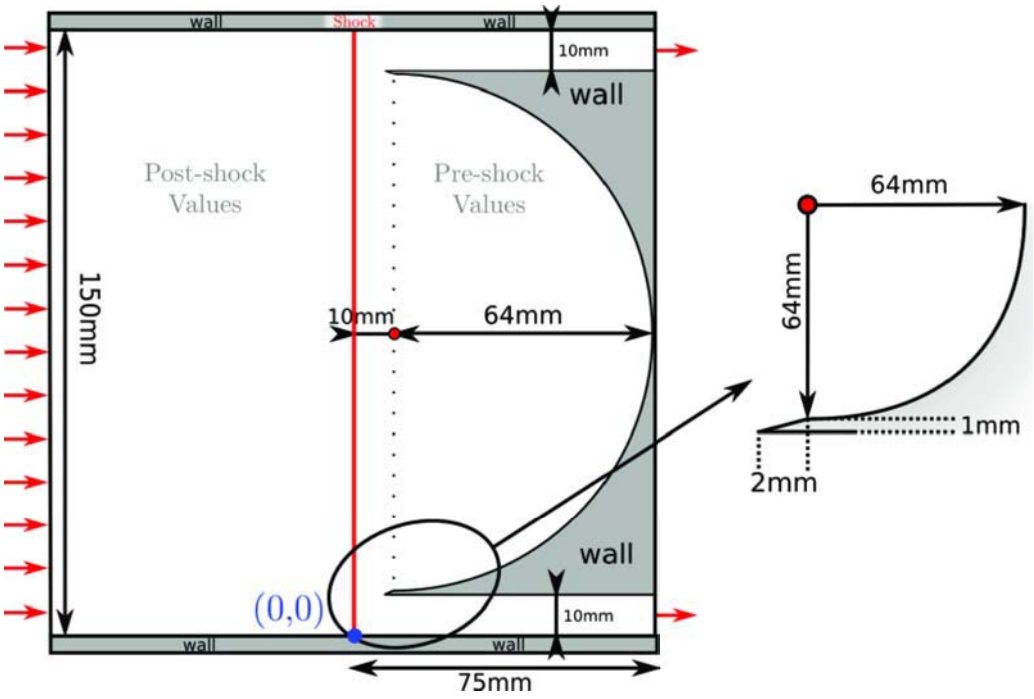


Figure 3.27: Domain characteristics for the shock impacting a cylindrical cavity. Red arrows represent inflow and outflow boundary conditions.

putation was done on a half-domain to easier compare with physical results of [72]. Figure 3.29 represents the density gradient magnitude at six different times to embrace the global behavior of the solution. In details, top right is chosen to be compared to Fig. 7 (a) of [72], bottom center to Fig. 8 (d) and bottom right to Fig. 9 (c) of same paper. Our results are clearly in agreement with physical results. In Figure 3.30 different zooms on solution at several times are plotted. On the top part, density gradient magnitude at a late time is given and is to be compared to Fig. 14 (b) in [72] while we superpose, in the bottom figure, the velocity vectors on the density magnitude gradient to show the created vortices at the entrance of the cavity (left) and highlight the instabilities lying along the wall.

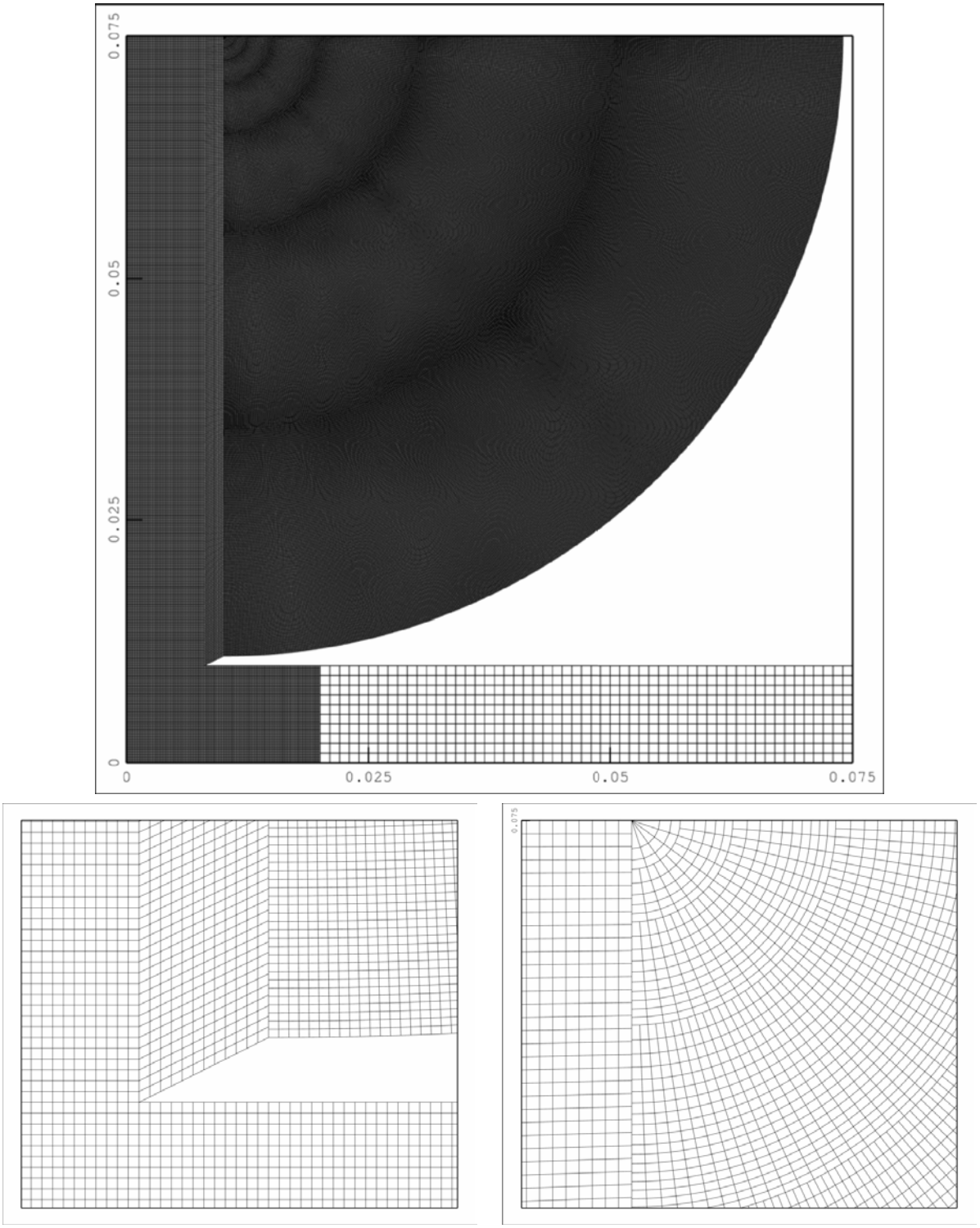


Figure 3.28: On top, we display the global view of the mesh where the different mesh zones are clearly visible. On bottom, zooms on the non-convex part of the mesh (on left) and on the junction between the polar part of the mesh and the quasi-uniform one (right). Non-conformity are clearly visible.

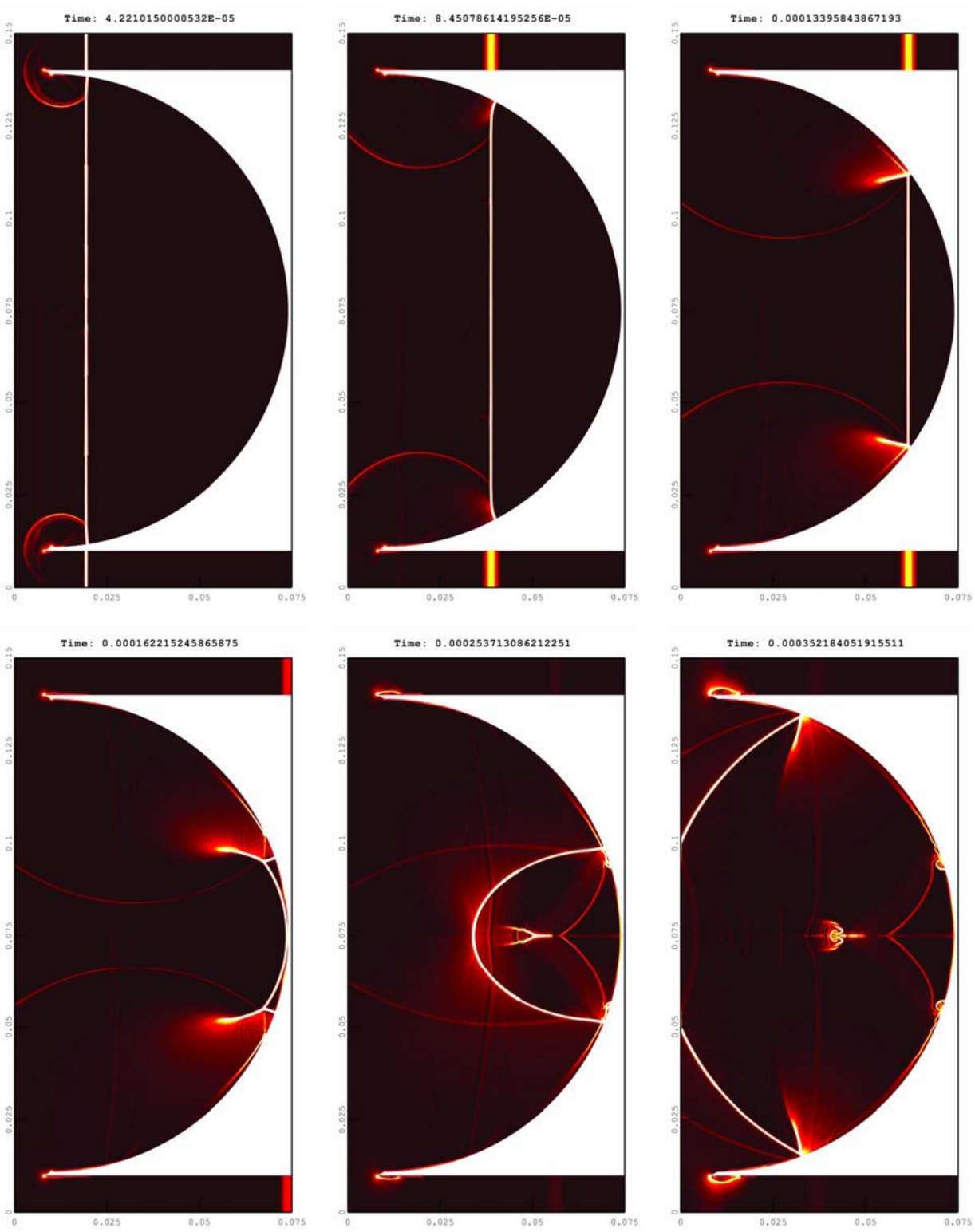


Figure 3.29: Gradient density magnitude is shown at different times. Time 0 corresponds to the initial shock at position x=0.

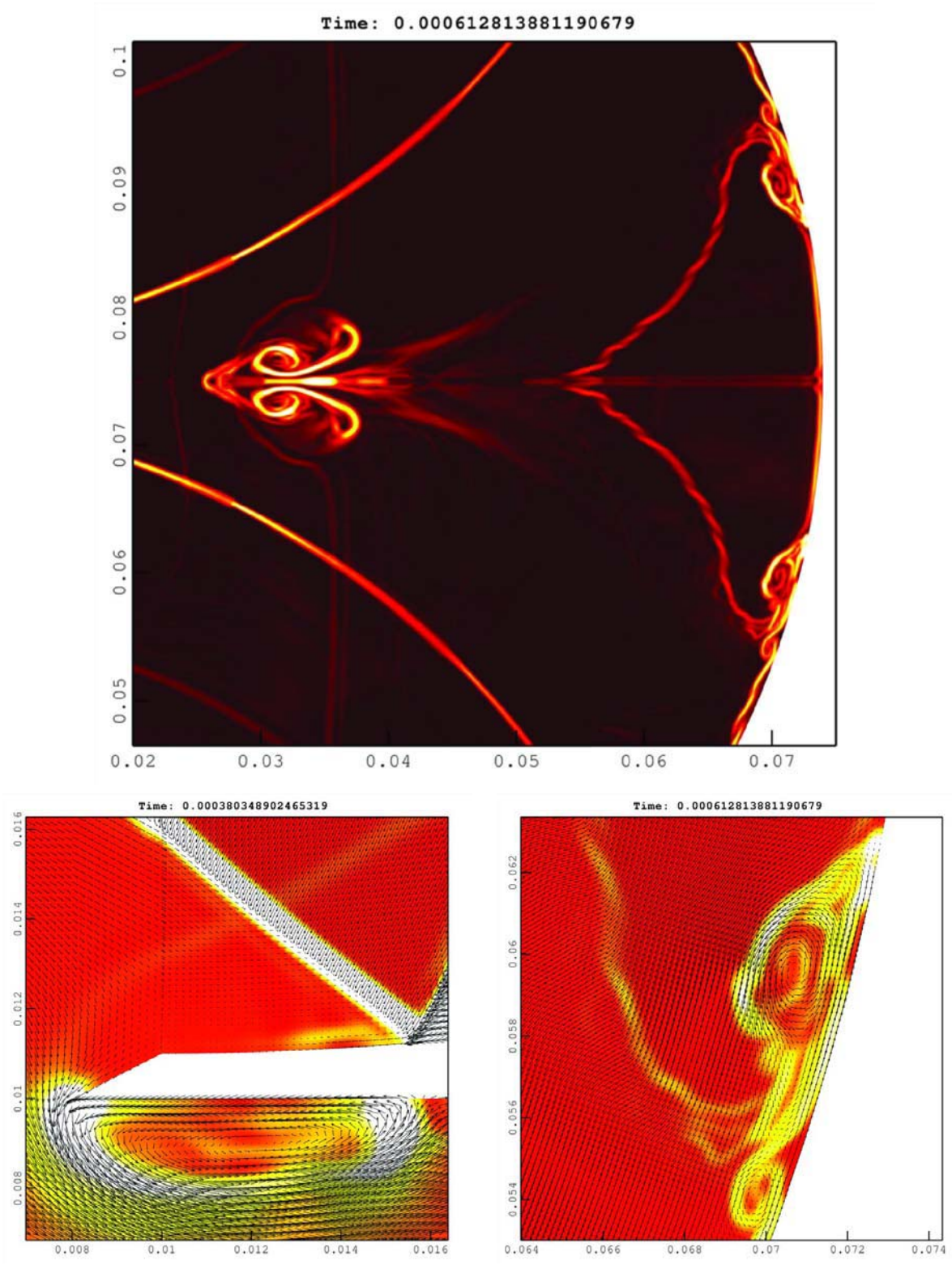


Figure 3.30: Zooms on different parts of the solution. On top, gradient density magnitude is shown at a late time when instabilities are well developed. On bottom, vortices at the entry of the cavity (left) and the instabilities (right) along the wall are displayed with density gradient magnitude in color and velocity vectors.

3.2.5 Conclusion and perspectives

The paper presents important new extensions of the MOOD method for unsteady advection and hydrodynamics equations, that ensure high-order approximations (up to the sixth-order) on unstructured meshes.
We introduced new efficient detection processes and proved that the MOOD method is intrinsically positivity-preserving for the hydrodynamics system of equations assuming that the first-order scheme is. This has been numerically assessed on the Noh problem for which our implementation of the MUSCL scheme fails due to negative pressures.
Then both for the advection equation and the hydrodynamics Euler system, we proposed numerical tests to confirm that the MOOD method provides very high-order of accuracy on unstructured meshes for smooth solutions (*e.g.* isentropic vortex in motion) and non-oscillatory behavior on discontinuous ones (*e.g.* Lax shock tube). Moreover the memory storage and CPU time have also been reported for the double Mach problem, proving that the MOOD method is competitive. The last numerical test showed that the MOOD method, on a relatively coarse and non-conformal polygonal mesh, is able to simulate complex physics from an experimental set-up of the impact of a shock wave on a cylindrical cavity.
Finally we plan to improve the detection procedure, especially for vectorial problems to achieve a very low diffusion but still preventing the oscillations from appearing. Application to full three-dimensional problem is also an attractive task since performing an efficient computational solution is always a challenging problem. The extension of the MOOD method to deal with steady-state solution needs also more investigations. Overall an important perspective is the polynomial reconstruction itself. We have observed that the main computational cost comes from the reconstruction stage and that the reconstruction quality strongly depends on the stencil employed. Such a point is of crucial importance from a computational point of view to obtain tractable complex numerical simulations.

Appendix: The Discrete Maximum Principle on mean values provides at most a second-order scheme.

We recall that a time explicit scheme preserves the Discrete Maximum Principle (DMP) if for all cell K_i

$$\min_{j\in\overline{\nu}(i)}(U_i^n, U_j^n) \leq U_i^{n+1} \leq \max_{j\in\overline{\nu}(i)}(U_i^n, U_j^n). \tag{3.36}$$

It has been shown in [65, 59, 46] that any scheme based on the DMP property reduces the accuracy to second-order for regular functions due to inaccurate approximation at extrema. Indeed following [96], let us consider the advection problem in $\mathbb{R}$ to avoid boundary condition issues

$$\left\{\begin{array}{rcl} \partial_t U + \partial_x U &=& 0, \\ U(x, t=0) &=& \cos(x). \end{array}\right. \tag{3.37}$$

We consider a uniform discretization $x_i = ih$, $i \in \mathbb{Z}$ and $h > 0$ being the cell size and initialize the mean value on cell $K_0 = [0, h]$ as

$$U_0^{t=0} = \frac{1}{h}\int_0^h \cos(x)\, dx = \frac{\sin(h)}{h}. \tag{3.38}$$

Now, let us perform one time step with $\Delta t = h/2$ of a finite volume scheme which respects the DMP property. The exact solution at time $t = h/2$ is $U^{ex}(x, h/2) = \cos(x - h/2)$ and accordingly the exact mean value on K_0 is

$$U_0^{ex,t} = \frac{1}{h}\int_0^h \cos(x - h/2)\, dx = \frac{2\sin(h/2)}{h}. \tag{3.39}$$

However a Taylor expansion provides

$$U_0^{ex,t} = \frac{2\sin(h/2)}{h} = 1 - \frac{h^2}{24} + O(h^4).$$

But the initial mean values are bounded by

$$U_0^{t=0} = \frac{\sin(h)}{h} = 1 - \frac{h^2}{6} + O(h^4).$$

Clearly, the exact mean value $U_0^{ex,t}$ on cell K_0 is greater than the maximum mean values over all cells at time $t = 0$ with an error of $h^2/8$ as

$$|U_0^{ex,t} - U_0^{t=0}| \leq |\frac{h^2}{24} - \frac{h^2}{6} + O(h^4)| = \frac{h^2}{8} + O(h^4).$$

Therefore a scheme which fulfills the DMP property necessarily provides a solution lower than $\sin(h)/h$, hence after the first cycle the numerical solution verifies $U_0^t \leq U_0^{t=0} = 1 - \frac{h^2}{6} + O(h^4)$. It follows that the approximation of the mean value has an error of order $O(h^2)$ compared to the exact mean value on cell K_0. Consequently the scheme is at most second-order accurate and DMP-type of criteria cannot be used strictly for higher than second-order schemes and has to be relaxed.

3.3 Part III: 6$^{\text{th}}$-order accuracy on 3D mixed-element meshes

This section is dedicated to the third article introducing the 3D version of the MOOD method. The reference is:

S. Diot, R. Loubère, S. Clain, *The MOOD method in the three-dimensional case: Very-High-Order Finite Volume Method for Hyperbolic Systems*, submitted to Int. J. Numer. Meth. Fl. (2012).

In next paragraph, we sum up the content of the publication and highlight with hindsight the pros and cons of the MOOD method at that time. We then reproduce the paper from the abstract to the conclusion only correcting the misprints and modifying the references to fit the global bibliography.

Summary & Review

The following article has been submitted few weeks before the defense and contains two important extensions to the MOOD method. Firstly the method is applied to mixed-element 3D meshes with polynomial up to degree five. This is an important step forward toward the effective use of the MOOD method for realistic computations.

Secondly the correction of the $u2$ detection process described in section 2.2 is presented in order to handle the convergence problems for the Euler system mentioned in previous section. This correction cancels the cell-dependence of the only parameter used in this criteria. The optimal convergence using this correction is assessed on the isentropic vortex in motion ran on a series of uniform hexahedral and pyramidal 3D meshes.

Furthermore as for the 2D case, we propose a precise computational cost comparison for the 3D explosion problem carried out on a regular mesh of pyramids. The same three computers as for the second publication are used and the results demonstrate the high efficiency of the method. In particular, the memory storage is very low compared to (W)ENO methods since basically only one reconstruction matrix per cell and per degree is stored. We thus remark that a one million cells simulation with the 4^{th}-order MOOD-$\mathbb{P}3$ method only uses about 16 GB of memory.

At last, we propose some new directions for the parallelization of the MOOD method in order to complete its design and to run huge realistic simulations.

Finally these results corroborate the claim that the MOOD method is simple in the sense that we have been able to develop in three years an effective 3D very high-order Finite Volume method which is more efficient and promising than the existing ones. We thus believe that this publication will be a major improvement.

Abstract

The Multi-dimensional Optimal Order Detection (MOOD) method for two-dimensional geometries has been introduced in "A high-order finite volume method for hyperbolic systems: Multi-dimensional Optimal Order Detection (MOOD)", J. Comput. Phys. 230 (2011), and enhanced in "Improved Detection Criteria for the Multi-dimensional Optimal Order Detection (MOOD) on unstructured meshes with very high-order polynomials", Comput. & Fluids 64 (2012). We present in this paper the extension to 3D mixed meshes composed of tetrahedra, hexahedra, pyramids and prisms. In addition, we simplify the $u2$ detection process previously developed and show on a relevant set of numerical tests for both the convection equation and the Euler system that the optimal high-order of accuracy is reached on smooth solutions while spurious oscillations near singularities are prevented. At last, the intrinsic positivity-preserving property of the MOOD method is confirmed in 3D and we provide simple optimizations to reduce the computational cost such that the MOOD method is very competitive compared to existing high-order Finite Volume methods.

3.3.1 Introduction

First introduced in [18], the Multi-dimensional Optimal Order Detection (MOOD) method proposes a new strategy to provide third-order approximations to hyperbolic scalar or vectorial problems for two-dimensional geometry with structured meshes. The author then gave an extension in [27] to general unstructured 2D meshes where they achieved a sixth-order convergence in space introducing new detection-limitation procedure. The issue we address in the present paper is to extend the MOOD method to three-dimensional geometries with general polyhedral unstructured meshes for the scalar advection equation and the hydrodynamics Euler system. The method casts in the generic framework of the finite volume method but fundamentally differs from the traditional techniques by the specific detection-limitation procedure implemented by the authors. Indeed, classical high-order polynomial reconstruction-based schemes such as MUSCL [48, 83, 84, 49, 42, 13, 50] or ENO/WENO methods [39, 40, 70, 95, 69] rely on an *a priori* limiting procedure to achieve some stability properties. For instance, in MUSCL-like methods unlimited slopes are reduced through the use of limiters to respect some Discrete Maximum Principle or Total Variation Diminishing properties. In the same way, ENO/WENO-like methods employ an essentially non-oscillatory polynomial which provides an accurate solution while preventing undesirable oscillations from appearing.

We state that such limitation strategies are *a priori* in the sense that only the data at time t^n are used to first perform the limitation procedure and then compute an approximation at time t^{n+1}. Generally, the "worst case scenario" (speculative approach) has to be considered as plausible and, consequently a "precautionary principle" is applied. It results that most of the time the limitation mechanism unnecessarily operates and may reduce the scheme accuracy due to restrictive assessments. The MOOD principle lies in a different approach since we first compute a candidate solution for time t^{n+1} and use this *a posteriori* information to check if the proposed approximation is valid. Roughly speaking, we compute a candidate solution without any limitation using local polynomial reconstructions to provide accurate approximation of the flux (the degree is set to a prescribed maximal value). We then detect if this solution locally fails to fulfill some stability criteria (detection of problematic cells) and further decrement polynomial degree only on problematic regions (limitation step) before recomputing a new candidate solution. An iterative procedure (the MOOD algorithm) is carried out by successively decrementing the degree to provide the *optimal* local polynomial reconstruction for each cell to satisfy the given stability criteria. At the end of the MOOD algorithm, the candidate solution is eligible and turns out to be the approximation at time t^{n+1}. The *a posteriori* strategy brings new benefits. We dramatically reduce the number of polynomial reconstructions regarded to the ENO/WENO method since our technique only requires one polynomial function for each cell. Most of the time, the polynomial with maximal degree is employed since the limitation mechanism is only activated for problematic cells (objective approach). From a physical point of view, the positivity preserving property (for the Euler equations as instance) is simply guaranteed by the *a posteriori* strategy applying a simple detection procedure which checks the physical admissibility of the solution.

The paper is organized as follows. In section 3.3.2, we detail the concept of the MOOD method, while the detection criteria are developed in section 3.3.3 both for the advection equation and

the hydrodynamics Euler system of equations. Numerical tests are proposed in section 3.3.4 to prove the efficiency of the method: we first consider the scalar advection equation and show effective high-order of accuracy for regular solutions with the fourth- and sixth-order schemes considering meshes made of hexahedra and pyramids. We then propose an H-shaped discontinuous profile in rotation to verify the non-oscillatory property of the MOOD method. Finally, numerical simulations are carried out for the Euler system to test the method with a nonlinear vectorial problem. As preliminary experiments, the classical 1D test cases, namely the Sod and Lax shock tubes and the Shu-Osher and Woodward-Collela problems, are run on 3D tetrahedral and pyramidal meshes. Then the numerical order of accuracy is checked on the 2D isentropic vortex in motion extended by invariance for a 3D mesh and a realistic 2D test case (introduced in [27]), namely the *impact of a shock wave on a cylindrical cavity*, is carried out on a mesh made of triangular and quadrangular prisms. At last, we present results for the 3D explosion problem on a pyramidal mesh and the interaction of a shock wave with a quarter of cone on a mesh of 1.1 millions of tetrahedra. We moreover provide computational cost (CPU and memory storage) for the 3D explosion problem for the MOOD method for different polynomial degrees. We conclude with section 3.3.5 and delineate some future perspectives.

3.3.2 The MOOD concept

We consider the generic hyperbolic equation defined on a domain $\Omega \subset \mathbb{R}^3$, $t > 0$ cast in the conservative form

$$\begin{aligned} \partial_t U + \nabla \cdot F(U) &= 0, && \text{(3.40a)} \\ U(\cdot,0) &= U_0, && \text{(3.40b)} \end{aligned}$$

where $U = U(\mathbf{x}, t)$ is the vector of unknown functions depending on $\mathbf{x} = (x, y, z) \in \Omega$ and on the time t. We denote by F the so-called physical flux where we shall consider the autonomous case $F = F(U)$ (Euler system as instance) and the non-autonomous situation $F = F(\mathbf{x}, U)$ such as $\nabla_{\mathbf{x}}.F(\mathbf{x}, .) = 0$ (scalar convection case). Function u_0 stands for the initial condition while the boundary conditions will be prescribed in section devoted to the numerical simulations.

3.3.2.1 Framework

In order to design the numerical scheme, we introduce the following notation illustrated in Figure 3.31. The computational domain Ω is assumed to be a polyhedron bounded set of $\mathbb{R}^3$ divided into polyhedral cells K_i, $i \in \mathcal{E}_{el}$ where $\mathcal{E}_{el}$ is the cell index set. For each cell K_i, we denote by $\mathbf{c}_i$ the cell centroid, and define the set $\underline{\nu}(i)$ of all the indexes $j \in \mathcal{E}_{el}$ such that elements K_j share a common face f_{ij} with K_i and the set $\overline{\nu}(i)$ of all the indexes $j \in \mathcal{E}_{el}$ such that $K_i \cap K_j \neq \emptyset$ (see illustrations in Figure 3.32). Moreover for each face $f_{ij} = K_i \cap K_j$, $\mathbf{n}_{ij}$ stands for the unit normal vector going from K_i to K_j and we denote by $\left(\xi_{ij,r}, q_{ij,r}\right]$, $r = 1, ..., R_{ij}$ the quadrature rule for the numerical integration on f_{ij} where $\xi_{ij,r}$ is the weight associated to the r^{th} quadrature point $q_{ij,r}$ with $\sum_{r=1}^{R} \xi_{ij,r} = 1$, $\forall i \in \mathcal{E}_{el}$ and $\forall j \in \underline{\nu}(i)$ (see Figure 3.31).

To avoid a specific treatment of the boundary faces we introduce the notion of virtual cell. To this end, assuming that cell K_i has a face $f_{ie} = K_i \cap \partial\Omega$ on $\partial\Omega$, we introduce the virtual cell K_j where $j \notin \mathcal{E}_{el}$ obtained by symmetrical transformation of the original cell K_i which

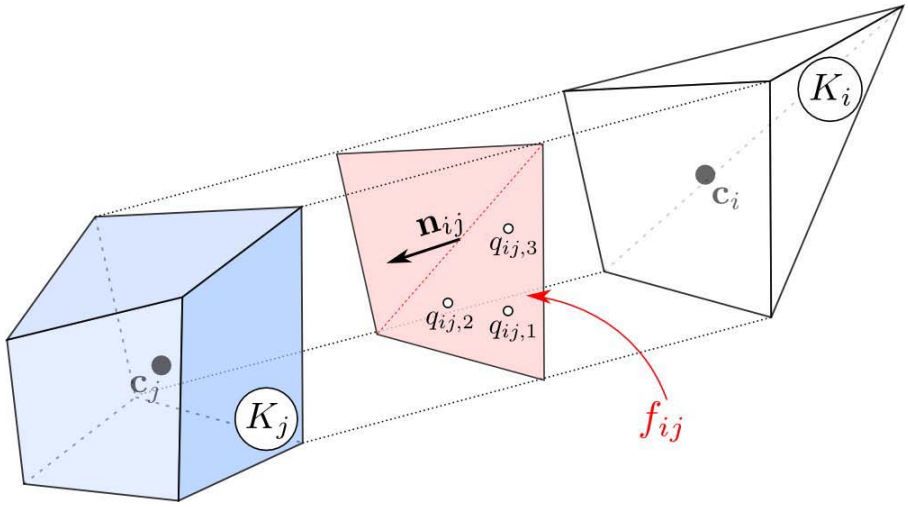


Figure 3.31: Notation for a three-dimensional mesh: exploded view of the face f_{ij} between two cells K_i and K_j. Centroids are respectively denoted by $\mathbf{c}_i$ and $\mathbf{c}_j$. Three quadrature points $q_{ij,r}, r = 1, 2, 3$, on f_{ij} are drawn for illustration. The unit normal vector pointing from K_i to K_j is denoted $\mathbf{n}_{ij}$.

represents the exterior side of Ω. We shall denote by $\mathcal{E}_{bd}$ the index set of all virtual cells such that $\widetilde{\mathcal{E}_{el}} = \mathcal{E}_{el} \cup \mathcal{E}_{bd}$ is the index set of all cells (including the virtual ones).

The generic first-order Finite Volume scheme associated to equation (3.40) writes

$$U_i^{n+1} = U_i^n - \Delta t \sum_{j \in \underline{\nu}(i)} \frac{|f_{ij}|}{|K_i|} \mathbb{F}(U_i^n, U_j^n, \mathbf{n}_{ij}), \tag{3.41}$$

where U_i^n is an approximation of the mean value of U at time t^n on K_i, $\mathbb{F}(U_i^n, U_j^n, \mathbf{n}_{ij})$ is a numerical flux which satisfies the properties of consistency and monotonicity for the scalar case, Δt stands for the time step while $|f_{ij}|$ and $|K_i|$ are the area of face f_{ij} and the volume of cell K_i respectively.

To provide high-order finite volume schemes, we use convex combinations of the initial building-block (3.41) with better approximations at the quadrature points to compute the numerical flux (see [18, 27] for instance). The high-order schemes are thus obtained from an original first-order Finite Volume and that is of crucial importance from a computational and implementation point of view (re-use of the original first-order code to achieve high-order approximations). We substitute the first-order approximations of the flux integral by higher-order versions, the scheme then writes

$$U_i^{n+1} = U_i^n - \Delta t \sum_{j \in \underline{\nu}(i)} \frac{|f_{ij}|}{|K_i|} \sum_{r=1}^{R_{ij}} \xi_{ij,r} \mathbb{F}(U_{ij,r}^n, U_{ji,r}^n, \mathbf{n}_{ij}), \tag{3.42}$$

where $U_{ij,r}^n, U_{ji,r}^n$ are high-order approximations of U at quadrature points $q_{ij,r}$ on both side of f_{ij}.

For meshes constituted of tetrahedral cells, all faces are triangles. Consequently R_{ij} and $\xi_{ij,r}$ are independent of i and j and the previous scheme rewrites as a convex combination of the

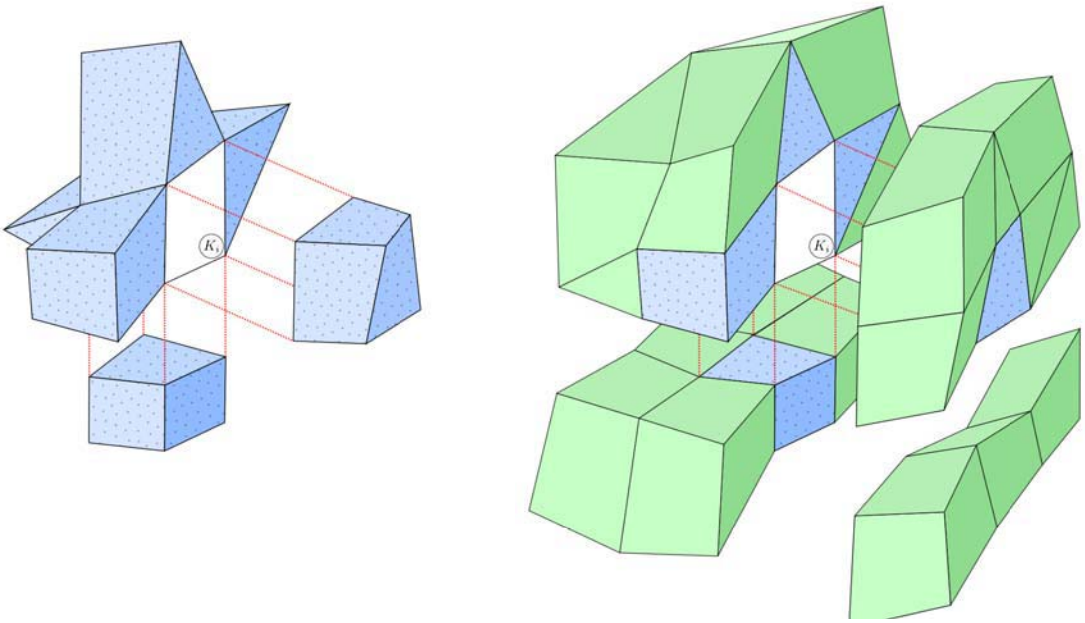


Figure 3.32: Illustrations for index sets $\underline{\nu}(i)$ (left) and $\overline{\nu}(i)$ (right) in 3D.

first-order scheme (3.41)

$$U_i^{n+1} = \sum_{r=1}^{R} \xi_r \left(U_i^n - \Delta t \sum_{j \in \underline{\nu}(i)} \frac{|f_{ij}|}{|K_i|} \mathbb{F}(U_{ij,r}^n, U_{ji,r}^n, \mathbf{n}_{ij}) \right). \tag{3.43}$$

Remark 3.20 *When dealing with general polyhedral cells, the quadrature rules may be different from a face to another and such a convex combination is not valid anymore. However, since each polygonal face can be split into triangles, one can recover equation (3.43) by considering each polyhedron as a polyhedron only constituted by triangular faces (and consequently with more faces than the original one).*

Let denote by $U_h^n = \sum_{i \in \mathcal{E}_{el}} U_i^n \mathbb{1}_{K_i}$ the constant piecewise representation of approximation $(U_i^n)_{i \in \mathcal{E}_{el}}$, we introduce operator $\mathcal{H}^R(U_h^n)$ such that relation (3.42) rewrites as

$$U_h^{n+1} = U_h^n + \Delta t \mathcal{H}^R(U_h^n). \tag{3.44}$$

From the original forward Euler discretization in time (3.44) we derive a high-order approximation in time using a Runge–Kutta 3 TVD method:

$$U_h^{n+1} = \frac{U_h^n + 2U_h^{(3)}}{3} \quad \text{with} \quad \left\{ \begin{array}{rcl} U_h^{(1)} & = & U_h^n + \Delta t \ \mathcal{H}^R(U_h^n) \\ U_h^{(2)} & = & U_h^{(1)} + \Delta t \ \mathcal{H}^R(U_h^{(1)}) \\ U_h^{(3)} & = & \widehat{U}_h^{(2)} + \Delta t \ \mathcal{H}^R(\widehat{U}_h^{(2)}) \end{array} \right. \tag{3.45}$$

where $\widehat{U}_h^{(2)}$ is the convex combination $(3U_h^n + U_h^{(2)})/4$.

The time discretization introduces a 3^{rd}-order error which makes the whole scheme to be formally 3^{rd}-order accurate. However setting $\Delta t = \Delta \mathbf{x}^{r/3}$ where r is the spatial order of accuracy and $\Delta \mathbf{x}$ is a characteristic length provide same order for spatial and time errors.

3.3.2.2 Reconstruction

We have formally defined an arbitrary high-order accurate Finite Volume scheme, providing that $U_{ij,r}$ is a high-order accurate point-wise approximation of $U(q_{ij,r})$ computed from cell K_i. In this subsection, we briefly describe the technique to produce such approximations and we refer to [18, 27] and references herein for details. Let us consider a scalar variable u and denote by $\widetilde{u}_i(\cdot,\mathsf{d})$ a local polynomial approximation of degree d reconstructed on cell K_i from the mean values of function u on a set of neighboring cells $\mathcal{S}_i^{\mathsf{d}}$ called stencil. For the sake of conservation, *i.e.* $\frac{1}{|K_i|}\int_{K_i} \widetilde{u}_i(\mathbf{x};\mathsf{d})\, d\mathbf{x} = u_i$, we assume that the polynomial has the following structure

$$\widetilde{u}_i(\mathbf{x};\mathsf{d}) = u_i \; + \sum_{1\leq|\boldsymbol{\alpha}|\leq\mathsf{d}} \mathcal{R}_i^{\boldsymbol{\alpha}} \left((\mathbf{x}-\mathbf{c})^{\boldsymbol{\alpha}} - \frac{1}{|K|}\int_K (\mathbf{x}-\mathbf{c})^{\boldsymbol{\alpha}}\, d\mathbf{x} \right), \tag{3.46}$$

where the polynomial coefficients $\mathcal{R}_i^{\boldsymbol{\alpha}}$ are fixed by solving a least-squares problem equivalent to minimizing the functional

$$E = \sum_{j\in\mathcal{S}_i^{\mathsf{d}}} \left(\frac{1}{|K_j|}\int_{K_j} \widetilde{u}_i(\mathbf{x};\mathsf{d})\, d\mathbf{x} - u_j \right)^2.$$

In practice, the polynomial coefficients are obtained by multiplying the pseudoinverse of the least-square problem matrix (that we store in memory) with the vector of mean values on the stencil, see [27] for details. We moreover recall that for the vectorial case, the reconstructions are performed for all the conservative components independently.

Finally, considering that polynomial reconstructions $\widetilde{u}_i(\mathbf{x};\mathsf{d})$ are provided for all cells K_i, $i \in \mathcal{E}_{el}$, we compute the approximation at each quadrature point of each face f_{ij} by $u_{ij,r} = \widetilde{u}_i(q_{ij,r};\mathsf{d})$. The so-called ($\mathsf{d}$-)unlimited scheme (3.42) is thus defined by employing the reconstructed values in the numerical flux without any restriction (*i.e.* no limitation).

Remark 3.21 *We recall that the reconstruction process is very time and memory consuming and would like to emphasize that contrarily to WENO methods we consider only one reconstruction stencil per cell and per degree, so that a lot of computational resources are saved.*

3.3.2.3 The MOOD concept

It is well-known that the first-order scheme (3.41) is robust but tremendously diffusive, while unlimited schemes of higher-order produce spurious oscillations in the vicinity of steep gradients. Limitation mechanisms have been developed to prevent the oscillations from appearing, such as

the slope or flux limitation in MUSCL methods [48, 83, 84, 49, 42, 13, 50] or the computation of an Essentially Non-Oscillatory polynomial reconstruction in WENO methods [39, 40, 70, 95, 69]. As we mention in the introduction, all the classical techniques act *a priori* in the sense that we determine the limitation process in function of the current data (*i.e.* the solution at time t^n). As a consequence, the *a priori* strategy imposes very drastic accuracy reduction due to strong and unnecessary limitations (the worst case scenario has to be considered). Moreover, computational resources are allocated to perform the limitation process where most of the time it is useless. In the ENO/WENO case for instance, several polynomial reconstructions are required even if the solution is locally regular and can be approximated with only one polynomial function.

In two recent papers [18, 27], we have introduced a new approach based on an *a posteriori* evaluation of the solution to determine if the limitation procedure has to be applied and where. The technique is *a posteriori* in the sense that we compute a candidate solution (a potential approximation at time t^{n+1}) and we use the data of the candidate solution to determine if the solution is valid. More precisely, the detection-limitation mechanism operates in several steps. A candidate solution is first computed with the highest-order unlimited scheme (the polynomials with maximal degree). Then a detection procedure is performed to determine the problematic cells, *i.e.* all cells where the approximation does not respect some given criteria (see next section). For problematic cells, the solution is recomputed with a lower-order unlimited scheme (using polynomials with lower degree) and we repeat the procedure detection-degree decrementing (the MOOD algorithm) till the cell satisfy the detection criteria or the polynomial degree is zero. In the last case, a robust first-order scheme (3.41) is triggered and a meaningful solution is thus provided. Note that we need to guarantee that the MOOD algorithm stops after a finite number of iterations.

We now set some fundamental notions to define the MOOD method. We name Cell Polynomial Degree, shortened as CellPD and denoted by d_i, the degree of the polynomial reconstruction on cell K_i. We name Face Polynomial Degrees, shortened as FacePD and denoted by d_{ij} and d_{ji}, the degrees of the polynomial reconstructions actually used to compute approximations, $U_{ij,r}$ and $U_{ji,r}$, of the solution on face f_{ij} at quadrature points $q_{ij,r}$ respectively from K_i and K_j. The computation of d_{ij} and d_{ji}, named FacePD strategy, consists in evaluating the FacePD $\mathsf{d}_{ij}, \mathsf{d}_{ji}$ that we employ on both sides of the interface f_{ij} with respect to the CellPD of the neighboring cells. In previous studies (see [18] for details), we have proposed and experimented several strategies and introduced the *upper-limiting* property for a FacePD strategy which states that for any degree $\bar{\mathsf{d}}$, the following property holds

$$\mathsf{d}_i = \bar{\mathsf{d}} \Longrightarrow \mathsf{d}_{ij} \leq \bar{\mathsf{d}} \quad \text{and} \quad \mathsf{d}_{ji} \leq \bar{\mathsf{d}}, \quad \forall j \in \underline{\nu}(i).$$

This guarantees (see [18]) that the MOOD algorithm stops after a finite number of iterations. In practice, we use the simple rule $\mathsf{d}_{ij} = \mathsf{d}_{ji} = \min(\mathsf{d}_i, \mathsf{d}_j)$.

As mentioned above, the detection mechanism is performed on the candidate solution $U_h^\star$ and criteria have to be set to specify what is a good solution. To this end, we denote by $\mathcal{A}$ the set of detection criteria (*e.g.* positivity of a variable or a maximum principle) that the numerical approximation has to respect on each cell and we say that a candidate solution is $\mathcal{A}$-eligible if

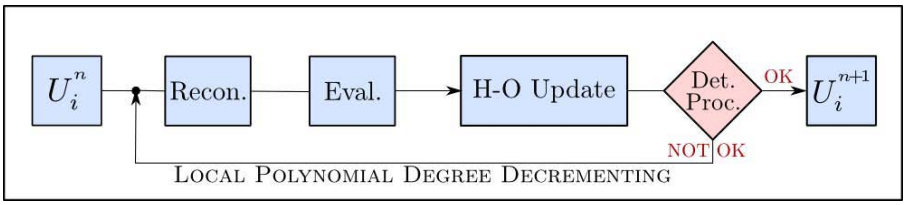


Figure 3.33: Flowchart of the MOOD algorithm: Recon. stands for polynomial reconstruction, Eval. for high-order evaluations at quadrature points, H-O Update for high-order update of the solution and Det. Proc. for detection process.

it fulfills all the criteria of $\mathcal{A}$. If the candidate solution is not $\mathcal{A}$-eligible on cell K_i, then we decrement the polynomial degree. However the solution may not be $\mathcal{A}$-eligible regardless of the set $\mathcal{A}$ even if the polynomial degree is zero for the cell. Consequently, we shall consider the solution *acceptable* on the cell if either it is $\mathcal{A}$-eligible, or is a first-order solution (*i.e.* CellPD has been decremented to zero). To sum up, the MOOD algorithm for the explicit discretization in time consists in the following stages depicted in Figure 3.33:

0. Initialize $\mathsf{d}_i = \mathsf{d}_{max}, \quad \forall i \in \mathcal{E}_{el}$.
1. Compute polynomial reconstruction of degree $\mathsf{d}_i, \quad \forall i \in \mathcal{E}_{el}$.
2. Compute FacePD d_{ij} and d_{ji} and evaluate high-order approximations at quadrature points on face $f_{ij}, \quad \forall j \in \underline{\nu}(i), \ \forall i \in \mathcal{E}_{el}$.
3. Compute candidate solution mean values through unlimited scheme (3.42), $\forall i \in \mathcal{E}_{el}$.
4. Detection process: decrement CellPD of cells where solution is not acceptable.
5. Stop if the solution is acceptable else go to stage 1.

Following [18, 27] we extend the MOOD algorithm initially designed for a one-time step scheme to the RK3-TVD scheme by applying it to each sub-step of the RK3-TVD (3.45) procedure. The MOOD method is now completely defined except from the detection criteria that have to be suited to the problem we intend to solve. Such a difficult task requires the complete next section.

3.3.3 Detection Criteria

The crucial point of the MOOD method is the elaboration of the detection criteria set $\mathcal{A}$ which characterizes the properties we want the numerical solution to fulfill. A fundamental purpose of the detection criteria is to obtain higher-order of accuracy for regular solutions while preventing numerical oscillations in the vicinity of discontinuous profiles. This would consequently provide an efficient and robust method. We face several difficulties to design such a set since accuracy and robustness are antagonist objectives. Moreover, in the Euler problem, a physically admissible solution is mandatory since the positivity of the density and the pressure is required to compute the numerical flux. It results that the detection criteria would cover a wide spectrum of properties and restrictions. A key point we shall detail in the following is the notion of "numerical regularity" in the sense that we have to determine if, for a local stencil and a set of data (for instance the mean values), we can associate a regular or a irregular function.

This point is really important since the choice of the reconstruction (namely the polynomial degree) depends on it.

The present section intends to extend and improve detection criteria initially introduced in [27] to evaluate the local "numerical regularity" of the approximation. We first begin the study for the advection equation in section 3.3.3.1 and address the hydrodynamics Euler system in section 3.3.3.2.

3.3.3.1 Advection equation

The scalar advection problem is characterized by the physical flux $F(U) = VU$ where $V \in \mathbb{R}^3$ stands for the velocity that we assume to be a regular function on Ω and satisfies $\nabla_{\mathbf{x}} V(\mathbf{x}) = 0$ while $U = U(t, \mathbf{x}) \in \mathbb{R}$ is the passive scalar quantity transported by the fluid.

When dealing with a constant velocity, the exact solution is simply given by $U(\mathbf{x}, t) = U_0(\mathbf{x} - Vt)$ and clearly fulfills a maximum principle, *e.g.* the minimum of the solution can not be lower than the initial condition minimum (and a similar property for the maximum). Consequently, it seems natural to impose such a condition at the numerical level and, as proposed in [18], we integrate in the set $\mathcal{A}$ the Discrete Maximum Principle (DMP) on mean values for the candidate solution $U_h^\star$ formulated like this:

$$\min_{j \in \overline{\nu}(i)} (U_i^n, U_j^n) \leq U_i^\star \leq \max_{j \in \overline{\nu}(i)} (U_i^n, U_j^n). \tag{3.47}$$

A solution is $\mathcal{A}$-eligible if condition (3.47) is satisfied for all the cells and we have proved in [18] that the scheme equipped with such set $\mathcal{A}$ provides a numerical solution which, under first-order scheme CFL condition, satisfies the DMP. However a strict application of relation (3.47) at smooth extrema unavoidably reduces the scheme accuracy to two. It suggests that relation (3.47) is too restrictive and should be relaxed.

In [27] we have relaxed the condition on cells which violate the DMP. More specifically, the relation (3.47) has been supplemented with a new criteria, the so-called $u2$ detection criteria which provides an effective arbitrary high-order of accuracy. As mention in the beginning of the section, the key point is to determine if the numerical solution is regular enough to be approximated by a high-order polynomial reconstruction and avoid the Gibbs phenomena. To this end, let assume that the candidate solution does not satisfy the DMP criteria on cell K_i. A first step consists in reconstructing quadratic polynomials on K_i denoted by $\widetilde{U}_i$ and on its neighbors K_j for $j \in \overline{\nu}(i)$ denoted by $\widetilde{U}_j$. In a second step, we define approximations to the local minimal and maximal curvatures, namely

$$\mathcal{X}_i^{min} = \min_{j \in \overline{\nu}(i)} \left(\partial_{xx} \widetilde{U}_i,\ \partial_{xx} \widetilde{U}_j \right) \quad \text{and} \quad \mathcal{X}_i^{max} = \max_{j \in \overline{\nu}(i)} \left(\partial_{xx} \widetilde{U}_i,\ \partial_{xx} \widetilde{U}_j \right), \tag{3.48}$$

$$\mathcal{Y}_i^{min} = \min_{j \in \overline{\nu}(i)} \left(\partial_{yy} \widetilde{U}_i,\ \partial_{yy} \widetilde{U}_j \right) \quad \text{and} \quad \mathcal{Y}_i^{max} = \max_{j \in \overline{\nu}(i)} \left(\partial_{yy} \widetilde{U}_i,\ \partial_{yy} \widetilde{U}_j \right), \tag{3.49}$$

$$\mathcal{Z}_i^{min} = \min_{j \in \overline{\nu}(i)} \left(\partial_{zz} \widetilde{U}_j,\ \partial_{zz} \widetilde{U}_j \right) \quad \text{and} \quad \mathcal{Z}_i^{max} = \max_{j \in \overline{\nu}(i)} \left(\partial_{zz} \widetilde{U}_i,\ \partial_{zz} \widetilde{U}_j \right), \tag{3.50}$$

where we emphasize that the second derivatives are constant and naturally referred to as *curvatures*. The $u2$ detection criterion holds in the following definition.

Definition 3.22 ($u2$ detection criterion) *A candidate solution $U_i^\star$ in cell K_i which violates the DMP is nonetheless eligible if the following holds*

$$\mathcal{X}_i^{max}\mathcal{X}_i^{min} > 0 \quad and \quad \left|\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}}\right| \geq 1-\varepsilon,$$

$$and \quad \mathcal{Y}_i^{max}\mathcal{Y}_i^{min} > 0 \quad and \quad \left|\frac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}}\right| \geq 1-\varepsilon,$$

$$and \quad \mathcal{Z}_i^{max}\mathcal{Z}_i^{min} > 0 \quad and \quad \left|\frac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}}\right| \geq 1-\varepsilon,$$

where ε is a smoothness parameter.

The definition derives from the idea that the comparison of local second derivatives of the quadratic reconstructions on a neighborhood provides a relevant information on the numerical smoothness of the underlying solution. More precisely, we consider that the underlying solution (characterized by the piecewise constant mean value) is (ε-)smooth if for each direction the curvatures have the same sign (no oscillation or inflection point) and are (ε-)*close* enough to each-other. Such a definition lies in a fitting of the parameter ε the value of which defines the threshold between what is considered as smooth extrema or as discontinuity. Therefore the determination of ε is of crucial importance since it rules the decrementing process activation.

From a practical point of view, the $u2$ detection criteria operates in two stages. First the test on the sign of curvatures (left inequalities) is performed. If oscillations are detected, *i.e.* the product is negative, the cell is considered as problematic and the decrementing procedure must be applied. The second stage is performed only if the product is positive. It consists in computing the ratios between minimal and maximal curvatures and comparing it to $1-\varepsilon$ (right inequalities). If the curvatures ratio does not respect the inequality, the cell is considered as problematic and the decrementing process must be applied.

In [27], we propose a parameter ε depending on a local characteristic length and on the spatial dimension of the domain. This was a first attempt to the determination of ε and deeper investigations have shown that a simpler definition provides same quality results. To set the ε value, we extend the parameter as a new function $\varepsilon_x = \varepsilon_x\left(\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}}\right)$ (for the x-direction) with respect to the curvatures which have to satisfy the restriction

$$\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}} \geq 1-\varepsilon_x. \tag{3.51}$$

The goal is to determine a relevant function ε_x which enables high-order approximation and robustness. We first note that the curvatures ratio ranges between zero and one so that ε_x

must range in $[0,1]$ to make sense. Moreover, the ratio is expected to be close to zero on discontinuities and close to one on smooth functions which are the two extreme cases. For a non-smooth function, we expect that the limiting procedure operates and since the closer to zero ε is, the less smooth the underlying function is considered, ε_x is expected to be close to zero on discontinuities, *i.e.* $\lim_{r\to 0^+} \varepsilon_x(r) = 0^+$. On the other hand, a ratio close to one indicates smooth functions, so we expect $\lim_{r\to 1^-} \varepsilon_x(r) = 1^-$ to relax the restriction. We thus propose to define ε_x as a continuous increasing function of the curvatures ratio such that $\varepsilon_x(0) = 0$ and $\varepsilon_x(1) = 1$. After several attempts, it appears that the simple function $\varepsilon_x(r) = r$ is an excellent choice. When substituting expression of $\varepsilon_x = \mathcal{X}_i^{min}/\mathcal{X}_i^{max}$ in relation (3.51), the x-direction curvatures criterion becomes

$$\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}} \geq 1 - \frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}},$$

and yields

$$\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}} \geq 1/2.$$

Finally we apply the same reasoning for y- and z-directions and obtain

$$\frac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}} \geq 1/2 \quad \text{and} \quad \frac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}} \geq 1/2.$$

The linearity of function ε_x simplifies the final inequalities and leads to the constant value $\varepsilon = 1/2$ in definition 3.22.

Remark 3.23 *The definition of ε is really simpler than the one proposed in [27]. However numerous numerical test cases have been carried out and no change in the quality of results have been reported.*

Remark 3.24 *Numerical experiments show that the choice of the neighborhood where the curvatures are computed should define a convex hull which contains the reference cell K_i. To constitute such a stencil, we used the index set of cells $\underline{\nu}(i)$ in 2D (see [27]) but this choice is not relevant for three-dimensional meshes and we use the index set $\overline{\nu}(i)$ in equations (3.48)-(3.50) to provide the expected results even for large form factor meshes.*

To conclude the section we propose in Figure 3.34 an algorithmic view of the complete detection process [DMP $\to u2$] for the advection equation constituted of the DMP of equation (3.47) relaxed by the $u2$ detection criteria of definition 3.22. We emphasize that the algorithm is given in the case of a cell K_i with $U_i^\star$ its associated candidate solution mean value.

If $\left[U_i^\star \leq \min_{j\in\overline{\nu}(i)}\left(U_i^n, U_j^n\right)\right]$ **or** $\left[U_i^\star \geq \max_{j\in\overline{\nu}(i)}\left(U_i^n, U_j^n\right)\right]$ **then**

- decrement the CellPD d_i

else

→ compute the curvatures $\mathcal{X}_i^{min}, \mathcal{X}_i^{max}, \mathcal{Y}_i^{min}, \mathcal{Y}_i^{max}, \mathcal{Z}_i^{min}, \mathcal{Z}_i^{max}$

If $\left[\mathcal{X}_i^{min}\mathcal{X}_i^{max} < 0\right]$ **or** $\left[\mathcal{Y}_i^{min}\mathcal{Y}_i^{max} < 0\right]$ **or** $\left[\mathcal{Z}_i^{min}\mathcal{Z}_i^{max} < 0\right]$ **then**

- decrement the CellPD d_i

else

→ compute the curvatures ratios $\dfrac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}}, \dfrac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}}, \dfrac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}}$

If $\left[\dfrac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}} < 1-\varepsilon\right]$ **or** $\left[\dfrac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}} < 1-\varepsilon\right]$ **or** $\left[\dfrac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}} < 1-\varepsilon\right]$ **then**

- decrement the CellPD d_i

end if

end if

end if

Figure 3.34: Algorithmic view of the [DMP→$u2$] detection process for the advection equation.

3.3.3.2 Hydrodynamics Euler system

The Euler system for three-dimensional geometries writes

$$\partial_t \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{pmatrix} + \partial_x \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{pmatrix} + \partial_y \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{pmatrix} + \partial_z \begin{pmatrix} \rho v \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{pmatrix} = 0, \tag{3.52}$$

where ρ stands for the density, u, v and w for the velocity components in the x, y and z directions respectively, p for the pressure and E for the total energy. This system is closed by the Equation Of State (EOS) of a perfect gas $p = (\gamma - 1)\rho\epsilon$, where ϵ is the specific internal energy, γ the ratio of specific heats and the total energy is constituted of the internal and kinetic energy

$$E = \rho\left((u^2 + v^2 + w^2)/2 + \epsilon\right].$$

At last, vector $U = (\rho, \rho u, \rho v, \rho w, E)$ represents the conservative variables of the system while $W = (\rho, u, v, w, p)$ are the primitive ones. Note that contrarily to WENO methods we do not use the characteristic variables.

To provide accurate and oscillation-free solutions we use on the one hand polynomial reconstruction and apply, on the other hand the MOOD algorithm. We mention that all the polynomial reconstructions are performed on conservative variables and only one CellPD is used for all variables (see [27] for the motivations and justifications). We now turn to the detection-limitation procedure and we have to design specific detection criteria for the Euler problem.

Following [27], a first and also mandatory detection criteria corresponds to ensuring the physical meaningfulness of the primitive variables. We then introduce the Physical Admissibility Detection (PAD in short) which considers that the candidate solution on a cell K_i is not valid if we have $\rho_i^\star$ or $p_i^\star$ are negative (after having computed pressure $p_i^\star$). We underline the important property that a high-order scheme (whichever the degree of the polynomial reconstruction) equipped with the PAD and a first-order scheme which preserves the positivity (of density and pressure) under a CFL condition is automatically positivity preserving. This property straightforwardly derives from the *a posteriori* nature of the MOOD method and has been proved in [27]. However the PAD detection process does not prevent spurious oscillations from appearing and we turn to the adaptation of the [DMP $\rightarrow u2$] detection process proposed in [27]. Initially defined for scalar quantity, we apply the [DMP $\rightarrow u2$] on the density ρ (detection) and recall that the decrementing is performed for all variables (limitation). Note that the smoothness parameter ε is still set as $1/2$ in the $u2$ definition as in previous section.

The set of constraints $\mathcal{A}$ for Euler system is thus constituted by the PAD followed by the [DMP $\rightarrow u2$] detection process applied to the density variable since we first check the PAD and if the cell is valid we continue with the [DMP $\rightarrow u2$] detection. In Figure 3.35 we give an algorithmic view of the complete detection process [PAD $\rightarrow$ DMP $\rightarrow u2$] for the hydrodynamics Euler system constituted of the PAD detection criteria, the DMP of equation (3.47) on the density relaxed by the $u2$ detection criteria of definition 3.22. We emphasize that the algorithm is given in the case of a cell K_i with $U_i^\star = \left(\rho_i^\star, (\rho u)_i^\star, (\rho v)_i^\star, (\rho w)_i^\star, E_i^\star\right]$ its associated candidate solution mean value and that the candidate pressure $p_i^\star$ has to be computed.

We now highlight some implementation aspects about the detection process which enable to improve the solution accuracy. Actually in the above algorithm, the [PAD $\rightarrow$ DMP $\rightarrow u2$] performs well but does not, in some cases, fully reach the optimal order of accuracy for smooth solutions. Deeper investigations on the isentropic vortex in motion problem have shown that the detection process inappropriately decrements some cells of the flat region while it operates well in the area where curvatures are not negligible. The undesirable limitation derives from the extra-small *curvatures* treatment by the $u2$ detection where some spurious micro-oscillations take place on the flat area and wrongly activate the curvature sign detection. It results that the sign criterion is not relevant when all the *curvatures* sizes are too small with respect to a mesh parameter δ. To overcome the over-detection phenomena, we introduce a relaxation parameter in the $u2$ criterion to fix the problem.

If $[\rho_i^\star \leq 0]$ **or** $[p_i^\star \leq 0]$ **then**

- decrement the CellPD d_i

else

If $\left[\rho_i^\star \leq \min\limits_{j\in\overline{\nu}(i)}(\rho_i^n, \rho_j^n)\right]$ **or** $\left[\rho_i^\star \geq \max\limits_{j\in\overline{\nu}(i)}(\rho_i^n, \rho_j^n)\right]$ **then**

- decrement the CellPD d_i

else

→ compute the curvatures $\mathcal{X}_i^{min}, \mathcal{X}_i^{max}, \mathcal{Y}_i^{min}, \mathcal{Y}_i^{max}, \mathcal{Z}_i^{min}, \mathcal{Z}_i^{max}$ from density ρ at t^n

If $\left[\mathcal{X}_i^{min}\mathcal{X}_i^{max} < 0\right]$ **or** $\left[\mathcal{Y}_i^{min}\mathcal{Y}_i^{max} < 0\right]$ **or** $\left[\mathcal{Z}_i^{min}\mathcal{Z}_i^{max} < 0\right]$ **then**

- decrement the CellPD d_i

else

→ compute the curvatures ratios $\dfrac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}}, \dfrac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}}, \dfrac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}}$

If $\left[\dfrac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}} < 1-\varepsilon\right]$ **or** $\left[\dfrac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}} < 1-\varepsilon\right]$ **or** $\left[\dfrac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}} < 1-\varepsilon\right]$ **then**

- decrement the CellPD d_i

end if

end if

end if

end if

Figure 3.35: Algorithmic view of the [PAD→DMP→$u2$] detection process for the Euler system.

Definition 3.25 (u2 detection criterion) *A candidate solution $U_i^\star$ in cell K_i for which the density $\rho_i^\star$ violates the DMP is nonetheless eligible if*

$$\mathcal{X}_i^{max}\mathcal{X}_i^{min} > -\delta \quad \textit{and} \quad \left(\max\left(|\mathcal{X}_i^{max}|, |\mathcal{X}_i^{min}|\right) < \delta \quad \textit{or} \quad \left|\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}}\right| \geq 1/2\right),$$

$$\textit{and} \quad \mathcal{Y}_i^{max}\mathcal{Y}_i^{min} > -\delta \quad \textit{and} \quad \left(\max\left(|\mathcal{Y}_i^{max}|, |\mathcal{Y}_i^{min}|\right) < \delta \quad \textit{or} \quad \left|\frac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}}\right| \geq 1/2\right),$$

$$\textit{and} \quad \mathcal{Z}_i^{max}\mathcal{Z}_i^{min} > -\delta \quad \textit{and} \quad \left(\max\left(|\mathcal{Z}_i^{max}|, |\mathcal{Z}_i^{min}|\right) < \delta \quad \textit{or} \quad \left|\frac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}}\right| \geq 1/2\right),$$

where δ is the greatest length of geometrical entity of dimension one defined by the length of the cells in $\mathbb{R}$, the maximal length of the cell interfaces in $\mathbb{R}^2$ and the maximal length of edges of the cell interface for three-dimensional meshes.

The correction only damps extra-small oscillations such that minimal and maximal curvatures product satisfies the left condition. When maximal curvatures are larger than δ, the condition

on the ratios of curvatures implies that the underlying function will be considered as non-smooth.

Remark 3.26 *The value of δ has been determined after numerous simulation experiments. It enables to fully reach the optimal order for the Euler system but does not affect the method in wisely capturing discontinuous profiles. The correction has even been tested for the convection equation and accuracy losses have not been reported.*

In the same way, we slightly relax the DMP criteria to reduce the computational effort to avoid the waste of resources when performing the $u2$ detection criterion on plateaus. We consider that a DMP violation is not relevant if

$$\max_{j\in\overline{\nu}(i)}(\rho_i^{RK}, U_j^{RK}) - \min_{j\in\overline{\nu}(i)}(\rho_i^{RK}, U_j^{RK}) < \delta^3$$

where index ^{RK} corresponds to one of the Runge-Kutta sub-steps.
The MOOD method for the Euler hydrodynamics system is now completely defined and numerical simulations are carried out for three-dimensional geometries presented in section 3.3.4.

3.3.3.3 Implementation and optimizations

To conclude the section we detail two important and simple optimizations that we apply to drastically improve the efficiency of the MOOD method.

Local re-updating. The MOOD method may seem computationally expensive since the MOOD algorithm we run for each time step, recompute the candidate solution several times whereas polynomial degrees have been only modified for a small number cells. At the first stage, an initial candidate solution is computed on all cells. Then the MOOD algorithm successively detects and limits the problematic cells. The evaluation of a new candidate solution during the MOOD algorithm by means of scheme (3.43) only involves the fluxes at the interfaces of corrected cells. Consequently only problematic cells and their neighbors by face must be re-computed. It drastically reduces the computational effort since in most cases the solution is acceptable on more than 80–90% of cells, even when shocks are present.

Reduced polynomial degree decrementing. The original decrementing procedure consists in dropping one-by-one polynomial degrees until zero is reached. Such an approach may be both costly in CPU and memory resources since reconstruction matrices must be stored for all degrees. It would nonetheless still be less memory consuming than for the WENO method due to the large number of polynomial functions involved in the WENO technique. Moreover numerical experiments suggest the following alternative: whether the solution is very smooth, whether the solution presents some discontinuities. To take advantage of it, we change the decrementing strategy by starting from the highest degree, reducing to degree 2 if any and setting degree equal to 0 if the candidate solution is still not $\mathcal{A}$-eligible. We then manage to reduce the number of decrementing stages and save computational resources. We point out that the size of the reconstruction stencil is also an important parameter since a large stencil (required for the maximal degree) will be influenced by a discontinuity located in the second or

third layer of cells around the reference one while a more compact one (for a $\mathbb{P}_2$ reconstruction) still preserves the local regularity of the underlying function. Another reason to use the $\mathbb{P}_2$ reconstruction is that it is also used for the $u2$ detection process and always has to be stored. Therefore in practice, we only store two reconstruction matrices per cell, one for the maximal degree and one for the degree two. It is thus important to remark that the storage cost of the matrix for degree two is always much lower than the one for the maximal degree. Indeed for two-dimensional situations, the memory cost of the pseudoinverse matrix associated to polynomial of degree 2 represents about 10 times 5 elements, about 16 times 9 for $\mathbb{P}_3$ reconstruction and around 28 times 20 for $\mathbb{P}_5$. Analogically for three-dimensional situations, the $\mathbb{P}_2$ reconstruction matrix represents about 16 times 9 elements while it is about 38 times 19 for $\mathbb{P}_3$ and 110 times 55 for $\mathbb{P}_5$.

To conclude this section, we would like to draw some remarks about the potentiality of the MOOD method to be parallelized. Within the MOOD algorithm, only classical unlimited schemes are used without modification so that the parallelization of this part of the method can be done as efficiently as the state-of-the-art methods (WENO method for instance). The only novelty brought by the MOOD method is the iterative process constituting the MOOD algorithm. A potential difficulty comes from the fact that the number of cells on which the numerical scheme acts changes from an iteration of the MOOD algorithm to another, since the procedure is only applied to problematic cells. However it may not dramatically affect the parallelization efficiency: firstly, because an efficient treatment of the list of problematic cells can be achieved and secondly, because the time spent to recompute new candidate solutions is negligible compared to the time to compute the initial one since the number of problematic cells is (in general) very low compared to the total number of cells. The parallelization capacity of the MOOD method is thus as good as the state-of-the-art higher-order finite volume methods.

3.3.4 Numerical results

The MOOD method has been implemented into a 3D unstructured code dealing with polyhedra having coplanar faces: tetrahedron, hexahedron, pyramid and prism. The polynomial reconstruction procedure is implemented independently of the degree d_{max} and we provide in the present paper numerical results up to $\mathsf{d}_{max} = 5$. Following [27] and remarks in section 3.3.3.3, we use the decrementing sequence $\mathbb{P}_{\mathsf{d}_{max}} - \mathbb{P}_2 - \mathbb{P}_0$. The reconstruction matrices are computed and stored in a preprocessing step since they only depend on geometry. Moreover fluxes across faces are approximated by the mean of Gaussian quadrature formulae on a triangular decomposition of the faces (see Figure 3.31). At last concerning the time discretization, the first-order time step Δt is controlled by a CFL coefficient equal to 0.5. For the convergence studies on smooth solutions we use the time step $\Delta t = \Delta x^{r/3}$ to achieve a global r^{th}-order of accuracy and compute the relative L^1 and L^∞ errors for a a bounded, L^1 function φ by

$$L^1 \text{ error: } \frac{\sum_{i\in\mathcal{E}_{el}} |\varphi_i^N - \varphi_i^{ex}||K_i|}{\sum_{i\in\mathcal{E}_{el}} |\varphi_i^{ex}||K_i|} \quad \text{and} \quad L^\infty \text{ error: } \frac{\max_{i\in\mathcal{E}_{el}} |\varphi_i^N - \varphi_i^{ex}|}{\max_{i\in\mathcal{E}_{el}} |\varphi_i^{ex}|},$$

where $(\varphi_i^{ex})_{i\in\mathcal{E}_{el}}$ and $(\varphi_i^N)_{i\in\mathcal{E}_{el}}$ are respectively the exact and the approximated cell mean values at final time $t = t_{\text{final}}$.

3.3.4.1 Advection equation

For the scalar advection equation, the MOOD method is employed with the [DMP $\rightarrow u2$] detection process and two test cases are carried out: the Triple Sine Translation (TST) to assess the effective very high-order of accuracy and the rotation of a discontinuous H-like shape to test its ability to damp the spurious oscillations.

⋆ Triple Sine Translation

Let Ω be the unit cube. We consider a constant translation velocity $V = (1, 1, 1)$ and the C^∞ initial condition

$$U_0(x, y, z) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z).$$

The final time is $t_{\text{final}} = 2.0$ and periodic boundary conditions imply that the exact final solution coincides with the initial one. The computations are first carried out on a series of successively refined regular hexahedral meshes from 8^3 to 64^3 cells. To underline the capability of the MOOD method to handle mixed element meshes, we also consider a series of meshes built from a series of regular hexahedral meshes from 4^3 to 48^3 cells into which we regularly split half of cells into 6 pyramids (see top line of Figure 3.36).

In Figure 3.36, we display the convergence curves for the L^1 and L^∞ errors of the MOOD-$\mathbb{P}_2$, MOOD-$\mathbb{P}_3$ and MOOD-$\mathbb{P}_5$ methods and give in Table 3.17, the corresponding errors and rates of convergence. As expected, the optimal rate of convergence is achieved. Notice that on the coarsest meshes the initial mean values are not representative of the underlying smooth function and are coherently handled by the method as discontinuous profiles. As such the sine function is under-resolved; for instance in 1D, averaging the function $\sin(2\pi x)$ or an Heaviside-like function on $[0; 1]$ using four cells provides to same mean values.

⋆ H-like shape rotation

We now turn to the rotation of an H-like shape in the unit cube Ω. The initial shape is given by

$$U_0(x, y, z) = \begin{cases} 1 & \text{if } (|x - 0.5| > 0.1) \text{ or } (|y - 0.5| < 0.1), \\ 0 & \text{elsewhere,} \end{cases}$$

in the cube $[0.2; 0.8]^3$ and 0 elsewhere. The rotation axis is the diagonal line joining the origin $(0, 0, 0)$ and the point $(1, 1, 1)$. We stop the simulation after one full rotation when the shape is back to its original position. Note that the velocity depends on the spatial position but is divergence-free so that the maximum principle also applies in that case. Numerical simulations are carried out on a 86215 tetrahedra mesh generated by the free mesher Gmsh. Results are displayed with an extruded view on the cut plane $z = 1/2$. Initialization details are illustrated in Figure 3.37.

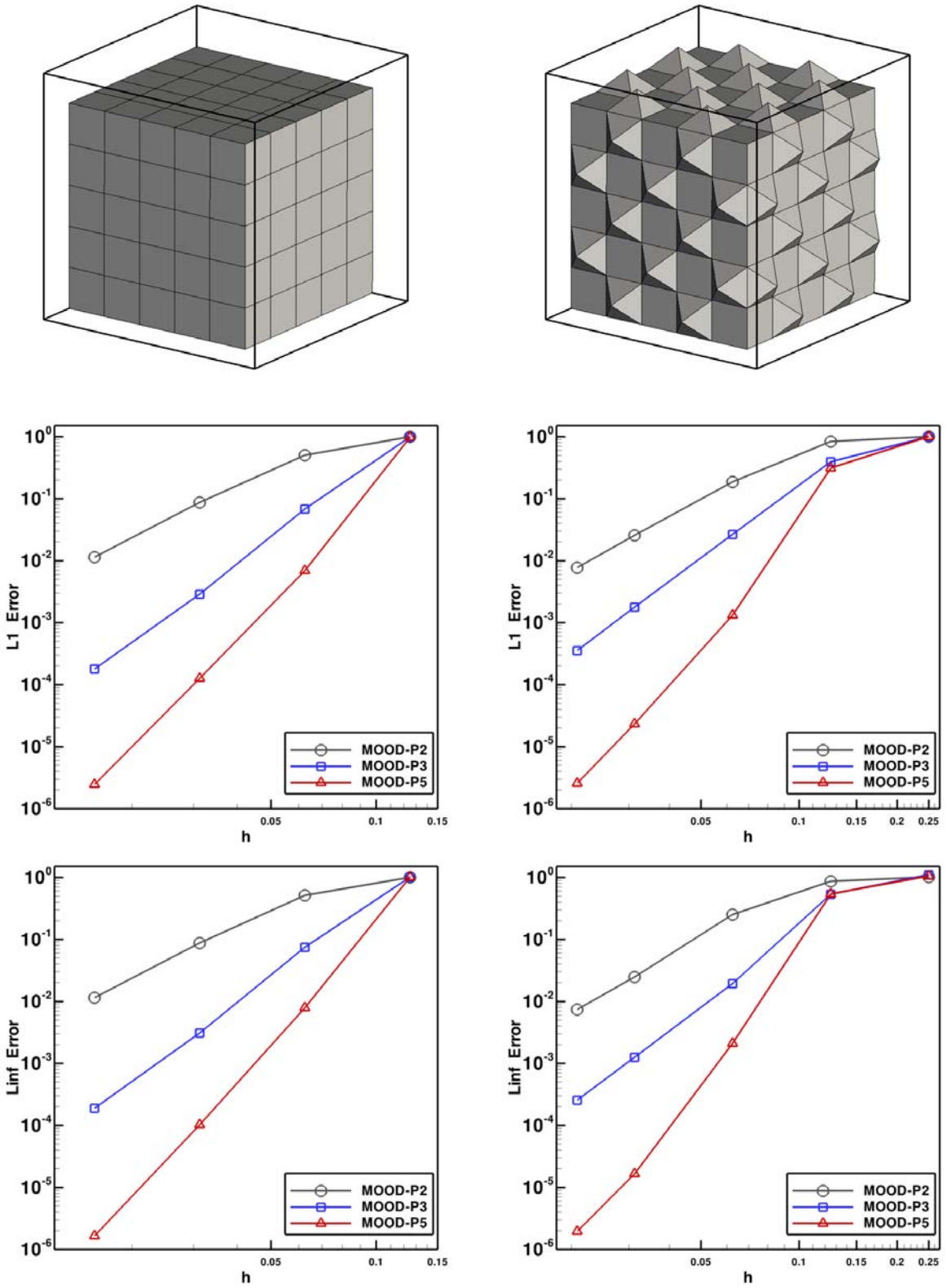


Figure 3.36: Triple sine translation: convergence curves for L^1 (middle) and L^∞ (bottom) errors for series of hexahedral (left) and hexahedral/pyramidal (right) meshes. Examples of such meshes are given on top line.

MOOD on hexahedra							
Deg.	h	L^1 error		L^2 error		L^∞ error	
$\mathbb{P}_2$	0.125	9.9682484294e-1	—	9.9680699621e-1	—	1.0038484635	—
	0.0625	5.0100730661e-1	0.99	5.0179199583e-1	0.99	5.1327069642e-1	0.97
	0.03125	8.6946371321e-2	2.53	8.6934402652e-2	2.53	8.7634409525e-2	2.55
	0.015625	1.1429255953e-2	2.93	1.1424251727e-2	2.93	1.1459635442e-2	2.93
Expected order			3		3		3
$\mathbb{P}_3$	0.125	9.8673015719e-1	—	9.8688657294e-1	—	1.0110942145	—
	0.0625	6.8019691177e-2	3.86	6.9068026293e-2	3.84	7.4325569597e-2	3.77
	0.03125	2.8693653411e-3	4.57	2.7741019990e-3	4.64	3.0819084097e-3	4.59
	0.015625	1.7856449887e-4	4.01	1.5709922281e-4	4.14	1.8874795858e-4	4.03
Expected order			4		4		4
$\mathbb{P}_5$	0.125	9.7842521971e-1	—	9.7924733246e-1	—	1.0169454936	—
	0.0625	6.9230110414e-3	7.14	6.9967747247e-3	7.13	7.9478234947e-3	7.00
	0.03125	1.2666634416e-4	5.77	1.1021776542e-4	5.99	1.0247433118e-4	6.28
	0.015625	2.4614368833e-6	5.69	2.0386571852e-6	5.76	1.6605387870e-6	5.95
Expected order			6		6		6

MOOD on mixed hexahedra/pyramids							
Deg.	h	L^1 error		L^2 error		L^∞ error	
$\mathbb{P}_2$	0.25	1.0000027468	—	1.0000052406	—	1.0143912168	—
	0.125	8.3799247906e-1	0.25	8.3412664416e-1	0.26	8.6799172420e-1	0.20
	0.0625	1.8662020042e-1	2.17	1.8646014762e-1	2.16	2.5210598518e-1	1.78
	0.03125	2.5647018453e-2	2.86	2.4729005004e-2	2.91	2.4798614346e-2	3.35
	0.020833	7.6897099615e-3	2.97	7.4071918780e-3	2.97	7.3982102275e-3	2.98
Expected order			3		3		3
$\mathbb{P}_3$	0.25	9.9952627605e-1	—	1.0018017690	—	1.1083447073	—
	0.125	3.9219135702e-1	1.35	4.1718119801e-1	1.26	5.3531820180e-1	1.05
	0.0625	2.6501056786e-2	3.89	2.2797888150e-2	4.19	1.9364138004e-2	4.79
	0.03125	1.7829686262e-3	3.90	1.5178093945e-3	3.91	1.2521397100e-3	3.95
	0.020833	3.5401059785e-4	3.99	3.0038884987e-4	4.00	2.5551614705e-4	3.92
Expected order			4		4		4
$\mathbb{P}_5$	0.25	1.0009285907	—	1.0025919881	—	1.0496962436	—
	0.125	3.1141644019e-1	1.68	4.0086400280e-1	1.32	5.4249781220e-1	0.95
	0.0625	1.3246287256e-3	7.88	1.1541322861e-3	8.44	2.1098853389e-3	8.00
	0.03125	2.3443624169e-5	5.82	2.0015998229e-5	5.85	1.6596915121e-5	6.99
	0.020833	2.0207215760e-6	6.05	1.7188485947e-6	6.05	1.5127171717e-6	5.91
Expected order			6		6		6

Table 3.17: L^1, L^2 and L^∞ errors and convergence rates for the TST problem with the MOOD-$\mathbb{P}_2$, MOOD-$\mathbb{P}_3$ and MOOD-$\mathbb{P}_5$ methods. Top lines: hexahedral meshes. Bottom lines: mixed hexahedral/pyramidal meshes.

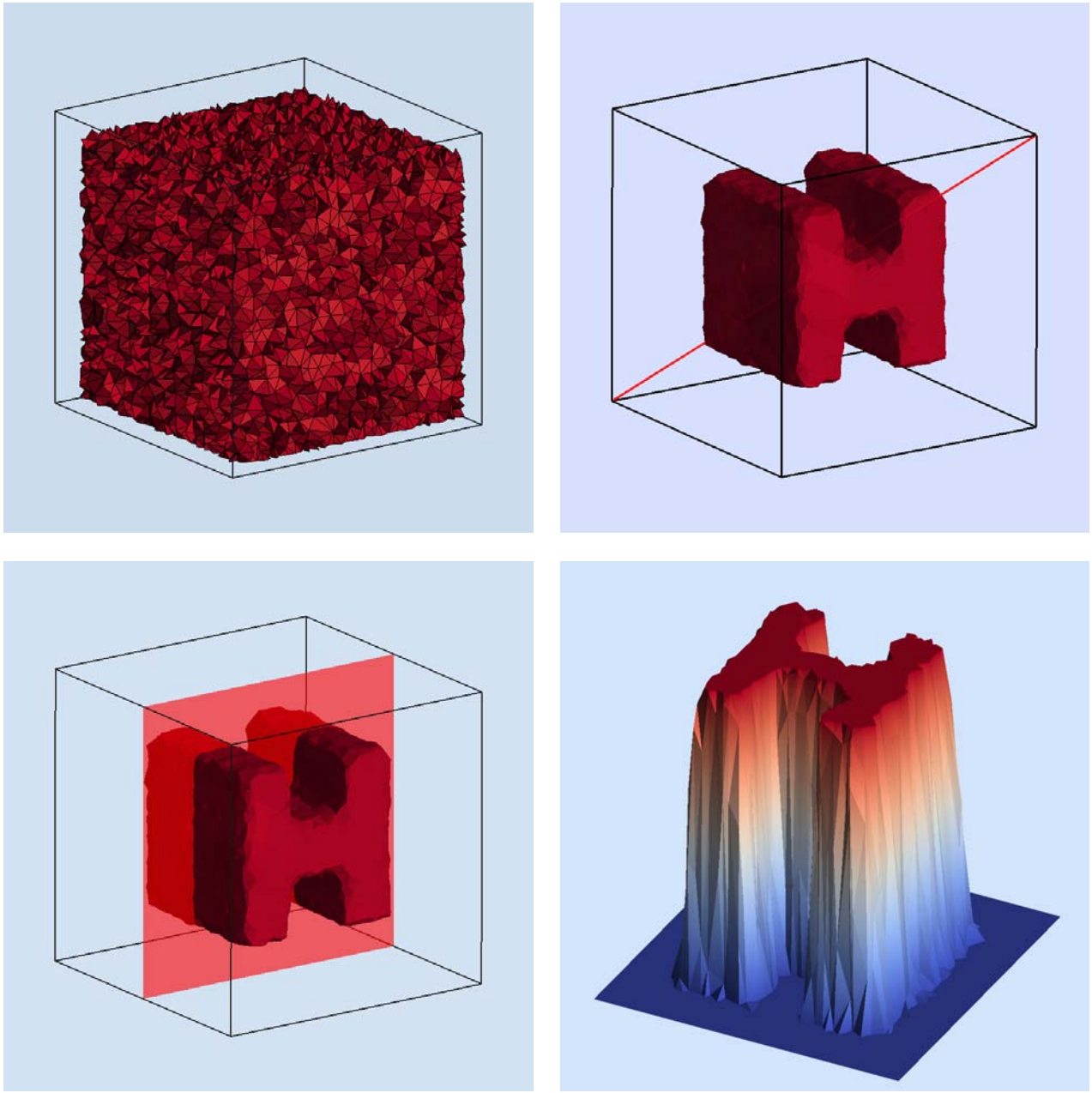


Figure 3.37: Initialization of the H-like shape rotation problem. Top left: interior view of the tetrahedral mesh. Top right: initialization of the H-like shape (isosurface 1/2, rotation axis is the red line). Bottom left: cut plane $z = 1/2$. Bottom right: extruded initial values from the cut plane.

We plot in Figure 3.38 the solution on the cut plane $z = 1/2$ for the unlimited $\mathbb{P}_3$ and $\mathbb{P}_5$ schemes and the MOOD-$\mathbb{P}_3$, MOOD-$\mathbb{P}_5$ methods. We notice that the unlimited schemes produce oscillations, depicted in green in the figure, whereas the MOOD method provides an oscillation-free solution even for polynomials of degree 5. It highlights the capacity of the [DMP $\rightarrow u2$] detection process to correctly treat discontinuous shapes on genuinely unstructured 3D meshes.

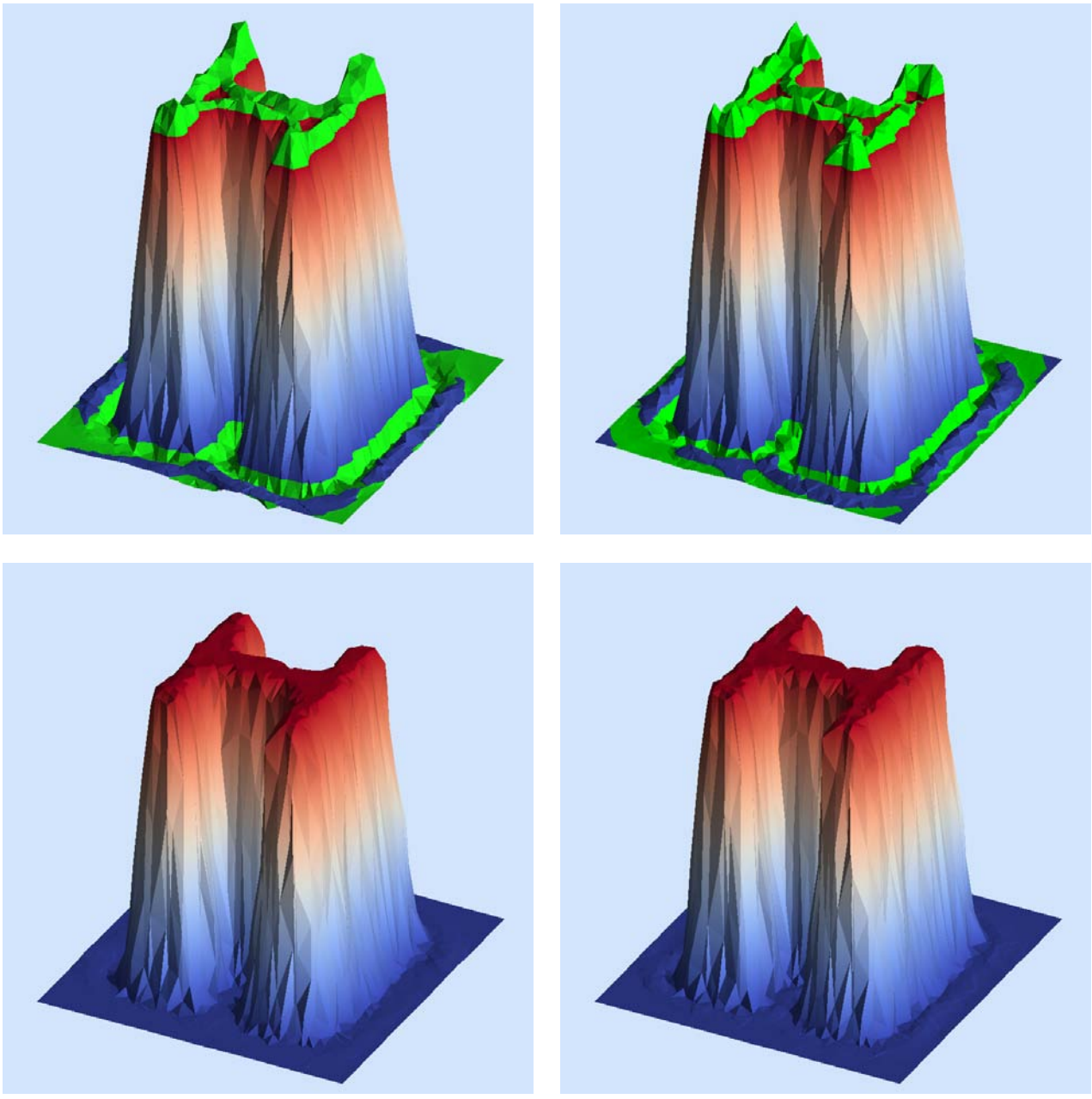


Figure 3.38: Results of the H-like shape rotation problem on the cut plane $z = 1/2$ for the unlimited $\mathbb{P}_3$ and $\mathbb{P}_5$ schemes (top line) and for the MOOD $\mathbb{P}_3$ and $\mathbb{P}_5$ methods (bottom line). The highlighted green cells correspond to values below 0 or above 1.

3.3.4.2 Euler system

We now consider the three-dimensional hydrodynamics Euler system on unstructured meshes. The first test cases proposed in section 3.3.4.2 deal with the Sod and Lax shock tubes following the Ox axis (invariant with respect to the other directions). The simulations are carried out on a tetrahedral mesh to study the MOOD method capacity to handle simple waves. Section 3.3.4.2 is dedicated to the Shu-Osher and Blastwave problems approximated on pyramidal cells which respectively involve a complex oscillatory solution and strong interactions between simple waves along the Ox direction. We address in section 3.3.4.2 the effective numerical accuracy of the method with the isentropic vortex problem for which an exact smooth solution exists. Then in section 3.3.4.2, we assess the ability of the MOOD method to simulate complex realistic physics on a mesh of triangular and quadrangular prisms by carrying out the impact of a shock wave on a cylindrical cavity proposed in [27]. At last, we provide the MOOD method results for two genuinely three-dimensional test cases. First we compare the behavior and computational cost (CPU and memory storage) of the MOOD method with different degrees and detection processes by simulating the so-called explosion problem [78] using unstructured pyramidal meshes in section 3.3.4.2; Then in section 3.3.4.2, we consider the interaction of a shock wave with a quarter of cone on a mesh of 1.1 millions of tetrahedra with the 4^{th}-order MOOD method.

⋆ Sod and Lax shock tubes

The original Sod [73] and Lax [51] problems concern one-dimensional Riemann shock tubes whose solutions consist of a left-moving rarefaction fan, a right-moving contact discontinuity and a right-moving shock wave. In the three-dimensional context, we reproduce the expansion following the Ox axis setting initial condition invariant in y, z and we prescribe reflecting boundary conditions on the cylinder sides as in [31]. The domain is filled with an ideal gas with $\gamma = 1.4$ and the discontinuity is located in $x = 0.5$ at $t = 0$. The initial density/velocity/pressure values and final time t_{final} are given by

– Sod: $(\rho, u, p)_L = (1.0, 0.0, 1.0)$ and $(\rho, u, p)_R = (0.125, 0.0, 0.1)$, $t_{\text{final}} = 0.2$,
– Lax: $(\rho, u, p)_L = (0.445, 0.698, 3.528)$ and $(\rho, u, p)_R = (0.5, 0.0, 0.571)$, $t_{\text{final}} = 0.13$.

The computational domain we consider is a cylinder of unit length and radius $R = 0.025$ with $0x$ line as symmetry axis which is paved with 7517 unstructured tetrahedra as shown in figure 3.39.

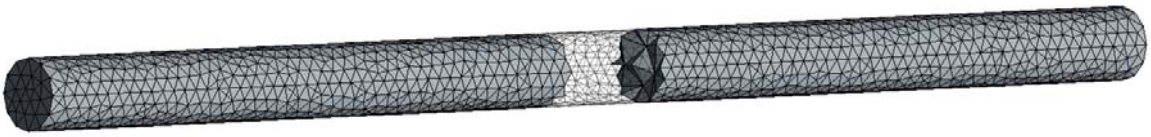


Figure 3.39: Mesh constituted of 7517 tetrahedra used for the Sod and Lax problems. Some cells are drawn non-opaque to see some interior tetrahedra.

We display in Figure 3.40 the numerical approximations of the density computed with the MOOD-$\mathbb{P}_3$ method using the [PAD $\rightarrow$ DMP $\rightarrow u2$] detection process and the exact solution (red

line). In order to provide a clear and relevant representation of the solution along the Ox axis, we slice the whole cylinder in 100 uniform cylinders (since the average characteristic length is 10^{-2}) and plot the average of the solution on each of them. As expected the MOOD-$\mathbb{P}_3$ method provides a very good approximation of the solution and maintains sharp discontinuities. In particular, we underline the very few numbers of points in the contact discontinuity.

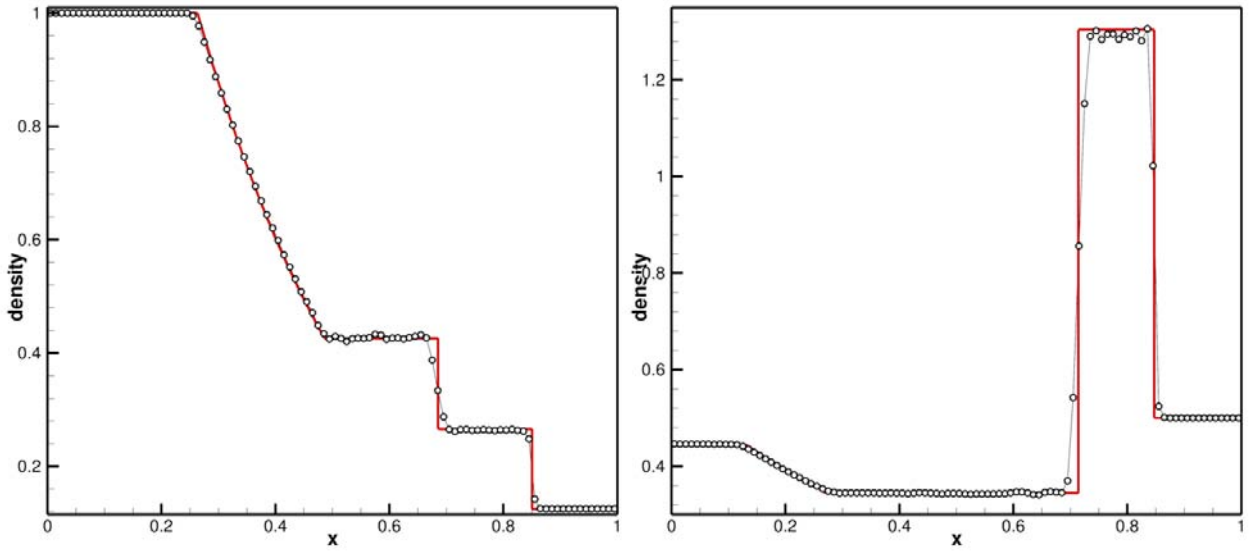


Figure 3.40: The MOOD-$\mathbb{P}_3$ density results are displayed for the Sod (left) and Lax (right) problems on tetrahedral mesh *vs* the exact solution (red line).

⋆ Shu-Osher and Blastwave problems

The Shu-Osher problem has been introduced in [70] to test the ability of a scheme to capture both small-scale smooth flow along with shock wave. The one-dimensional computational domain is $\Omega = [-5; 5]$ and the final time is $t_{\text{final}} = 1.8$. An initial x-directional shock wave located at $x = -0.4$ separates the domain into a left post-shock state $(\rho, u, p)_L = (3.857143, 2.629369, 10.333333)$ and a right state $(\rho, u, p)_R = (1 + 0.2\sin(5x), 0, 1.0)$. We consider a perfect gas with $\gamma = 1.4$. Reflecting boundary conditions are used to preserve the invariance following axis Oy, Oz except from the left boundary condition which is an inflow one.

The Blastwave problem has been introduced by Collela and Woodward in [91] to test the performance of numerical schemes on problems involving strong and thin shock structures. The initial conditions consist of two parallel planar flow discontinuities on domain $\Omega = [0, 1]$ separated by the planes $x_1 = 0.1$ and $x_2 = 0.9$. The density is unity on the whole domain and the gas is assumed initially at rest. The pressure is given by $p_L = 1000$ on the left, $p_C = 0.01$ in the center and $p_R = 100$ on the right. Reflecting boundary conditions are prescribed and the final time is $t_{\text{final}} = 0.038$.

We consider a 21600 regular pyramids mesh (see Figure 3.42-top right for a pattern example) obtained from a $400 - 3 - 3$ regular hexahedral mesh for which each cell is split into six pyramids. The original hexahedral mesh is built by setting $\Delta x = \Delta y = \Delta z$ with $\Delta x = 0.075$ for Shu-Osher problem and $\Delta x = 0.0075$ for the Blastwave problem. Since there is no exact solution for both tests we have computed reference solutions using a first-order finite volume scheme with very fine meshes. As in the previous simulations, the solutions are plotted following the Ox direction considering an underlying 400 points uniform one-dimensional mesh and circles in Figure 3.41 represent the mean density on three-dimensional slices of thickness Δx.

Density approximations obtained with the MOOD-$\mathbb{P}_3$ are presented in Figure 3.41 and compared to the reference solution (red line). For the Shu-Osher problem (left) we report that the [PAD $\rightarrow$ DMP $\rightarrow u2$] detection criteria does not over-smooth the oscillatory solution and accurately capture the high-frequencies waves. On the other hand, for the Blastwave problem (right) we observe sharp contact discontinuities and shock waves are well-preserved. No spurious oscillations are generated and the central structure of the solution is very well approximated.

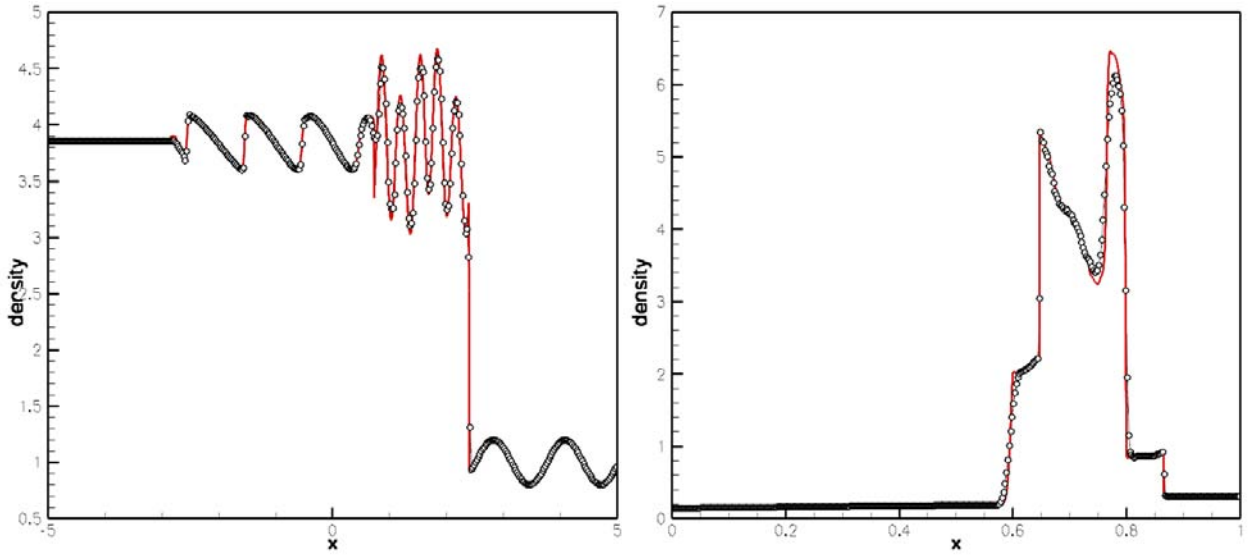


Figure 3.41: Results for the Shu-Osher (left) and Blastwave (right) problems on pyramids. MOOD-$\mathbb{P}_3$ density results are displayed on the left and right columns respectively *vs* the reference solution (red line).

$\star$ Isentropic vortex

The isentropic vortex problem was initially introduced for the two-dimensional space [68, 93] to test the accuracy of numerical methods since the exact solution is smooth and has an analytical expression. We simply extend the original problem for the three-dimensional situation taking the two-dimensional solution invariant following Oz. Let us consider the computational domain $\Omega = [-5, 5] - [-5, 5] - [0, z_{max}]$ and an ambient flow characterized with $\rho_\infty = 1.0$, $u_\infty = 1.0$, $v_\infty = 1.0$, $w_\infty = 1.0$, $p_\infty = 1.0$, with a normalized ambient temperature $T^*_\infty = 1.0$ computed with the perfect gas equation of state and $\gamma = 1.4$.
A z-invariant vortex is centered on the axis line $\mathbf{x}_{\text{vortex}} = (x_{\text{vortex}}, y_{\text{vortex}}, z) = (0, 0, z)$ $z \in \mathbb{R}$

and supplemented to the ambient gas at the initial time $t = 0$ with the following conditions $u = u_\infty + \delta u$, $v = v_\infty + \delta v$, $T^* = T^*_\infty + \delta T^*$ where

$$\delta u = -y' \frac{\beta}{2\pi} \exp\left(\frac{1-r^2}{2}\right), \quad \delta v = x' \frac{\beta}{2\pi} \exp\left(\frac{1-r^2}{2}\right), \quad \delta T^* = -\frac{(\gamma-1)\beta}{8\gamma\pi^2} \exp\left(1-r^2\right],$$

with $r = \sqrt{x'^2 + y'^2}$ and $x' = x - x_{\text{vortex}}, y' = y - y_{\text{vortex}}$. The vortex strength is given by $\beta = 5.0$ and the initial density follows relation

$$\rho = \rho_\infty \left(\frac{T^*}{T^*_\infty}\right)^{\frac{1}{\gamma-1}} = \left(1 - \frac{(\gamma-1)\beta}{8\gamma\pi^2} \exp\left(1-r^2\right]\right)^{\frac{1}{\gamma-1}}. \tag{3.53}$$

The domain is paved either with $N-N-4$ hexahedra , $N = 20, 40, 60, 80, 120$ or with $N-N-24$ pyramids (each hexahedron from the previous mesh is split into 6 pyramids, see Figure 3.42). To reduce the computational effort, only four cells are considered in the z-direction and z_{max} is taken such that $\Delta x = \Delta y = \Delta z$, that is to say $z_{max} = 4\Delta x = 40/N$. The minimal/maximal number of cells is 1600/57600 hexahedra and 9600/153600 pyramids. We prescribe periodic boundary conditions everywhere.

In Figure 3.42 we display the convergence curves for the L^1 and L^∞ errors on the density approximations for MOOD-$\mathbb{P}_2$, MOOD-$\mathbb{P}_3$, MOOD-$\mathbb{P}_5$ methods, while we provide in Table 3.18 the corresponding errors and convergence rates. We report effective orders corresponding to the expected optimal rates of convergence for both types of meshes and underline the MOOD method capacity to provide effective high-order of accuracy on a smooth but non-trivial solution for the three-dimensional Euler system.

⋆ Impact of a shock wave on a cylindrical cavity

Based on the experiment proposed in [72], we have introduced this test case in [27] for the two-dimensional case. We here extend it to 3D by invariance along the z-direction. It consists in a planar shock wave which impacts a cylindrical cavity creating complex structures and instabilities. The original purpose of this stringent numerical test is to prove the ability of the MOOD method to capture physics in realistic conditions. In this paper it moreover assesses the capacity of the MOOD method to deal with mixed triangular and quadrangular prisms since the mesh is obtained by extrusion (only two layers) along the Oz axis of a 2D mesh containing 101127 cells (triangles and quadrangles). We moreover point out that important differences between cell sizes are present in the domain, since the largest characteristic length is 0.008 and the smallest one is 0.00015. At last, we run the simulation on the lower half part of the domain but plot a full domain using a symmetry argument. Details of the mesh are provided in Figure 3.43.

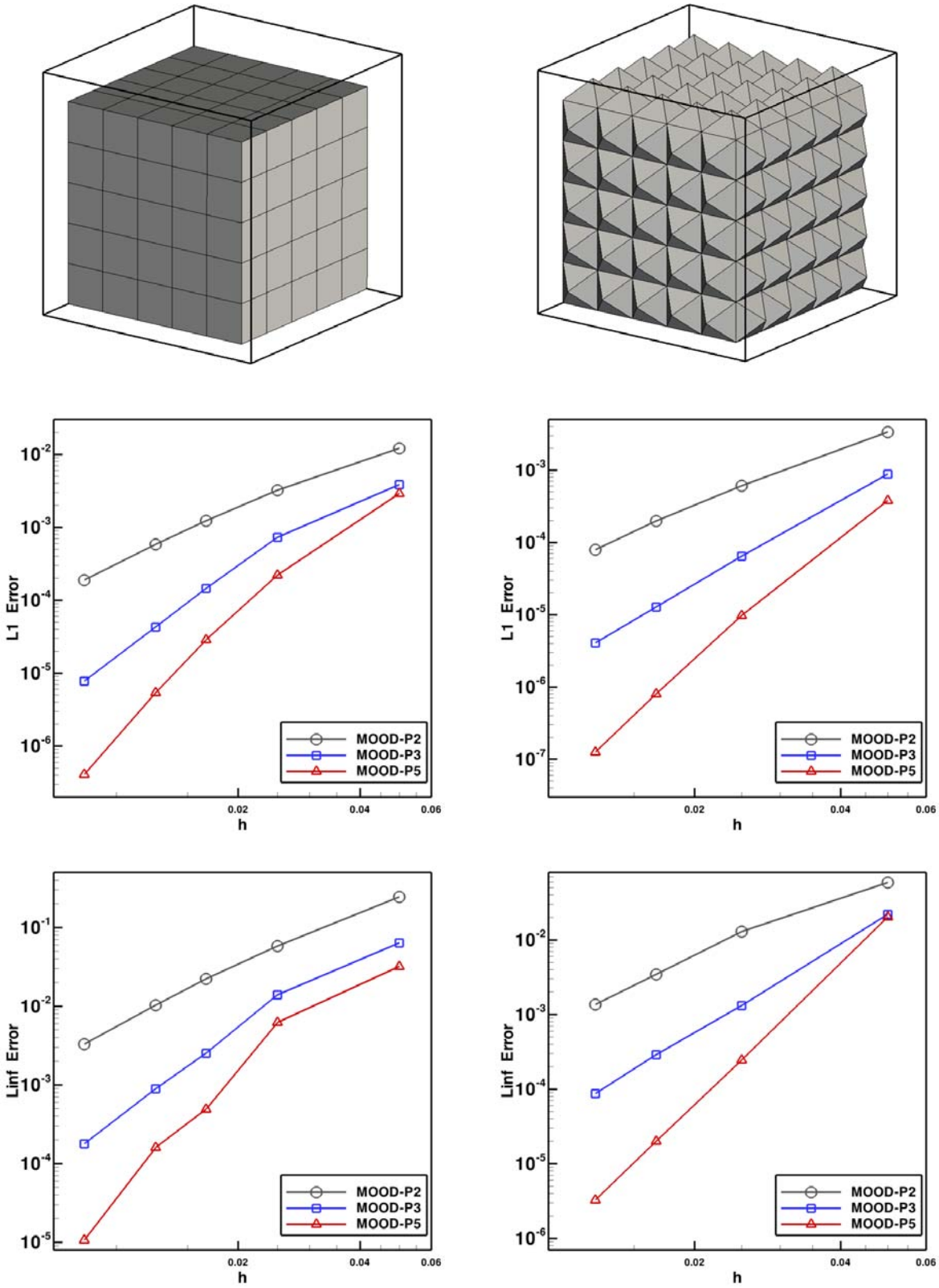


Figure 3.42: Isentropic vortex in motion: convergence curves for L^1 (middle) and L^∞ (bottom) errors for series of hexahedral (left) and pyramidal (right) meshes. Examples of such meshes are given on top line.

MOOD on hexahedra								
Deg.	Cell nb	L^1 error		L^2 error		L^∞ error	
$\mathbb{P}_2$	$20-20-4$	1.2149125472e-2	—	3.1900165943e-2	—	2.4441600308e-1	—
	$40-40-4$	3.2245099895e-3	1.91	8.0410260168e-3	1.98	5.8071095773e-2	2.07
	$60-60-4$	1.2274201731e-3	2.38	3.0952199533e-3	2.35	2.2241472168e-2	2.37
	$80-80-4$	5.8449248920e-4	2.57	1.4891720330e-3	2.54	1.0333244490e-2	2.66
	$120-120-4$	1.8870676632e-4	2.78	4.8726522430e-4	2.75	3.2966795901e-3	2.82
Expected order			3		3		3
$\mathbb{P}_3$	$20-20-4$	3.8426301161e-3	—	9.2866634542e-3	—	6.3860302401e-2	—
	$40-40-4$	7.2909293814e-4	2.39	1.6463735019e-3	2.49	1.3928173243e-2	2.19
	$60-60-4$	1.4537313954e-4	3.97	3.4725838178e-4	3.84	2.5316808689e-3	4.20
	$80-80-4$	4.3014601762e-5	4.23	1.1403422006e-4	3.87	8.9234157884e-4	3.62
	$120-120-4$	7.7653485653e-6	4.22	2.0827671718e-5	4.19	1.7793186459e-4	3.98
Expected order			4		4		4
$\mathbb{P}_5$	$20-20-4$	2.8991068920e-3	—	4.8543664172e-3	—	3.2038381504e-2	—
	$40-50-4$	2.2151699683e-4	3.71	5.5851141683e-4	3.12	6.2194475329e-3	2.36
	$60-60-4$	2.8610132561e-5	5.04	7.5286576723e-5	4.94	4.9068468256e-4	6.26
	$80-80-4$	5.4168534310e-6	5.78	1.5519206048e-5	5.49	1.5955744462e-4	3.90
	$120-120-4$	4.0840597698e-7	6.38	1.1795674119e-6	6.36	1.0709587465e-5	6.66
Expected order			6		6		6
MOOD on pyramids							
Deg.	Cell nb	L^1 error		L^2 error		L^∞ error	
$\mathbb{P}_2$	$20-20-24$	3.3660908651e-3	—	8.2020368268e-3	—	5.8966752971e-2	—
	$40-40-24$	6.0800306087e-4	2.47	1.4780372369e-3	2.47	1.2917288297e-2	2.19
	$60-60-24$	1.9831385885e-4	2.76	5.0256415975e-4	2.66	3.4489695638e-3	3.25
	$80-80-24$	7.9096059248e-5	3.19	2.0028509695e-4	3.19	1.3642153624e-3	3.22
Expected order			3		3		3
$\mathbb{P}_3$	$20-20-24$	8.8005733635e-4	—	2.0405839361e-3	—	2.2060839273e-2	—
	$40-40-24$	6.4460987694e-5	3.77	1.4763173293e-4	3.78	1.3204082077e-3	4.06
	$60-60-24$	1.2809782719e-5	3.98	2.9223354775e-5	3.99	2.8960192576e-4	3.74
	$80-80-24$	4.0713141263e-6	3.98	9.3121356054e-6	3.97	8.6899534957e-5	4.18
Expected order			4		4		4
$\mathbb{P}_5$	$20-20-24$	3.7944742185e-4	—	9.8016940506e-4	—	2.0273963181e-2	—
	$40-40-24$	9.7451977113e-6	5.28	2.4664732108e-5	5.31	2.4540502338e-4	6.36
	$60-60-24$	8.0304771455e-7	6.15	2.0569058735e-6	6.12	2.0022941712e-5	6.18
	$80-80-24$	1.2520658320e-7	6.46	3.1436119294e-7	6.53	3.2667224251e-6	6.30
Expected order			6		6		6

Table 3.18: L^1, L^2 and L^∞ errors and convergence rates for the isentropic vortex problem with the MOOD-$\mathbb{P}_2$, MOOD-$\mathbb{P}_3$ and MOOD-$\mathbb{P}_5$ methods. Top lines: hexahedral meshes. Bottom lines: pyramidal meshes.

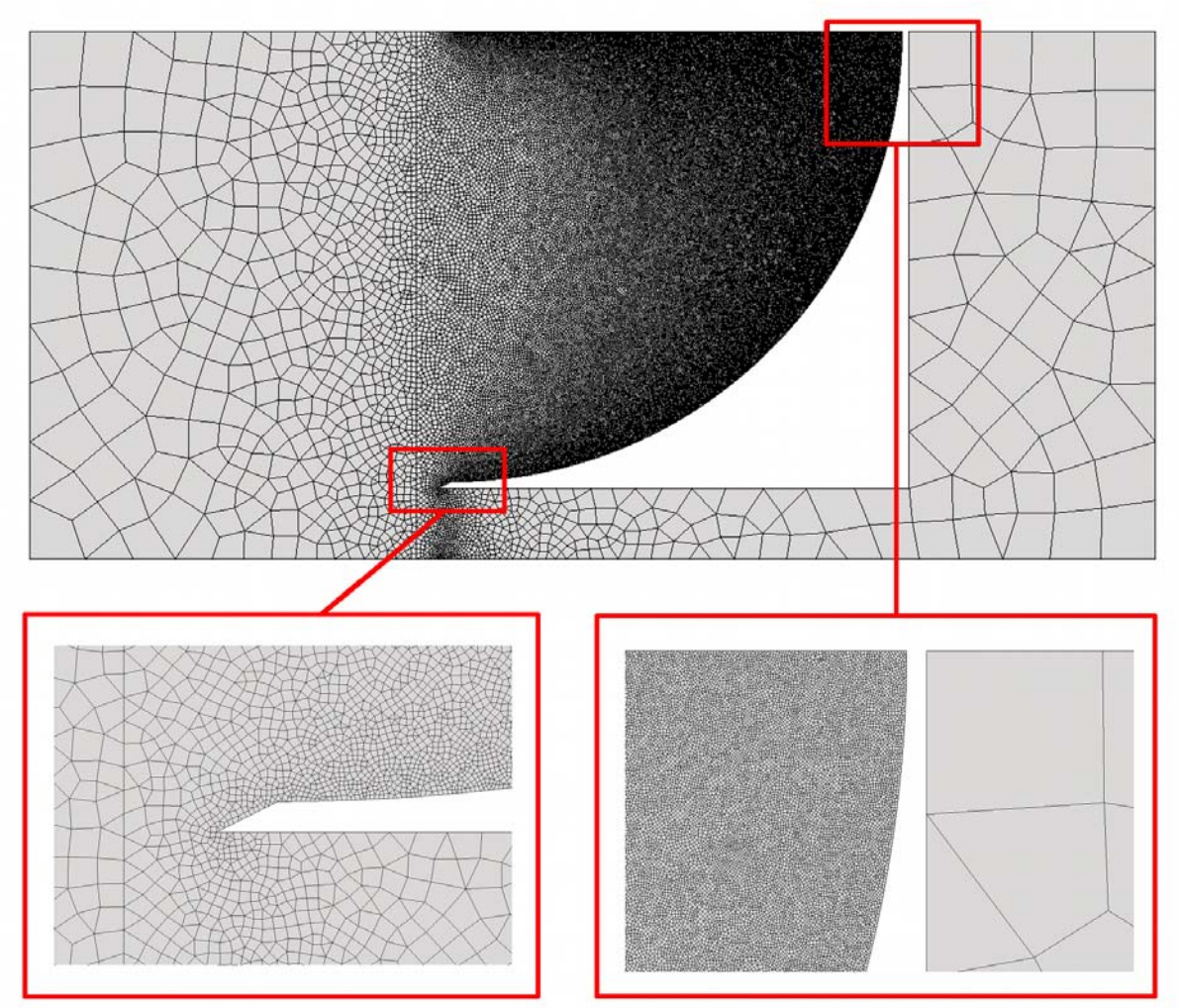


Figure 3.43: Impact of a shock on a cylindrical cavity: details of the mesh containing 202254 triangular and quadrangular prisms.

The detailed configuration and boundary conditions are provided in [27], and we recall that we consider the case of a nominal incident shock Mach number of 1.33 in ambient air (with $\gamma = 1.4$) at 0.95 bar pressure and that the variables initialization consists in the pre-shock values $(\rho, u, v, w, p) = (1.1175, 0.0, 0.0, 0.0, 95000.0)$ and the post-shock ones $(\rho, u, v, w, p) = (1.7522, 166.3435, 0.0, 0.0, 180219.75)$ leading to conditions of [72] at temperature $T = 296.15K$.
In Figure 3.44, we plot the magnitude of the density gradient computed with the MOOD $\mathbb{P}_2$ method equipped with the [PAD $\rightarrow$ DMP $\rightarrow u2$] detection process at different times of the simulation in order to give an overview of the physical phenomena. We emphasize that the instabilities along the cylindrical wall are very well captured. Finally in Figure 3.45, we provide a zoom of the final solution on the created instabilities which perfectly match the experimental results of [72].

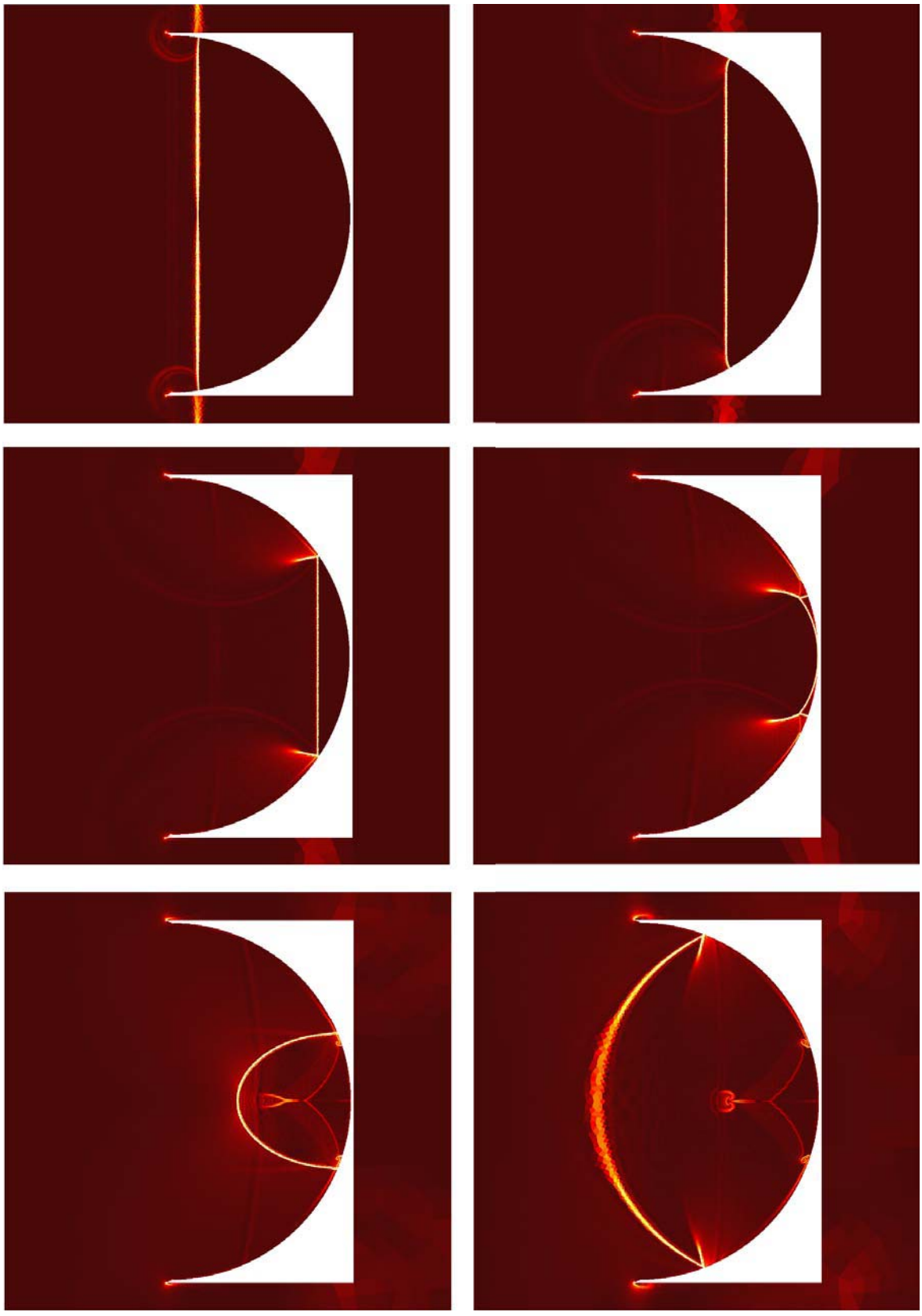


Figure 3.44: Impact of a shock on a cylindrical cavity: magnitude of the density gradient at different times from left to right and top to bottom.

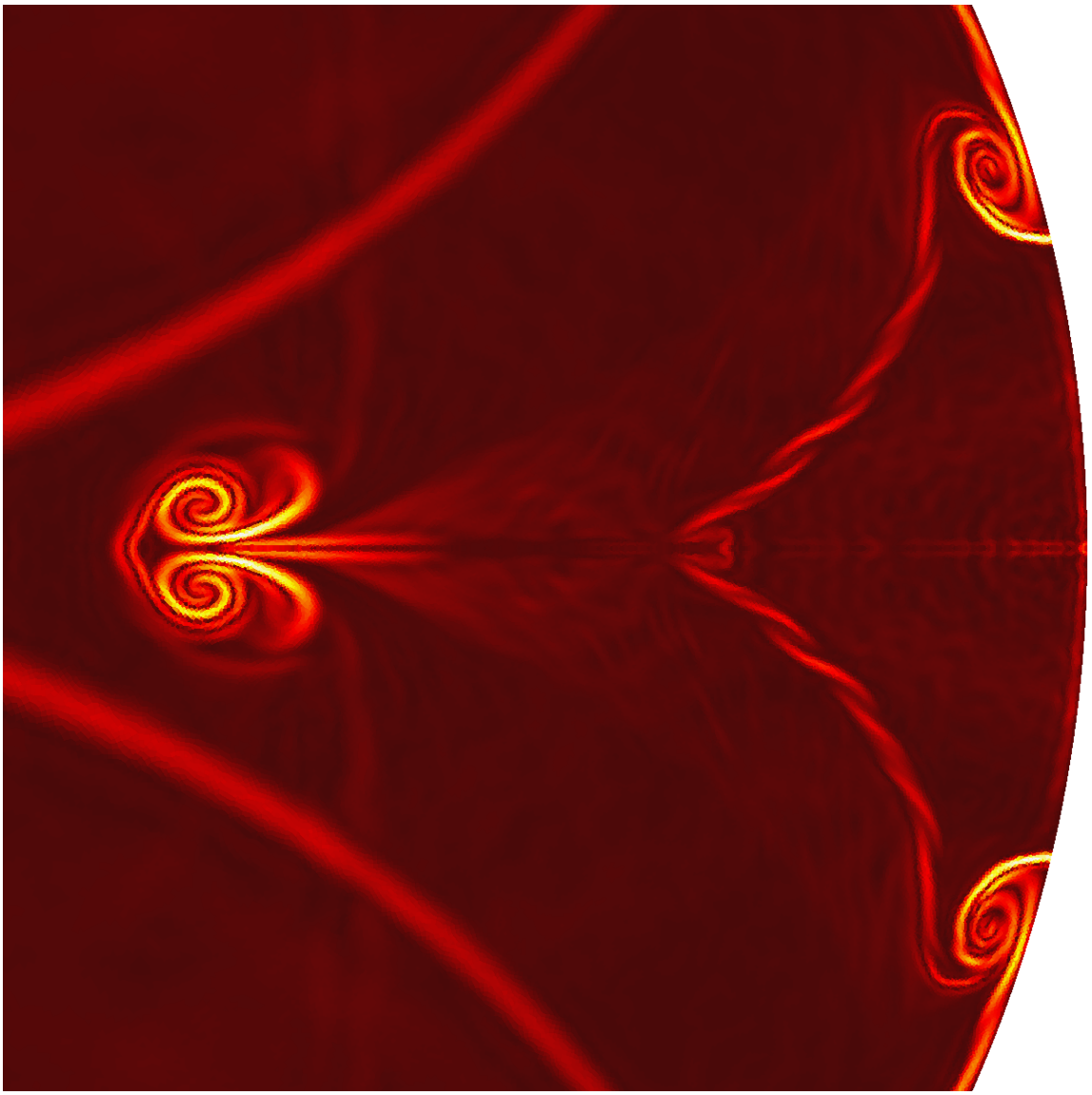


Figure 3.45: Impact of a shock on a cylindrical cavity: Zoom on the created instabilities at final time.

⋆ The explosion problem

We consider the so-called explosion problem [78] given by a gas initially at rest in the unit cube where a quarter of the ball of radius $r_c = 0.4$ centered at the origin has a density $\rho_b = 1.0$, a pressure $p_b = 1.0$ whereas the exterior is characterized by $\rho_e = 0.125$, $p_e = 0.1$. The domain is partitioned into 20^3 hexahedral cells for which each hexahedron is split into 6 pyramids leading to a mesh of 48000 pyramids. Simulations are carried out till the final time $t_{\text{final}} = 0.25$. A reference solution has been computed with a two-dimensional cylindrical staggered numerical Lagrangian scheme [55].

We report in Figure 3.46 the density approximations in function of the radius for a classical MUSCL scheme [62], the MOOD-$\mathbb{P}_2$, MOOD-$\mathbb{P}_3$, MOOD-$\mathbb{P}_5$ methods equipped with the [PAD $\rightarrow$ DMP $\rightarrow u2$] detection and the reference solution. Note that we use the same type of representation than for the previous test cases by slicing the radius in 100 uniform cells.

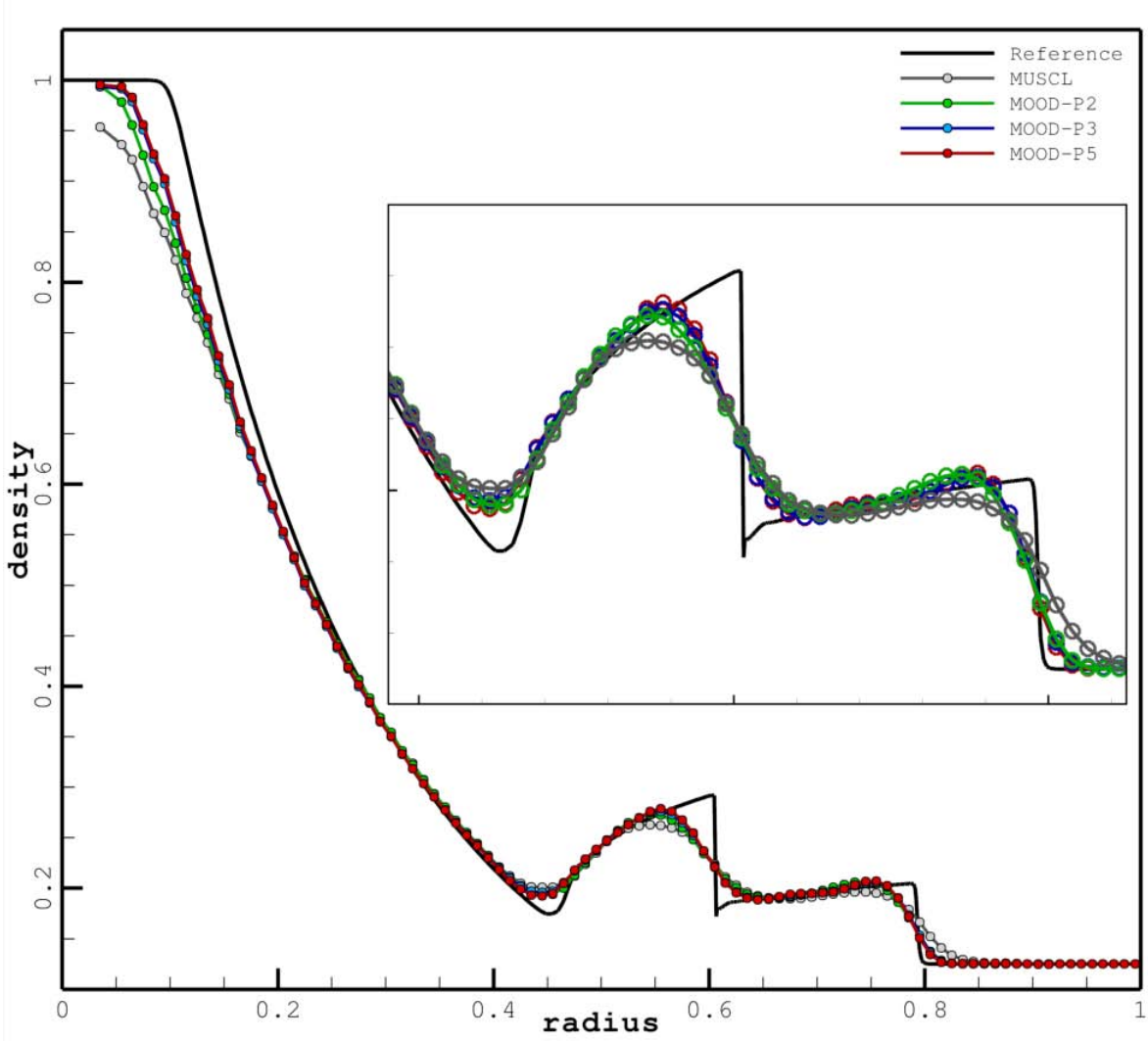


Figure 3.46: Density results for the explosion problem in 3D. Comparison between a classical MUSCL method and the MOOD-$\mathbb{P}_2$, MOOD-$\mathbb{P}_3$ and MOOD-$\mathbb{P}_5$ methods with [PAD $\rightarrow$ DMP $\rightarrow u2$] detection process on tetrahedral mesh. The straight line corresponds to the reference solution.

The solution shape is well reproduced by all methods and the higher the polynomial degree is the sharper the contact discontinuity and the shock wave are. In a three-dimensional context with discontinuous solutions, the MOOD-$\mathbb{P}_3$ method seems to be the right balance between accuracy and cost. The slight improvement gained by the MOOD-$\mathbb{P}_5$ compared to MOOD-$\mathbb{P}_3$ may not justify the computational over-cost (see further). We also notice that the head of the rarefaction wave is badly resolved by the MUSCL method whereas the MOOD-$\mathbb{P}_2$ and especially the MOOD-$\mathbb{P}_3$ method give accurate approximations.

To compare the different detection strategies, we present in Figure 3.47 the final solutions obtained by the MOOD method with the PAD alone and the [PAD $\rightarrow$ DMP $\rightarrow u2$] detection processes using $\mathbb{P}_5$ polynomial reconstructions.

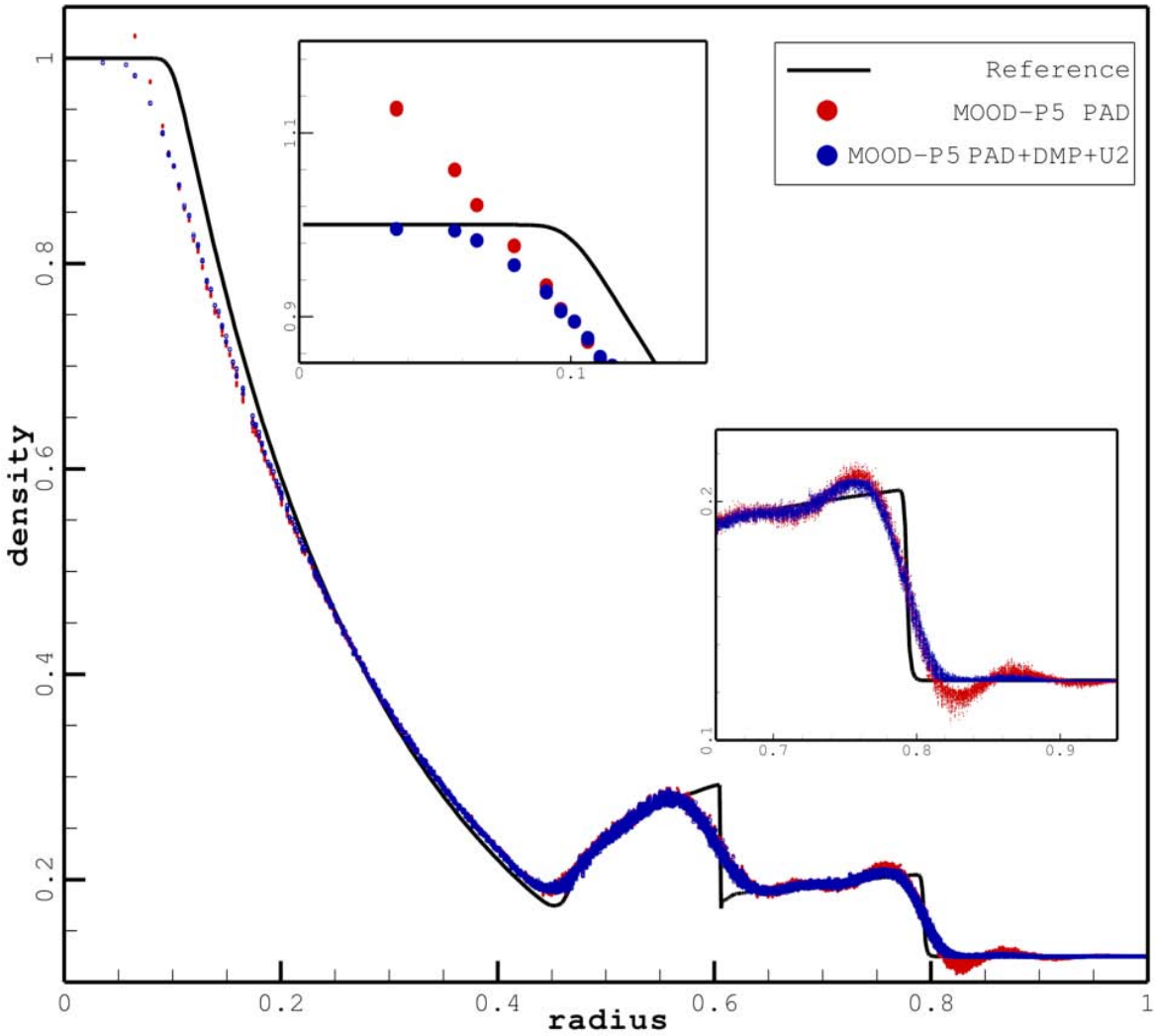


Figure 3.47: Density results for the explosion problem in 3D obtained with the MOOD-$\mathbb{P}_5$ method. Comparison between the PAD alone and [PAD $\rightarrow$ DMP $\rightarrow u2$] detection process. The straight line corresponds to the reference solution and the symbols represent mean values of all cells.

Note that contrary to previous figures, we plot the density values for all cells by associat-

ing them with the radius corresponding to the cell centroid. As expected, the PAD detection process does not damp spurious oscillations close to the shock wave (see the zoom panel) and extra oscillations are also visible on the head of the rarefaction. We recall that the numerical approximation using the PAD detection process is the most accurate one on smooth solutions since only the physical admissibility of the solution is required so that few numerical diffusion is produced. On the opposite, the [PAD $\rightarrow$ DMP $\rightarrow u2$] detection process damps the oscillations close to the shock and to the head of rarefaction but also maintain a very good accuracy with a slight non monotonic behavior.

Finally we provide in Tables 3.19 the computational cost of the MOOD method for this test case in this particular configuration when running on a single core of the three following machines (using -O3 flag for gfortran compiler):

M1: server with two Intel Xeon E5335 (4 cores) @ 2.00Ghz, 8MB of L2 Cache, 16GB of RAM

M2: laptop with Intel Core2Duo P7550 (2 cores) @ 2.26GHz, 3MB of L2 Cache, 8GB of RAM

M3: desktop with Intel Core i5 2500 (4 cores) @ 3.30GHz, 6MB of L2 Cache, 8GB of RAM

Note that the same three machines have been used in [27] to assess the computational cost of the MOOD method for two-dimensional geometries.

MOOD with [**PAD** $\rightarrow$ **DMP** $\rightarrow u2$]	**Machine 1** Intel Xeon E5335 @ 2.00Ghz	**Machine 2** Intel Core2Duo P7550 @ 2.26GHz	**Machine 3** Intel Core i5 2500 @ 3.30GHz	**Memory storage**
MOOD-$\mathbb{P}_2$	66μs/it./cell	57μs/it./cell	30μs/it./cell	0.4 GB
MOOD-$\mathbb{P}_3$	163μs/it./cell	136μs/it./cell	69μs/it./cell	0.8 GB
MOOD-$\mathbb{P}_5$	439μs/it./cell	385μs/it./cell	185μs/it./cell	3.0 GB

Table 3.19: CPU time in microseconds per iteration per cell and memory storage in Gigabytes for the MOOD-$\mathbb{P}_k$ methods ($k = 2, 3, 5$) with the [PAD $\rightarrow$ DMP $\rightarrow u2$] detection process on three different computers.

We first observe that the memory storage doubles when the polynomial degree is increased by one: 0.4 for $\mathbb{P}_2$, 0.8 for $\mathbb{P}_3$, 1.6 for $\mathbb{P}_4$ (not presented in the table) and 3.0 for $\mathbb{P}_5$. Notice that the memory consumption is very low since only two reconstruction pseudoinverse matrices (for $\mathbb{P}_2$ and $\mathbb{P}_{\mathsf{d}_{max}}$) per cell are effectively stored. The CPU cost increases by a factor about 2.4 from $\mathbb{P}_2$ to $\mathbb{P}_3$ and about 2.7 from $\mathbb{P}_3$ to $\mathbb{P}_5$.

By extrapolation of these results we estimate the cost of the MOOD method for larger meshes. As instance for one million cells mesh and 1000 time steps the method cost should be:

- MOOD-$\mathbb{P}_2$ is 66000 seconds on M1, that is to say $\sim$ 18 hours ($\sim$ 16 hours on M2 and $\sim$ 8.3 hours on M3) with about 8 Gb of memory storage,
- MOOD-$\mathbb{P}_3$ is 163000 seconds, $\sim$ 2 days on M1 ($\sim$ 1.5 day and $\sim$ 19 hours on M2 and M3) with about 16 Gb of memory storage,
- MOOD-$\mathbb{P}_5$ is 439000 seconds, $\sim$ 5 days on M1 ($\sim$ 4.5 days and $\sim$ 2.1 days on M2 and M3) with about 62 Gb of memory storage.

Consequently simulations with nowadays sequential computers with a one million cells mesh (assuming one thousand time steps) can be obtained for about one day of computation with MOOD-$\mathbb{P}_3$ method. The MOOD method is thus a very competitive very high-order finite volume method, and these results shall be improved by an efficient parallelization.

$\star$ Interaction of a shock wave with a quarter of cone

To conclude the numerical tests section, we run the test case named *interaction of a shock wave with a quarter of cone* with the 4^{th}-order MOOD-$\mathbb{P}_3$ method equipped with the [PAD$\rightarrow$ DMP$\rightarrow u2$] detection process. This 3D extension of the so-called *interaction of a shock wave with a wedge* has been proposed in [31] as instance.

The domain consists in a quarter on cylinder of radius $R = 2.25$ centered on the Ox axis which covers the interval $[-1.1; 3.0]$ in the x-direction. Note that three modifications have been made in comparison to [31] in order to reduce the computational cost: the test is run on a quarter of cylinder instead of a half one, the initial interface is placed at $x = -0.2$ instead of $x = -1.0$ and the domain covers in the x-direction the interval $[-1.1; 3.0]$ instead of $[-1.5; 3.0]$. Finally the mesh obtained by the free mesher Gmsh contains 1161854 tetrahedra in three refinement zones and exactly matches the initial interface, see top of Figure 3.49.

We recall that the circular cone under consideration is such that its length is 1, its tip and foot radii are 0.02 and 0.5 respectively while its tip is placed at the origin. Moreover wall boundary conditions are prescribed everywhere except from the top and bottom of the quarter of cylinder where the exact solution according to the Rankine-Hugoniot conditions is imposed. At last the initial pre- and post shock conditions are given by $(\rho, u, v, w, p) = (2.122, 0.0, 0.0, 0.0, 1.805)$ and $(\rho, u, v, w, p) = (1.4, 0.0, 0.0, 0.0, 1.0)$ respectively with $\gamma = 1.4$ and the final time is chosen such that it corresponds to the final time of [31].

In Figure 3.48, we propose numerical Schlieren-type images on the solution in the $Ox - Oy$ and $Ox - Oz$ planes. We remark that the symmetry is very well conserved since both images are almost identical and that all waves that are present in results of [31] are also resolved here although much less cells (more than 3.5 times less) are considered. This proves that the MOOD method performs very well on 3D unstructured meshes. Finally on bottom of Figure 3.49, we provide a 3D view for which isosurfaces have been chosen to represent the principal waves in the whole domain. It is thus clear that the method properly reproduces the cylindrical symmetry even on this fully unstructured 3D tetrahedral mesh.

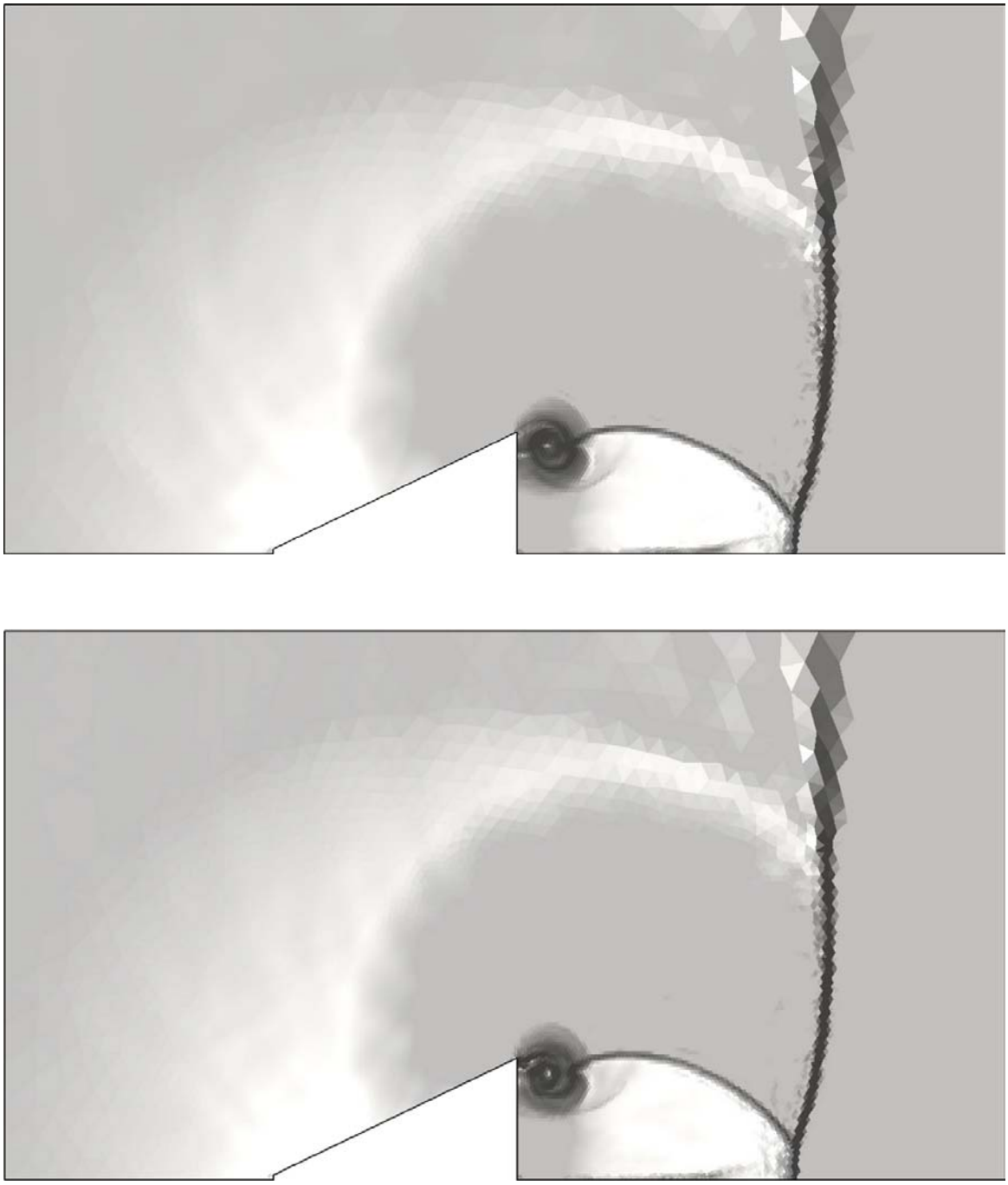


Figure 3.48: Interaction of a shock wave with a half cone: on top, numerical Schlieren-type image on the $Ox - Oy$ plane; on bottom, numerical Schlieren-type image on the $Ox - Oz$ plane.

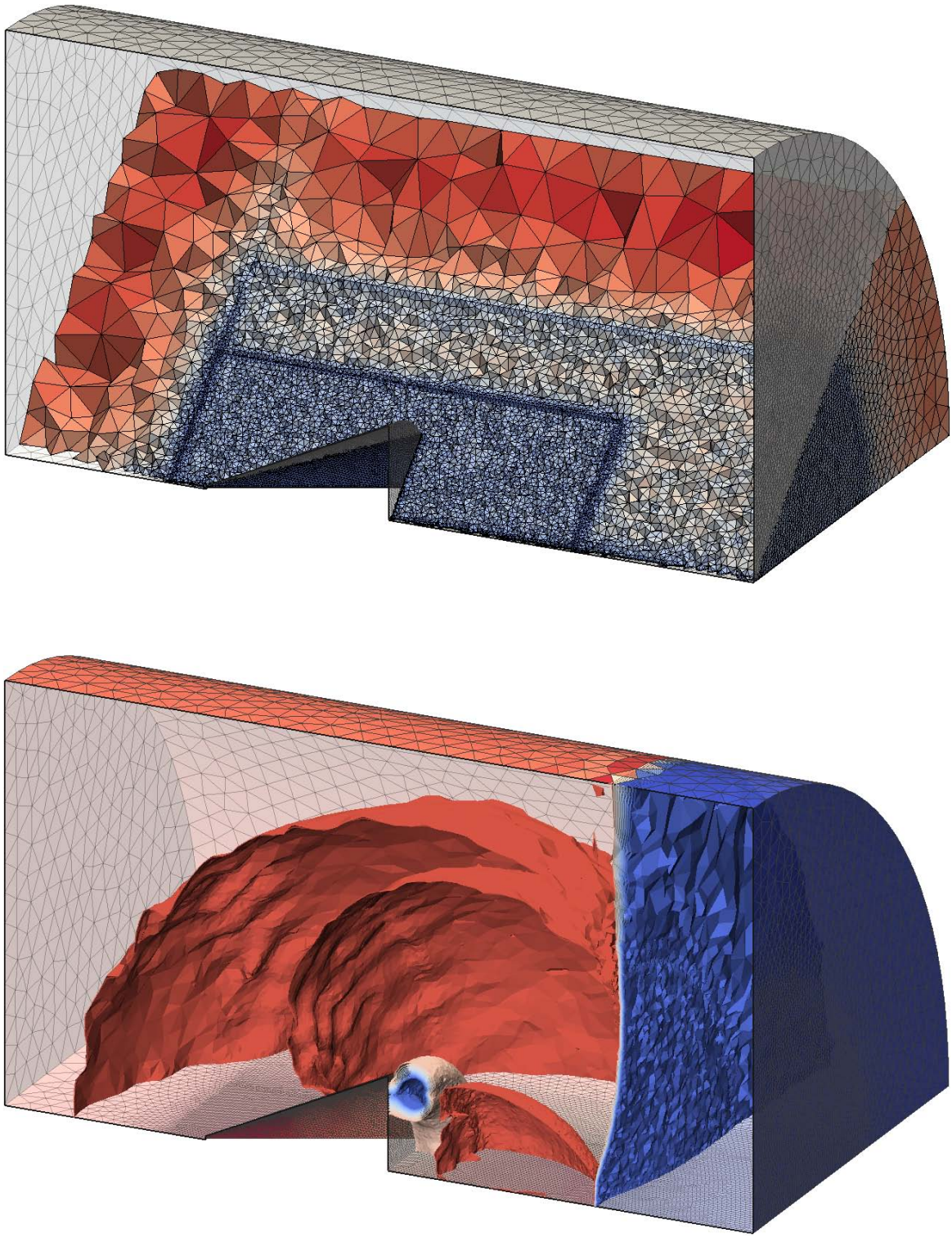


Figure 3.49: Interaction of a shock wave with a half cone: on top, view of the interior of the tetrahedral mesh with the different zones of refinement; on bottom, isosurfaces corresponding to the principal waves.

3.3.5 Conclusion

In this paper we have proposed the three-dimensional extension of the so-called MOOD method [18, 27]. The *Multi-dimensional Optimal Order Detection* expression refers to an original way of determining the optimal local polynomial degree to be used in the reconstruction step of a classical high-order unlimited scheme. To each cell corresponds a polynomial reconstruction for which we *a posteriori* determine the degree according to given criteria (positivity as instance) against which we test each candidate solution. The detection criteria is based on a relaxed version of the discrete maximum principle (DMP) associated with a so-called $u2$ detection procedure which analyses the numerical curvatures in the neighborhood of a DMP violating cell and determine if the underlying function is regular or not. In the latter case the polynomial degree in the associated cell is decremented and the solution is locally recomputed. We have detailed the numerical method for three-dimensional unstructured meshes and improved the detection criteria both for the advection equation and the Euler system. Moreover some optimizations for the three dimensional case have been provided to significantly improve the efficiency of the method.

The MOOD method has been implemented on several kinds of unstructured meshes with $\mathbb{P}_k$ polynomial reconstructions (k varying from 1 to 5). We have provided some sanity checks with simple configurations and performed more advanced full three-dimensional tests to assess the ability of the MOOD method to accurately capture waves on real unstructured meshes. For the scalar convection equation with a regular initial shape the method gives an effective high-order of accuracy corresponding to the optimal one and we have shown that spurious oscillations are damped when discontinuous profiles are convected. The results for unidirectional problems for the Euler system with three-dimensional unstructured meshes show that small scaled structures are captured while shock waves are resolved within few cells. For the isentropic vortex test case extended to the three-dimensional context with non-trivial exact solution, effective high-orders of accuracy are measured and optimal orders are reported for $\mathbb{P}_2, \mathbb{P}_3$ and $\mathbb{P}_5$ polynomials. We prove that the MOOD method is able to capture the realistic physics of the impact of a shock wave on a cylindrical cavity on a non trivial mesh made of a mix of triangular and quadrangular prisms. At last, the three-dimensional explosion problem has been carried out to show the improvement gained with the use of high-order MOOD methods and the slight numerical diffusion generated by the $u2$ detection process which enables to prevent spurious numerical oscillations from appearing. We have also provided the solution computed with the PAD detection process alone to support the intrinsic positivity-preserving property of the MOOD method and measures of the CPU cost to underline that the MOOD method is effective on nowadays personal computers. Finally the interaction of a shock wave on a quarter of cone with the 4^{th}-order MOOD method proves that the MOOD method provides a very good reproduction of the physics on a unstructured non-regular 3D mesh of 1.1 millions of tetrahedra.

In a near future, we plan to adapt the MOOD within an ADER technique to avoid the multiple time steps of the Runge-Kutta approach and overcome the third-order accuracy restriction. Furthermore although the MOOD method significantly reduces the necessary computational resources (CPU and memory storage), a parallelized version is of crucial importance to treat huge size simulations. Finally the application of the MOOD method to more complex physics (multi-material, multi-phase, etc.) is also an important challenge that has to be tackled.

Conclusion and Perspectives

Version française

Dans cette thèse, nous avons conçu et développé un nouveau type de méthodes Volumes Finis d'ordre très élevé pour la simulation des équations d'Euler régissant la dynamique des gaz compressibles non-visqueux, dénommé MOOD pour Multidimensional Optimal Order Detection. La principale nouveauté de notre approche est le traitement *a posteriori* des phénomènes parasites engendrés par l'approximation d'ordre élevé des zones non-régulières de la solution, en opposition avec les méthodes classiques qui utilisent des limitations *a priori*. Il en résulte que sur les zones régulières de la solution, le schéma non limité est utilisé sans aucune modification et que la limitation ne s'effectue que sur les parties non-régulières en recalculant localement la solution avec un schéma d'ordre moins élevé.

Nous avons défini un cadre et des outils adaptés à l'approche *a posteriori* et prouvé sur de nombreux cas tests numériques en dimension deux et trois que la méthode MOOD est non seulement viable mais surtout plus efficace que l'état de l'art. Le concept proposé n'étant pas intrinsèquement lié aux équations que nous avons considérées, tout semble donc favorable à l'extension de la méthode MOOD à d'autres classes de problèmes. Dans ce qui suit, nous proposons un ensemble de pistes à suivre dans le futur qui permettront de renforcer et populariser la méthode MOOD.

En premier lieu, nous présentons des évolutions possibles de la méthode dans le contexte dans lequel nous l'avons présentée dans cette thèse. Nous distinguons au moins trois possibilités intéressantes.

La première que nous avons évoquée dans la section 2.4 est la parallélisation de la méthode de manière à la rendre encore plus efficace et permettre ainsi de considérer des simulations à plusieurs dizaines de millions de mailles sur une machine de calcul *personnelle* (via OpenMP, GPGPU) et plus encore sur un supercalculateur (via MPI). Or la structure de la méthode MOOD semble adaptée au calcul parallèle puisque les éléments qui la constituent (reconstruction polynomiale, processus de détection) ne prennent en compte que des données locales sur un voisinage identique aux méthodes existantes. La parallélisation effective n'est donc qu'une question de temps et d'opportunités.

Une deuxième direction intéressante, évoquée en section 1.3.3, pour réduire le coût de la méthode est d'utiliser une méthode de type ADER à la place de la méthode RK3-TVD comme discrétisation en temps. D'une part cela diviserait par trois le nombre d'étapes pour obtenir

la solution mise à jour, et par conséquent diminuerait significativement le temps de calcul. D'autre part cela permettrait d'avoir une discrétisation en temps d'ordre aussi élevé qu'en espace. Rappelons toutefois que les méthodes ADER restent complexes et que par suite la simplicité du schéma proposé dans cette thèse serait réduite. Cependant dans l'optique d'un code de calcul performant sur une application précise, cette amélioration est particulièrement pertinente.

Enfin nous pensons qu'une méthode MOOD simplifiée à l'extrême et d'ordre limité à deux serait une passerelle concrète vers le monde industriel dans lequel simplicité, robustesse et rapidité sont des facteurs prédominants. En effet, la reconstruction à l'ordre deux est robuste et extrêmement performante même en dimension trois et nous avons montré dans [18, 27, 28] que numériquement la méthode MOOD-$\mathbb{P}1$ fournit de meilleurs résultats que la méthode MUSCL classique. De plus, puisque les paradigmes de la méthode MOOD sont indépendants de l'ordre, nous pourrions aisément proposer aux ingénieurs ayant assimilés ces paradigmes, une migration vers la version à l'ordre élevé.

La deuxième grande classe de perspectives importantes est celle de l'étude mathématique de la méthode et de ce qui la compose. La reconstruction polynomiale à partir des valeurs moyennes est un élément central des méthodes Volumes Finis d'ordre élevé. Pour autant sa compréhension mathématique est encore incomplète dans le cadre multidimensionnel non-structuré et l'ensemble de ses propriétés n'est pas établi. La plupart des caractéristiques associées à la reconstruction sont empiriques et relèvent souvent plus de la recette de cuisine que d'un processus mathématique rigoureux. Il semble pourtant nécessaire que cet élément de base soit bien compris pour d'une part assurer la robustesse, et d'autre part pouvoir réduire son coût, par exemple en diminuant le nombre de voisins nécessaires.

Ensuite la méthode MOOD devra être analysée précisément afin de mettre à jour ses propriétés mathématiques en tant que schéma numérique suivant les différents processus de détection utilisés. Par exemple, il serait intéressant de pouvoir démontrer que la méthode MOOD associée au processus de détection [DMP $\rightarrow u2$] pour la convection satisfait une propriété de Variation Totale Bornée (TVB), ou encore qu'elle assure que les inégalités d'entropie associées au système d'Euler sont vérifiées. On connaît toute la difficulté de démontrer ce genre de propriétés dans le cadre multidimensionnel non-structuré, mais on peut espérer que l'approche *a posteriori* fournisse de nouvelles techniques de démonstration.

La troisième évolution majeure de la méthode concerne bien entendu son application à d'autres classes de problèmes, et plusieurs travaux ont déjà été entrepris : la méthode MOOD a été étendue aux équations de Shallow-Water avec succès [19] et son application au problème de convection-diffusion est en cours. L'extension du concept MOOD ne pose pas de problèmes majeurs et ne requiert en fait que l'existence d'un schéma de base robuste garantissant une solution dans les situations critiques (singularités, discontinuités). Ensuite il faut développer des mécanismes d'optimisations du type de celles de la section 1.3.2 de manière à assurer l'efficacité de la méthode MOOD. Un des challenges majeurs à venir sera l'application de la méthode MOOD aux équations de Navier-Stokes incompressibles.

Nous réservons la conclusion de ces perspectives au point qui est peut-être le plus important

pour le futur. Nous avons montré dans cette thèse qu'une nouvelle approche à la limitation des schémas d'ordre très élevé est possible, et qu'elle se base sur un principe *a posteriori* qui aurait pu paraître rudimentaire et inefficace. Une nouvelle voie de réflexion est donc ouverte.

De plus, le recul pris sur ce travail permet de repenser la méthode MOOD présentée ici dans le cadre des Volumes Finis pour l'hydrodynamique comme un cas particulier d'un concept MOOD plus général. En effet, il suffit d'appréhender la méthode présentée comme un algorithme de sélection locale du schéma le plus approprié aux caractéristiques de la solution que l'on recherche. Il est alors clair que les seuls éléments importants sont : une liste de schémas ordonnée du plus précis au plus robuste et un ensemble de propriétés que l'on désire pour la solution. De ce fait, la méthode Volumes Finis, l'ordre du schéma, ou encore les équations que l'on considère ne sont pas des éléments importants de ce concept.

La méthode MOOD développée ici, pourrait donc devenir le concept MOOD qui s'appuyant sur une liste ordonnée de schémas numériques et un ensemble de propriétés à satisfaire permet de sélectionner localement le schéma le plus adapté aux équations que l'on résoud. Par exembple, une utilisation proche de ce que nous avons présenté serait de se baser sur une méthode de type Galerkin Discontinu (au lieu des Volumes Finis) qui contient intrinsèquement une représentation d'ordre élevée de la solution. La décrementation du degré n'agirait donc plus sur la reconstruction polynomiale mais directement sur la base locale dans laquelle la solution est représentée.

Pour conclure, nous pensons que la méthode MOOD possède l'ensemble des qualités requises pour trouver un écho favorable dans le monde du calcul scientifique tout autant que celui de l'industrie. Nous espèrons donc que les idées originales proposées dans ce travail de doctorat vont initier de nouvelles collaborations multi-disciplinaires.

English version

In this thesis, we have designed and developed a novel type of very high-order Finite Volume methods to simulate the Euler equations ruling the non-viscous compressible gas dynamics, named MOOD for Multidimensional Optimal Order Detection. The major novelty of our approach is the *a posteriori* treatment of spurious phenomena generated by the high-order approximation of the irregular parts of the solution, whereas classical methods use *a priori* limitations. As a result, the unlimited scheme is applied on the smooth part of the solution and limitation only acts in the vicinity of the singularities by locally recomputing the solution with a lower-order scheme.

We have defined a framework along with tools suitable for the *a posteriori* approach. Numerical tests for the two- and three-dimensional cases have been carried out and showed that the MOOD method is not only viable but more efficient than the state-of-the-art very high-order Finite Volume methods. The innovative concept is not intrinsically tied to the equations we consider, so that all conditions seem to be favorable to the extension of the MOOD method to other classes of problems. In the following, we propose a set of selected directions for future investigations which may reinforce and popularize the MOOD method.

Firstly, we present possible improvements of the MOOD method in the same context as this thesis. At least three interesting possibilities come up.

The first one was mentioned in section 2.4 and consists in the parallelization of the method to make it more efficient so that simulations using tens of millions of cells on a personal workstation (via OpenMP, GPGPU) and even more on supercomputers (via MPI) would be possible. Furthermore the MOOD method seems to be suitable for parallel computations since its constitutive elements (polynomial reconstruction, detection process) only deal with local informations on the same neighborhood as existing methods. Therefore an effective parallelization is a question of time and opportunities.

A second interesting direction, mentioned in section 1.3.3, to reduce the computational cost of the method is to use an ADER method as a substitute for the RK3-TVD as time discretization. On the one hand, the number of steps to get the time update of the solution would be divided by three so that it would significantly lower the computational cost. On the other hand it would provide a time discretization of arbitrary order. Let us recall that the ADER methods are still complex so that the simplicity of our scheme would be reduced; however from the perspective of a high-performance simulation code for a specific application, this improvement is particularly relevant.

Finally we think that an extremely simplified second-order MOOD method would be a practical gateway to the industrial world for which simplicity, robustness and speed are prevailing considerations. Indeed, the second-order reconstruction is robust and extremely efficient even in the three-dimensional case, and we have proved in [18, 27, 28] that the method MOOD-$\mathbb{P}1$ method numerically provides better results than the classical MUSCL one. Moreover, since the paradigms of the MOOD method are independent of the scheme order, we could easily propose a high-order update to the engineers who have assimilated these paradigms.

The second class of important perspectives concerns the mathematical analysis of the MOOD method and its constitutive elements. The polynomial reconstruction from mean values is a key process of very high-order Finite Volume methods. Its mathematical understanding is still incomplete for unstructured meshes of multidimensional geometries and all its properties are not set yet. Most of the characteristics of the reconstruction are empirical and is often closer to a recipe than to a rigorous mathematical process. A fine understanding of this basic element is also necessary to ensure the robustness and to be able to lower its cost, for instance in reducing the stencil size.

Then the MOOD method will have to be accurately analyzed in order to demonstrate its mathematical properties as a numerical scheme in regard to the detection process used. For instance, it would be interesting to prove that the MOOD method using the [DMP+$u2$] detection process for the scalar convection equation fulfills a Total Variation Bounded (TVB) property, or else that the entropy inequalities associated to the Euler system hold. Such issues are usually difficult to demonstrate in an unstructured multidimensional context, but we may expect that the *a posteriori* approach provides new techniques of demonstration.

The third main evolution of the MOOD method naturally concerns its application to other classes of problems, and several investigations have already been initiated: the MOOD method has been extended to the Shallow-Water equations successfully [19] and its application to the

convection-diffusion problem is an on-going study. No major problem arises in the extension of the MOOD concept. It only requires the existence of a basic robust scheme which guarantees a solution in critical situations (*e.g.* singularities, discontinuities). Then optimization mechanisms such as in section 1.3.2 have to be developed to ensure the efficiency of the MOOD method. One of the challenges to come is the application of the MOOD method to the incompressible Navier-Stokes equations.

The conclusion of these perspectives is dedicated to the possibly most important point for the future of the method. We have shown in this study that a novel approach to the limitation of very high-order schemes is feasible and driven by an *a posteriori* principle which could have been judged unrefined and inefficient. A new way of thinking is thus wide open.
Furthermore, standing back from this work, we find out that the MOOD method we have presented as a Finite Volume method for hydrodynamics may be considered as a particular case of a more general MOOD concept. Actually, the MOOD technique may be generalized as an algorithm of local selection of the optimal (*i.e.* most appropriate) scheme according to the solution characteristics we seek. It is then clear that the truly important elements are: a list of schemes ordered from the most accurate to the most robust and a set of properties desired for the solution. As a matter of fact, the Finite Volume method, the scheme order, or even the equations we consider are not important elements of this concept.
Consequently, the MOOD method may become a MOOD concept which, considering an ordered list of numerical schemes and a set of properties to satisfy, locally selects the most appropriate scheme to the equations we solve. As instance, this could be considered to develop a MOOD strategy using Discontinuous Galerkin method (instead of Finite Volume) which intrinsically contains a high-order representation of the solution. The decrementing process would then act on the local functions basis on which solution is projected, instead of the degree of the polynomial reconstruction.

To conclude, we believe that the MOOD method has the set of required properties to expect a positive response from the scientific computation community and the industrial world. We thus hope that the original ideas proposed in this doctoral dissertation will initiate new multidisciplinary collaborations.

Bibliography

[1] R. Abgrall On Essentially Non-oscillatory Schemes on Unstructured Meshes: Analysis and Implementation, J. Comput. Phys. 114 (1994) 45–58. (Cited pages 2, 5, 18, 24, 55, 86, and 92.)

[2] R. Abgrall, Essentially non-oscillatory Residual Distribution schemes for hyperbolic problems, J. Comput. Phys. 214 (2006) 773–808. (Cited pages 1, 4, and 87.)

[3] R. Abgrall, T. Barth, Residual distribution schemes for conservation laws via adaptive quadrature, SIAM J. Sci. Comput. 24 (2002) 732–769. (Cited pages 1 and 4.)

[4] R. Artebrant, H. J. Schroll, Limiter-free third order logarithmic reconstruction, SIAM J. Sci. Comput., 28 (2006) 359–381. (Cited page 15.)

[5] R. Artebrant, H. J. Schroll, Limiter-free third order logarithmic reconstruction, J. Sci. Comput., 30 (2006) 193–221. (Cited page 15.)

[6] D. S. Balsara, C.-W. Shu, Monotonicity Preserving Weighted Essentially Non-oscillatory Schemes with Increasingly High Order of Accuracy, J. Comput. Phys. 160 (2000) 405– 452. (Cited pages 2 and 5.)

[7] T. J. Barth, Numerical methods for conservation laws on structured and unstructured meshes, VKI March 2003 Lectures Series. (Cited pages 55, 59, and 86.)

[8] T. J. Barth, D. C. Jespersen, The design and application of upwind schemes on unstructured meshes, AIAA Report 89-0366 (1989). (Cited pages 21, 55, and 59.)

[9] T. J. Barth, P. O. Fredrickson, Higher-Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction, AIAA Aerospace Sciences Meeting, 28th Reno (1990) 90–0013. (Cited pages 15, 16, and 92.)

[10] F. Bassi, S. Rebay, A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible NavierStokes Equations, J. Comput. Phys. 131 (1997) 267–279. (Cited pages 1 and 4.)

[11] C. Berthon, Robustness of MUSCL Schemes for 2D unstructured meshes, J. Comput. Phys. 218 (2006) 495–509. (Cited page 22.)

[12] M. Bôcher, Introduction to the theory of Fourier's series, Annals of Math. 7 (1906) 81–152 (Cited pages 2 and 5.)

[13] T. Buffard, S. Clain, Monoslope and Multislope MUSCL Methods for unstructured meshes, J. Comput. Phys. 229 (2010) 3745–3776. (Cited pages 55, 86, 122, and 127.)

[14] J. Casper, H. Atkins, A finite-volume high order ENO scheme for two dimensional hyperbolic systems, J. Comp. Phys. 106 (1993) 62–76. (Cited pages 2, 5, and 24.)

[15] C. E. Castro, High Order ADER FV/DG Numerical Methods for Hyperbolic Equations, PhD thesis, Monographs of the School of Doctoral Studies in Environmental Engineering (2007). (Cited page 26.)

[16] S. Clain, Finite volume L^∞-stability for hyperbolic scalar problems, preprint HAL available at `http://hal.archives-ouvertes.fr/hal-00467650/fr/`. (Cited page 58.)

[17] S. Clain, V. Clauzon, L^∞ stability of the MUSCL methods, Numer. Math. 116 (2010) 31–64. (Cited pages 58, 63, and 86.)

[18] S. Clain, S. Diot, R. Loubère, A high-order finite volume method for systems of conservation laws – Multi-dimensional Optimal Order Detection (MOOD), J. Comput. Phys. 230 (2011) 4028–4050. (Cited pages 39, 40, 45, 47, 53, 86, 88, 93, 97, 98, 122, 124, 126, 127, 128, 129, 157, 160, and 162.)

[19] S. Clain, J. Figueiredo, C. Ribeiro, A finite volume scheme for the shallow-water system with the polynomial reconstruction operator, in proceeding CSEI2012 (2012). (Cited pages 160 and 162.)

[20] I. Christov, B. Popov, New non-oscillatory central schemes on unstructured triangulations for hyperbolic systems of conservation laws, J. Comput Phy. 227 (2008) 5736–5757. (Cited page 55.)

[21] B. Cockburn, S. Y. Lin, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems, J. Comput. Phys. 84 (1989) 90–113. (Cited page 87.)

[22] B. Cockburn, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: General framework, Math. Comp. 52 (1989) 411–435. (Cited page 87.)

[23] B. Cockburn, S. Hou, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case, Math. Comp. 54 (1990) 545–581. (Cited page 87.)

[24] B. Cockburn, C.-W. Shu, The Runge-Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems, J. Comput. Phys. 141 (1998) 199–224. (Cited pages 1, 4, 79, 87, and 109.)

[25] A. Csík, M. Ricchiuto, H. Deconinck, A Conservative Formulation of the Multidimensional Upwind Residual Distribution Schemes for General Nonlinear Conservation Laws, J. Comput. Phys. 179 (2002) 286–312. (Cited page 87.)

[26] A. Dervieux, J. A. Desideri, Compressible flow solvers using unstructured grids, Rapport de recherche INRIA. n°1732 (1992). (Cited page 21.)

[27] S. Diot, S. Clain, R. Loubère, Improved detection criteria for the Multi-dimensional Optimal Order Detection (MOOD) on unstructured meshes with very high-order polynomials, Comput. Fluids 64 (2012) 43–63. (Cited pages 23, 36, 38, 39, 40, 45, 46, 47, 53, 122, 123, 124, 126, 127, 128, 129, 130, 131, 133, 136, 142, 145, 148, 153, 157, 160, and 162.)

[28] S. Diot, R. Loubère, S. Clain, The MOOD method in the three-dimensional case: Very-High-Order Finite Volume Method for Hyperbolic Systems, under review for Int. J. Numer. Meth. Fl. (2012). (Cited pages 36, 38, 39, 40, 45, 46, 48, 50, 53, 160, and 162.)

[29] M. Dumbser, D. S. Balsara, E. F. Toro and C. D. Munz, A Unified Framework for the construction of One-Step Finite-Volume and Discontinuous Galerkin Schemes on Unstructured Meshes, J. Comput. Phys. 227 (2008) 8209–8253 (Cited page 27.)

[30] M. Dumbser, M. Castro, C. Parés, E. F. Toro, ADER schemes on unstructured meshes for nonconservative hyperbolic systems: Applications to geophysical flows, Computers and Fluids 38 (2009) 1731–1748. (Cited page 87.)

[31] M. Dumbser, M. Käser, V. A. Titarev and E. F. Toro, Quadrature-Free Non-Oscillatory Finite Volume Schemes on Unstructured Meshes for Nonlinear Hyperbolic Systems, J. Comput. Phys. 226 (2007) 204–243. (Cited pages 2, 5, 23, 24, 25, 27, 86, 87, 111, 142, and 154.)

[32] M. Dumbser, M. Käser, Arbitrary High Order Non-Oscillatory Finite Volume Schemes on Unstructured Meshes for Linear Hyperbolic Systems, J. Comput. Phys. 221 (2007) 693–723. (Cited pages 18 and 87.)

[33] O. Friedrich, Weighted Essentially Non-Oscillatory Schemes for the Interpolation of Mean Values on Unstructured Grids, J. Comput. Phys. 144 (1998) 194–212. (Cited pages 2, 5, 16, 18, 25, 79, 86, and 92.)

[34] S. K. Godunov, A Finite Difference Method for the Computation of Discontinuous Solutions of the Equations of Fluid Dynamics, Mat. Sb. 47 (1959) 357–393. (Cited pages 1, 4, and 9.)

[35] J. B .Goodman, R. J. LeVeque, A geometric approach to high resolution TVD schemes, SIAM J. Numer. Anal. 25 (1988) 268–284. (Cited page 21.)

[36] F. Haider, P. Brenner, B. Courbet, J.-P. Croisille Efficient Implementation of High Order Reconstruction in Finite Volume Methods, FVCA VI Problems & Perspectives 4 (2011) 553–560. (Cited page 15.)

[37] R. Harris, Z. J. Wang, Y. Liu, Efficient quadrature-free high-order spectral volume method on unstructured grids: Theory and 2D implementation, J. Comput. Phys. 227 (2008) 1620–1642. (Cited page 87.)

[38] A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comput. Phys. 49 (1983) 357–393. (Cited page 21.)

[39] A. Harten, S. Osher, Uniformly highorder accurate nonoscillatory schemes I, SIAM J. Num. Anal. 24 (1987) 279–309. (Cited pages 24, 55, 122, and 127.)

[40] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly highorder accurate nonoscillatory schemes III, J. Comput. Phys. 71 (1987) 279–309. (Cited pages 2, 5, 24, 55, 86, 122, and 127.)

[41] C. Hu, C. W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. 150 (1999) 97–127. (Cited pages 25, 86, 107, and 108.)

[42] M. E. Hubbard, Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids, J. Comput. Phys. 155 (1999) 54–74. (Cited pages 2, 5, 21, 22, 55, 59, 86, 122, and 127.)

[43] T. J. R. Hughes, L. P. Franca, M. Mallet, A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes

equations and the second law of thermodynamics, Comput. Method Appl. Mech. Eng. 54 (1986) 223–234. (Cited pages 1 and 4.)

[44] T. J. R. Hughes, M. Mallet, A new finite element formulation for computational fluid dynamics. IV: A discontinuity-capturing operator for multidimensional advective-diffusive systems, Comput. Methods Appl. Mech. Eng. 58 (1986) 329–336. (Cited pages 1 and 4.)

[45] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228. (Cited pages 2, 5, 13, 79, and 86.)

[46] G.-S. Jiang, E. Tadmor, Non-oscillatory central schemes for multidimensional hyperbolic conservative laws, SIAM J. Sci. Comput. 19 (1998) 1892–1917. (Cited pages 94 and 119.)

[47] C. Johnson, U. Nävert, J. Pitkäranta, Finite element methods for linear hyperbolic problems, Comput. Methods Appl. Mech. Eng. 45 (1984) 285–312. (Cited pages 1 and 4.)

[48] V. P. Kolgan, Application of the minimum-derivative principle in the construction of finite-difference schemes for numerical analysis of discontinuous solutions in gas dynamics, Transactions of the Central Aerohydrodynamics Institute 3 (1972) 68–77, in Russian. (Cited pages 20, 55, 59, 86, 122, and 127.)

[49] V. P. Kolgan, Finite-difference schemes for computation of three dimensional solutions of gas dynamics and calculation of a flow over a body under an angle of attack, Transactions of the Central Aerohydrodynamics Institute 6 (1975) 1–6, in Russian. (Cited pages 20, 55, 59, 86, 122, and 127.)

[50] V. P. Kolgan, Application of the principle of minimizing the derivative to the construction of finite-difference schemes for computing discontinuous solutions of gas dynamics, J. Comput. Phys. 230 (2010) 2384–2390. (Cited pages 20, 55, 59, 86, 122, and 127.)

[51] P. D. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical computation, Comm. Pure and Appl. Math. 7 (1954) 159–193. (Cited page 142.)

[52] R. J. Leveque, High-Resolution Conservative Algorithms for Advection in Incompressible Flow, SIAM J. Num. Anal. 33 (1996) 627–665. (Cited pages 66, 86, and 99.)

[53] R. Liska, B. Wendroff, Comparison of several difference schemes on 1D and 2D test problems for the Euler equations, SIAM J. Sci. Comput. 25 (2003) 995–1017. (Cited page 74.)

[54] X. D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212. (Cited page 24.)

[55] R. Loubère, P.-H. Maire, P. Vachal, Staggered Lagrangian discretization based on cell-centered Riemann solver and associated hydrodynamics scheme, Commun. Comput. Phys. 10 (2011) 940–978. (Cited pages 113 and 151.)

[56] P.-H. Maire, A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes, J. Comput. Phys. 228 (2009) 2391–2425. (Cited page 113.)

[57] C. Ollivier-Gooch, Quasi-ENO Schemes for Unstructured Meshes Based on Unlimited Data-Dependent Least-Squares Reconstruction, J. Comput. Phys. 133 (1997) 6–17. (Cited pages 15, 60, 61, 86, and 92.)

[58] C. Ollivier-Gooch, A. Nejat, K. Michalak, Obtaining and verifying high-order unstructured finite volume solutions to the Euler equations, AIAA Journal 47 (2009) 2105–2120. (Cited page 24.)

[59] S. Osher, S. Chakravarthy, High resolution schemes and the entropy condition, SIAM J. Numer. Anal. 21 (1984) 955–984. (Cited pages 94 and 119.)

[60] S. Osher, J. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulation, J. Comput. Phys. 79 (1988). (Cited pages 2 and 5.)

[61] B. Parent, Positivity-preserving High-resolution Schemes for Systems of Conservation Laws, J. Comput. Phys. 231 (2012) 173–189. (Cited page 22.)

[62] J. S. Park, S.-H. Yoon, C. Kim, Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids, J. Comput. Phys. 229 (2010) 788–812. (Cited pages 55, 56, 58, 59, 65, 68, 74, 86, 102, and 151.)

[63] J. Qiu, C.-W. Shu, Runge–Kutta Discontinuous Galerkin Method using WENO Limiters, SIAM J. Sci. Comput. 26 (2005) 907–929. (Cited pages 1 and 4.)

[64] M. Ricchiuto, A. Bollermann, Stabilized residual distribution for shallow water simulations, J. Comput. Phys. 228 (2009) 1071–1115. (Cited page 87.)

[65] R. Sander, A third-order accurate variation non-expansive difference scheme for single nonlinear conservation law, Math. Comput. 51 (1988) 535–558. (Cited pages 94 and 119.)

[66] J. Shi, C. Hu, C.-W. Shu, A technique of treating negative weights in WENO schemes, J. Comput. Phys. 175 (2002) 108–127. (Cited page 86.)

[67] J. Shi, Y.-T Zhang, C.-W. Shu, Resolution of high order WENO schemes for complicated flow structures, J. Comput. Phys. 186 (2003) 690–696. (Cited page 79.)

[68] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes in Mathematics 1697 Springer (1998) 325–432. (Cited pages 86, 105, and 144.)

[69] C.-W. Shu, High order weighted essentially non-oscillatory schemes for convection dominated problems, SIAM Review Vol. 51 No 1 (2009) 82–126. (Cited pages 55, 122, and 127.)

[70] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing scheme, J. Comput. Phys. 77 (1988) 439–471. (Cited pages 55, 58, 86, 91, 122, 127, and 143.)

[71] C. W. Schulz-Rinne, J. P. Collins, H. M. Glaz, Numerical solution of the Riemann problem for two-dimensional gas dynamics, SIAM J. Sci. Comput. 14 (1993) 1394–1414. (Cited page 74.)

[72] B. W. Skews, H. Kleine, Flow features resulting from shock wave impact on a cylindrical cavity, J. Fluid. Mech. 580 (2007) 481–493. (Cited pages 48, 105, 113, 115, 145, and 148.)

[73] G. A. Sod, A survey of several finite difference methods for systems of non-linear hyperbolic conservation laws, J. Comput. Phys. 27 (1978) 1–31. (Cited page 142.)

[74] T. Sonar, On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conservation laws on general triangulations: polynomial recovery, accuracy and stencil selection, Comput. Methods Appl. Mech. Engrg. 140 (1997) 157–181. (Cited page 24.)

[75] S. Spekreijse, Multigrid Solution of Monotone Second-Order Discretizations of Hyperbolic Conservation Laws, Math. Comp. 49 (1987) 135–155. (Cited pages 2, 5, and 21.)

[76] G. W. Stewart, Matrix Algorithms, Volume 1: Basic Decompositions, Society for Industrial and Applied Mathematics SIAM (1998). (Cited pages 16 and 17.)

[77] V. A. Titarev, E. F. Toro, ADER schemes for three-dimensional non-linear hyperbolic systems, J. Comput. Phys. 204 (2005) 715–736. (Cited pages 2, 5, 24, and 87.)

[78] E. F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, 3^{rd} revision, Springer-Verlag Berlin and Heidelberg GmbH & Co. K (2009). (Cited pages 11, 41, 74, 142, and 151.)

[79] E. F. Toro, R. C. Millington, L. A. M. Nejad, Towards very high order Godunov schemes, Godunov Methods. Theory and Applications, Kluwer/Plenum Academic Publishers (2001) 907–940. (Cited page 24.)

[80] E. F. Toro, V. A. Titarev, Solution of the generalized Riemann problem for advection-reaction equations, Proc. Roy. Soc. London A 458 (2002) 271–281. (Cited page 26.)

[81] E. F. Toro, A. Hidalgo, ADER finite volume schemes for nonlinear reaction-diffusion equations, Applied Numerical Mathematics 59 (2009) 73–100. (Cited page 87.)

[82] P. Tsoutsanis, V.A. Titarev, D. Drikakis. WENO schemes on arbitrary mixed-element unstructured meshes in three space dimensions, J. Comput. Phys. 230 (2011) 1585–1601. (Cited pages 2, 5, 24, 25, and 86.)

[83] B. Van Leer, Towards the Ultimate Conservative Difference Scheme I. The Quest for Monotonicity, Lecture Notes in Physics 18 (1973) 163–168. (Cited pages 20, 86, 122, and 127.)

[84] B. Van Leer, Towards the Ultimate Conservative Difference Scheme II. Monotonicity and Conservation Combined in a Second Order Scheme, J. Comput. Phys. 14 (1974) 361–370. (Cited pages 20, 55, 59, 86, 122, and 127.)

[85] B. Van Leer, Towards the Ultimate Conservative Difference Scheme III. UpstreamCentered Finite Difference Schemes for Ideal Compressible Flow, J. Comput. Phys. 23 (1977) 263–275. (Cited page 20.)

[86] B. Van Leer, Towards the Ultimate Conservative Difference Scheme IV. A New Approach to Numerical Convection, J. Comput. Phys. 23 (1977) 276–299. (Cited page 20.)

[87] B. Van Leer, Towards the Ultimate Conservative Difference Scheme V. A Second Order Sequel to Godunovs Method, J. Comput. Phys. 32 (1979) 101–136. (Cited pages 2, 5, and 20.)

[88] Z. J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, J. Comput. Phys. 178 (2002) 210–251. (Cited page 87.)

[89] Z. J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids: extension to two dimensional scalar equation, J. Comput. Phys. 179 (2002) 665–697. (Cited page 87.)

[90] E. Weinan, C.-W. Shu, A numerical resolution study of high order essentially non-oscillatory schemes applied to incompressible flow, J. Comput. Phys. 110 (1994). (Cited pages 2 and 5.)

[91] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1984) 115–173. (Cited pages 24, 74, 76, 79, 109, and 143.)

[92] W. R. Wolf, J. L. F. Azevedo, High-order ENO and WENO schemes for unstructured grids, International Journal for Numerical Methods in Fluids, 55 (2007) 917–943. (Cited pages 55 and 86.)

[93] H. C. Yee, M. Vinokur, M. J. Djomehri, Entropy Splitting and Numerical Dissipation, J. Comput. Phys. 162 (2000) 33–81. (Cited pages 105 and 144.)

[94] S.-H. Yoon, C. Kim, K. H. Kim, Multi-dimensional limiting process for three-dimensional flow physics analyses, J. Comput. Phys. 227 (2008) 6001–6043. (Cited pages 55 and 56.)

[95] Y.-T. Zhang and C.-W. Shu Third-order WENO scheme on three dimensional tetrahedral meshes, Com. Comput. Phys. 5 (2009) 836–848. (Cited pages 86, 122, and 127.)

[96] X. Zhang and C.-W. Shu Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments, Proc. R. Soc. A 467 (2011) 2752–2776. (Cited pages 97 and 119.)

Appendix A

Quadrature rules with positive weights up to 6^{th}-order

In this appendix, we provide the Gaussian quadrature rules for 1D, 2D and 3D elements up to 6^{th}-order of accuracy that we used in our simulation codes. These quadrature rules are all made up with positive weights to ensure the convex combination property. Moreover, we recall that polygonal and polyhedral cells have simply been treated by triangular and tetrahedral decompositions to.

A.1 Quadrature rules for a segment

We consider a segment K defined by two points $a, b \in \mathbb{R}\,(\text{or}\mathbb{R}^2)$ and seek to approximate the integral $\int_K f(\mathbf{x})d\mathbf{x}$. To this end we consider the Gaussian quadrature formulae that provides an approximation under the form

$$\int_K f(\mathbf{x})d\mathbf{x} \approx |K| \sum_{r=1}^{R} \xi_r f(q_r).$$

We give in the following table, for degree one to five, the quadrature weights ξ_r and the barycentric coordinates (α_r, β_r) corresponding to the quadrature points $q_r = \alpha_r a + \beta_r b$.

Degree	R	Bary. coord.	Weight ξ_r
1	1	$\left(\frac{1}{2}, \frac{1}{2}\right)$	1
2,3	2	$\left(\frac{1}{2} \pm \frac{1}{2}\sqrt{\frac{1}{3}}, \frac{1}{2} \mp \frac{1}{2}\sqrt{\frac{1}{3}}\right)$	$\frac{1}{2}$
4,5	3	$\left(\frac{1}{2}, \frac{1}{2}\right)$	$\frac{8}{18}$
		$\left(\frac{1}{2} \pm \frac{1}{2}\sqrt{\frac{3}{5}}, \frac{1}{2} \mp \frac{1}{2}\sqrt{\frac{3}{5}}\right)$	$\frac{5}{18}$

A.2 Quadrature rules for a triangle

We consider a triangle K defined by three points $a, b, c \in \mathbb{R}^2$ (or$\mathbb{R}^3$) and seek to approximate the integral $\int_K f(\mathbf{x})d\mathbf{x}$. To this end we consider the quadrature formulae that provides an approximation under the form

$$\int_K f(\mathbf{x})d\mathbf{x} \approx |K| \sum_{r=1}^{R} \xi_r f(q_r).$$

We give in the following table, for degree one to five, the quadrature weights ξ_r and the barycentric coordinates $(\alpha_r, \beta_r, \gamma_r)$ corresponding to the quadrature points $q_r = \alpha_r a + \beta_r b + \gamma_r c$. For the sake of clarity, the barycentric coordinates are given with their multiplicities which provide the number of permutations that have to be performed to obtain all quadrature points.

Degree	R	Bary. coord.	Multiplicity	Weight ξ_r
1	1	$\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$	1	1
2	3	$\left(\frac{1}{2}, \frac{1}{2}, 0\right)$	3	$\frac{1}{3}$
3,4	6	$\left(a_i, a_i, 1-2a_i\right) \quad i \in \{1,2\}$	3	
		$a_1 = 0.091576213509771$		0.109951743655322
		$a_2 = 0.445948490915965$		0.223381589678011
5	7	$\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$	1	$\frac{9}{40}$
		$\left(a_i, a_i, 1-2a_i\right) \quad i \in \{1,2\}$	3	
		$a_1 = \frac{5-\sqrt{15}}{21}$		$\frac{155-\sqrt{15}}{1200}$
		$a_2 = \frac{5+\sqrt{15}}{21}$		$\frac{155+\sqrt{15}}{1200}$

A.3 Quadrature rules for a tetrahedron

We consider a tetrahedron K defined by four points $a, b, c, d \in \mathbb{R}^3$ and seek to approximate the integral $\int_K f(\mathbf{x})d\mathbf{x}$. To this end we consider the quadrature formulae that provides an approximation under the form

$$\int_K f(\mathbf{x})d\mathbf{x} \approx |K| \sum_{r=1}^{R} \xi_r f(q_r).$$

We give in the following table, for degree one to five, the quadrature weights ξ_r and the barycentric coordinates $(\alpha_r, \beta_r, \gamma_r, \delta_r)$ corresponding to the quadrature points $q_r = \alpha_r a + \beta_r b + \gamma_r c + \delta_r d$.

For the sake of clarity, the barycentric coordinates are given with their multiplicities which provide the number of permutations that have to be performed to obtain all quadrature points.

Degree	R	Bary. coord.	Multiplicity	Weight ξ_r
1	1	$\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)$	1	1
2	4	$\left(a, a, a, 1-3a\right)$ $a = \frac{5-\sqrt{5}}{20}$	4	$\frac{1}{4}$
3,4,5	15	$\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)$	1	$\frac{16}{135}$
		$\left(a_i, a_i, a_i, 1-3a_i\right) \quad i \in \{1,2\}$	4	
		$a_1 = \frac{7-\sqrt{15}}{34}$		$\frac{2665-14\sqrt{15}}{37800}$
		$a_2 = \frac{7+\sqrt{15}}{34}$		$\frac{2665+14\sqrt{15}}{37800}$
		$\left(a, a, \frac{1}{2}-a, \frac{1}{2}-a\right)$	6	
		$a_2 = \frac{10-2\sqrt{15}}{40}$		$\frac{10}{189}$

Appendix B

Linear dependence of the polynomial reconstruction and evaluation on mean values

In the first section, we propose to *explicitly* write each polynomial coefficients as a linear combination of the mean value on the reference cell K_i and the ones of the stencil $\mathcal{S}_i^{\mathsf{d}}$. We then draw a remark about the ways to obtain the *curvatures* used in the $u2$ detection process. The second part is dedicated to explicitly writing each reconstructed value as a similar linear combination. This result is new up to our knowledge and may be an important improvement for the parallelized version of the MOOD method.

Notice that the notation used in this chapter is the one defined in section 1.2 and the index $^{\mathsf{d}}$ corresponding to the degree of the polynomial reconstruction is omitted for the sake of clarity.

B.1 Linear dependence of the polynomial coefficients on neighbors mean values

In this section we seek to write the polynomial coefficients $\mathcal{R}_i^{\boldsymbol{\alpha}}$ as a linear combination of the mean value on the reference cell K_i and the ones of the stencil $\mathcal{S}_i^{\mathsf{d}}$, that is

$$\mathcal{R}_i^{\boldsymbol{\alpha}} = \sum_{j \in \mathcal{S}_i \cup \{i\}} \gamma_{i,j}^{\boldsymbol{\alpha}} \, u_j, \tag{B.1}$$

where $\gamma_{i,j}^{\boldsymbol{\alpha}}$ are the weights of the combination and only depend on geometrical entities.

To this end, we recall that the solution of the reconstruction problem, *i.e.* the polynomial coefficients $\mathcal{R}_i = \left\{\mathcal{R}_i^{\boldsymbol{\alpha}}\right\}_{1 \leq |\boldsymbol{\alpha}| \leq \mathsf{d}}$, are obtained by

$$\mathcal{R}_i = \mathsf{X}_i^{\dagger} U_{\mathcal{S}_i},$$

where $U_{\mathcal{S}_i} = \left(u_{\mathcal{S}_i(1)} - u_i, \text{—}, u_{\mathcal{S}_i(N_{\mathcal{S}_i})} - u_i\right]^t$ and $\mathsf{X}_i^{\dagger}$ is the pseudoinverse of the reconstruction matrix X_i for which each row corresponds to a multiindex $\boldsymbol{\alpha}$ and each column to one neighbor

in the stencil S_i. Then we exploit the linearity of the matrix-vector product and obtain that each polynomial coefficient writes

$$\mathcal{R}_i^{\boldsymbol{\alpha}} = \sum_{j=1}^{N_{\mathcal{S}_i}} \left(\mathsf{X}_i^\dagger \left(\boldsymbol{\alpha}, \mathcal{S}_i(j) \right) \; u_j \right) - \left(\sum_{j=1}^{N_{\mathcal{S}_i}} \mathsf{X}_i^\dagger \left(\boldsymbol{\alpha}, \mathcal{S}_i(j) \right) \right) u_i,$$

where the notation $\mathsf{X}_i^\dagger \left(\boldsymbol{\alpha}, \mathcal{S}_i(j) \right)$ stands for the term of $\mathsf{X}_i^\dagger$ of the row corresponding to the multiindex $\boldsymbol{\alpha}$ and the column corresponding to neighboring cell $K_{\mathcal{S}_i(j)}$.

Therefore the contribution of the neighbor mean value of cell $K_{\mathcal{S}_i(j)}$ in equation (B.1) is

$$\gamma_{i,j}^{\boldsymbol{\alpha}} = \mathsf{X}_i^\dagger \left(\boldsymbol{\alpha}, \mathcal{S}_i(j) \right),$$

and the one of K_i is

$$\gamma_{i,i}^{\boldsymbol{\alpha}} = - \sum_{j=1}^{N_{\mathcal{S}_i}} \mathsf{X}_i^\dagger \left(\boldsymbol{\alpha}, \mathcal{S}_i(j) \right).$$

Finally this development may be used to directly compute the *curvatures* necessary to the $u2$ detection process from the vector of mean values on the reconstruction stencil and on the reference cell. Nevertheless let us remark that the polynomial coefficients may have been directly obtained through a linear combination of the form

$$\mathcal{R}_i^{\boldsymbol{\alpha}} = \sum_{j \in \mathcal{S}_i \cup \{i\}} \gamma_{i,j}^{\boldsymbol{\alpha}} \, (u_j - u_i).$$

However the equation (B.1) is important for the result provided in the next section.

At last we discuss the way to obtain the *curvatures* in the $u2$ detection process. The use of the pseudoinverse is not always considered in the literature, and there exist high-order Finite Volume methods for which the decomposition, say QR for instance, of the matrix X_i is stored and directly used to compute the polynomial coefficients through $\mathcal{R}_i = R^{-1} Q^T U_{\mathcal{S}_i}$. This technique reduces the condition number problems since R is only inverted one time. However it is impossible to compute each polynomial coefficient independently such as we proposed above. In consequence, if we were to use such a technique to obtain the *curvatures* for the $u2$ detection process, we would have to compute almost all polynomial coefficients to only get the curvatures. On the opposite, a major advantage of the pseudoinverse technique, is the possibility to directly obtain the curvatures without computing all polynomial coefficients.

B.2 Linear dependence of the reconstructed values on neighbors mean values

In this section, we propose to write the evaluation of the reconstructed polynomial $\widetilde{u}_i$ of degree d at any point $\mathbf{x}$ as a linear combination of the mean value on the reference cell K_i and the

ones of the stencil $\mathcal{S}_i^{\mathsf{d}}$, that is

$$\widetilde{u}_i(\mathbf{x}) \;=\; \sum_{j\in\mathcal{S}_i\cup\{i\}} \beta_{i,j}(\mathbf{x})\; u_j. \tag{B.2}$$

B.2.1 Obtaining the linear combination

We recall the two necessary elements that we need: first, the reconstruction form

$$\widetilde{u}_i(\mathbf{x}) \;=\; u_i + \sum_{1\le|\boldsymbol{\alpha}|\le\mathsf{d}} \mathcal{R}_i^{\boldsymbol{\alpha}}\left((\mathbf{x}-\mathbf{c}_{K_i})^{\boldsymbol{\alpha}} - \frac{1}{|K_i|}\int_{K_i}(\mathbf{x}-\mathbf{c}_{K_i})^{\boldsymbol{\alpha}}\,d\mathbf{x}\right),$$

and the linear combination of previous section to obtain the polynomial coefficients

$$\mathcal{R}_i^{\boldsymbol{\alpha}} \;=\; \sum_{j\in\mathcal{S}_i\cup\{i\}} \gamma_{i,j}^{\boldsymbol{\alpha}}\; u_j.$$

For the sake of clarity, let us define

$$\mathsf{M}_i^{\boldsymbol{\alpha}}(\mathbf{x}) \;=\; (\mathbf{x}-\mathbf{c}_{K_i})^{\boldsymbol{\alpha}} - \frac{1}{|K_i|}\int_{K_i}(\mathbf{x}-\mathbf{c}_{K_i})^{\boldsymbol{\alpha}}\,d\mathbf{x},$$

where M stands for monomials. The polynomial reconstruction can thus be written

$$\widetilde{u}_i(\mathbf{x}) \;=\; u_i + \sum_{1\le|\boldsymbol{\alpha}|\le\mathsf{d}} \mathcal{R}_i^{\boldsymbol{\alpha}}\mathsf{M}_i^{\boldsymbol{\alpha}}(\mathbf{x}),$$

and replacing $\mathcal{R}_c^{\boldsymbol{\alpha}}$ by the linear combination leads to

$$\widetilde{u}_i(\mathbf{x}) \;=\; u_i + \sum_{1\le|\boldsymbol{\alpha}|\le\mathsf{d}} \left(\sum_{j\in\mathcal{S}_i\cup\{i\}} \gamma_{i,j}^{\boldsymbol{\alpha}}\; u_j\right) \mathsf{M}_i^{\boldsymbol{\alpha}}(\mathbf{x}),$$

that equivalently writes

$$\widetilde{u}_i(\mathbf{x}) \;=\; u_i + \sum_{j\in\mathcal{S}_i\cup\{i\}} \left(\sum_{1\le|\boldsymbol{\alpha}|\le\mathsf{d}} \gamma_{i,j}^{\boldsymbol{\alpha}}\; \mathsf{M}_i^{\boldsymbol{\alpha}}(\mathbf{x})\right) u_j.$$

Therefore the contribution of the neighbor mean value of cell $K_{\mathcal{S}_i(j)}$ in equation (B.2) is

$$\beta_{i,\mathcal{S}(j)}(\mathbf{x}) \;=\; \sum_{1\le|\boldsymbol{\alpha}|\le\mathsf{d}} \gamma_{i,j}^{\boldsymbol{\alpha}}\; \mathsf{M}_i^{\boldsymbol{\alpha}}(\mathbf{x}) \;=\; \sum_{1\le|\boldsymbol{\alpha}|\le\mathsf{d}} \mathsf{X}_i^{\dagger}\left(\boldsymbol{\alpha},\mathcal{S}_i(j)\right) \mathsf{M}_i^{\boldsymbol{\alpha}}(\mathbf{x}),$$

and the one of K_i is

$$\beta_{i,i}(\mathbf{x}) \;=\; 1 + \sum_{1\le|\boldsymbol{\alpha}|\le\mathsf{d}} \gamma_{i,i}^{\boldsymbol{\alpha}}\; \mathsf{M}_i^{\boldsymbol{\alpha}}(\mathbf{x}) \;=\; 1 - \sum_{1\le|\boldsymbol{\alpha}|\le\mathsf{d}} \sum_{j=1}^{N_{\mathcal{S}_i}} \mathsf{X}_i^{\dagger}\left(\boldsymbol{\alpha},\mathcal{S}_i(j)\right)\; \mathsf{M}_i^{\boldsymbol{\alpha}}(\mathbf{x}).$$

The definition of all $\beta_{i,j}(\mathbf{x}), j\in\mathcal{S}_i\cup\{i\}$ is therefore complete and only depends on geometry. In other words, for given mesh, polynomial degree, reconstruction stencil and quadrature point, all weights appearing in equation (B.2) may be precomputed and stored.

B.2.2 Using the linear combination

Following this claim, the evaluation of a polynomial at a given quadrature point does not require the computation of the polynomial coefficients. More precisely, the high-order approximation at a given point $\mathbf{x}$ is computed by a dot product of two vectors of size $N_{\mathcal{S}_i}+1$, where the first vector is determined during the preprocessing step and the other is vector of the mean value on the reference cell K_i and the ones of the stencil $\mathcal{S}_i^{\mathsf{d}}$

$$\widetilde{u}_c(\mathbf{x}) = \begin{pmatrix} \beta_{i,i}(\mathbf{x}) \\ \beta_{i,\mathcal{S}_i(1)}(\mathbf{x}) \\ \vdots \\ \beta_{i,\mathcal{S}_c(N_{\mathcal{S}_i})}(\mathbf{x}) \end{pmatrix} - \begin{pmatrix} u_i \\ u_{\mathcal{S}_i(1)} \\ \vdots \\ u_{\mathcal{S}_c i(N_{\mathcal{S}_i})} \end{pmatrix}.$$

We can then imagine to reconstruct all quadrature points values in a cell in one matrix-vector operation. For instance, considering the $(q_1, q_2, \ldots, q_R)$ where we want to reconstruct $\widetilde{u}_i(\dashv$, the operation is

$$\begin{pmatrix} \widetilde{u}_i(q_1) \\ \widetilde{u}_i(q_2) \\ \vdots \\ \widetilde{u}_i(q_R) \end{pmatrix} = \begin{pmatrix} \beta_{i,i}(q_1) & \beta_{i,\mathcal{S}_i(1)}(q_1) & \cdots & \beta_{i,\mathcal{S}_i(N_{\mathcal{S}_i})}(q_1) \\ \beta_{i,i}(q_2) & \beta_{i,\mathcal{S}_i(1)}(q_2) & \cdots & \beta_{i,\mathcal{S}_i(N_{\mathcal{S}_i})}(q_2) \\ \vdots & \vdots & \cdots & \vdots \\ \beta_{i,i}(q_R) & \beta_{i,\mathcal{S}_i(1)}(q_R) & \cdots & \beta_{i,\mathcal{S}_i(N_{\mathcal{S}_i})}(q_R) \end{pmatrix} \begin{pmatrix} u_i \\ u_{\mathcal{S}_i(1)} \\ \vdots \\ u_{\mathcal{S}_i(N_{\mathcal{S}_i})} \end{pmatrix},$$

where the matrix is only depending on the geometry, the polynomial degree and the quadrature points. Remark that it may be more interesting in a computational point of view to reconstruct values at all quadrature points of a face, if the code if face-based.

Finally we discuss the computational interest of such a technique. Though it seems more efficient not to reconstruct the polynomial coefficients, one has to be careful about the number of operations. Indeed let us assume that we intend to reconstruct the values of the reconstructed polynomial $\widetilde{u}_i$ at all quadrature points of a cell K_i.

Using the traditional technique, we need to first compute the polynomial coefficients and it demands $\mathcal{N}(\mathsf{d})$ dot products between vector of size $N_{\mathcal{S}_i}$ (where we recall that $\mathcal{N}(\mathsf{d})$ is the number of polynomial coefficients). Then the reconstructed values are obtained by R (*i.e.* number of quadrature points on all faces) dot products between vector of size $\mathcal{N}(\mathsf{d})$. Consequently the total number of operation is

$$N_{old} = \mathcal{N}(\mathsf{d})(2N_{\mathcal{S}_i} - 1) + R(2\mathcal{N}(\mathsf{d}) - 1).$$

Considering the proposed technique we need R dot products between vectors of size $N_{\mathcal{S}_i}$ so that the total number of operations is

$$N_{new} = R(2N_{\mathcal{S}_i} - 1).$$

It is now interesting to compare these two operations numbers when the stencil size is set according to $\mathcal{N}(\mathsf{d})$. We consider the three cases, $N_{\mathcal{S}_i} = \mathcal{N}(\mathsf{d})$, $N_{\mathcal{S}_i} = (3/2)\mathcal{N}(\mathsf{d})$ and $N_{\mathcal{S}_i} = 2\mathcal{N}(\mathsf{d})$ and seek the condition linking the number of quadrature points to the number of polynomial coefficients such that the cost of both methods is the same.

⋆ Case $N_{\mathcal{S}_i} = \mathcal{N}(\mathsf{d})$

The first case implies that $N_{old} = N_{new} + \mathcal{N}(\mathsf{d})(2\mathcal{N}(\mathsf{d}) - 1)$, so that the cost of the new technique is always lower than the traditional one. However the condition $N_{\mathcal{S}_i} = \mathcal{N}(\mathsf{d})$ means that the stencil size is the minimal required one, and we claimed in section 1.2 that this is not sufficient in the general case.

⋆ Case $N_{\mathcal{S}_i} = (3/2)\mathcal{N}(\mathsf{d})$

The second case is realistic since $N_{\mathcal{S}_i} = (3/2)\mathcal{N}(\mathsf{d})$ is basically the relation that holds in 2D. We write the equality between both numbers of operation and obtain

$$\begin{aligned} & \mathcal{N}(\mathsf{d})(3\mathcal{N}(\mathsf{d}) - 1) + R(2\mathcal{N}(\mathsf{d}) - 1) &&= R(3\mathcal{N}(\mathsf{d}) - 1) \\ \Leftrightarrow \quad & \mathcal{N}(\mathsf{d})(3\mathcal{N}(\mathsf{d}) - R - 1) &&= 0 \\ \Leftrightarrow \quad & 3\mathcal{N}(\mathsf{d}) - R - 1 &&= 0 \\ \Leftrightarrow \quad & R &&= 3\mathcal{N}(\mathsf{d}) - 1. \end{aligned}$$

It means that if the number of quadrature points on all faces of the cell K_i is lower than three times the number of polynomial coefficients the new technique is cheaper than the old one.

In order to better understand the implications of such a constraint, we give in the following table, the number of faces f_{lim} below which the new technique is cheaper. We moreover recall the number of polynomial coefficients $\mathcal{N}(\mathsf{d})$ and the number of quadrature points per face QP_f necessary to reach optimal order according to the degree of the polynomial.

We assume that in 3D faces are triangular, but it does not restrict the results since other faces can be triangulated. In consequence a pyramid may be seen as a 6 faces polyhedron, a prism as an 8 faces one and an hexahedron as a 12 faces one.

Domain	Degree d	$\mathcal{N}(\mathsf{d})$	QP_f	f_{lim}
$\Omega \subset \mathbb{R}^2$	1	2	1	6
	3	9	2	13
	5	20	3	19
$\Omega \subset \mathbb{R}^3$	1	3	1	8
	3	19	6	9
	5	55	7	23

The values of f_{lim} shows that in 2D the new technique basically always performs better while in 3D it is true for tetrahedra, pyramids or prisms. Furthermore the greater the polynomial degree is, the more gain the new technique brings, in particular for the degree five.

⋆ Case $N_{\mathcal{S}_i} = 2\mathcal{N}(\mathsf{d})$

This case is closer to the 3D case since $N_{\mathcal{S}_i} = (3/2)\mathcal{N}(\mathsf{d})$ is basically the relation that holds. We write the equality between both numbers of operation and obtained

$$\begin{aligned} & \mathcal{N}(\mathsf{d})(4\mathcal{N}(\mathsf{d})-1) + R(2\mathcal{N}(\mathsf{d})-1) &=& R(4\mathcal{N}(\mathsf{d})-1) \\ \Leftrightarrow \quad & \mathcal{N}(\mathsf{d})(4\mathcal{N}(\mathsf{d})-2R-1) &=& 0 \\ \Leftrightarrow \quad & 4\mathcal{N}(\mathsf{d})-2R-1 &=& 0 \\ \Leftrightarrow \quad & R &=& 2\mathcal{N}(\mathsf{d})-1/2 \end{aligned}$$

It means that if the number of quadrature points on all faces of the cell K_i is lower than two times the number of polynomial coefficients the new technique is cheaper than the old one.

As for the previous case, we provide in the following table the number of faces f_{lim} below which the new technique is cheaper.

Domain	Degree d	$\mathcal{N}(\mathsf{d})$	QP_f	f_{lim}
$\Omega \subset \mathbb{R}^2$	1	2	1	3
	3	9	2	8
	5	20	3	13
$\Omega \subset \mathbb{R}^3$	1	3	1	5
	3	19	6	6
	5	55	7	15

The values of f_{lim} shows that in 2D the new technique always performs better except for degree one on cells with more than 3 faces. However in 3D, the technique is interesting for degree 5 again, or for tetrahedral meshes. This result supports the need for reduction of the stencil size.

We conclude this appendix with two important remarks. First this direct approach to obtain the high-order approximations at quadrature points is more flexible since only one global step (a matrix-vector product) is necessary whereas the traditional technique demands to first reconstruct the polynomial coefficients and then the reconstructed values. It may be of crucial importance for the sake of parallelization. Finally this few examples completely support the fact that we should try to reduce the stencil size since the less cells there are in the stencil, the more efficient the new technique is.

Résumé

Nous introduisons et développons dans cette thèse un nouveau type de méthodes Volumes Finis d'ordre très élevé pour les systèmes hyperboliques de lois de conservations. Appelée MOOD pour Multidimensional Optimal Order Detection, elle permet de réaliser des simulations très précises en dimensions deux et trois sur maillages non-structurés. La conception d'une telle méthode est rendue délicate par l'apparition de singularités dans la solution (chocs, discontinuités de contact) pour lesquelles des phenomènes parasites (oscillations, création de valeurs non physiques...) sont générés par l'approximation d'ordre élevé.

L'originalité de cette thèse réside dans le traitement de ces problèmes. À l'opposé des méthodes classiques qui essaient de contrôler ces phénomènes indésirables par une limitation *a priori*, nous proposons une approche de traitement *a posteriori* basée sur une décrémentation locale de l'ordre du schéma. Nous montrons en particulier que ce concept permet très simplement d'obtenir des propriétés qui sont habituellement difficiles à prouver dans le cadre multi-dimensionel non-structuré (préservation de la positité par exemple).

La robustesse et la qualité de la méthode MOOD ont été prouvées sur de nombreux tests numériques en 2D et 3D. Une amélioration significative des ressources informatiques (CPU et stockage mémoire) nécessaires à l'obtention de résultats équivalents aux méthodes actuelles a été démontrée.

Abstract

We introduce and develop in this thesis a new type of very high-order Finite Volume methods for hyperbolic systems of conservation laws. This method, named MOOD for Multidimensional Optimal Order Detection, provides very accurate simulations for two- and three-dimensional unstructured meshes. The design of such a method is made delicate by the emergence of solution singularities (shocks, contact discontinuities) for which spurious phenomena (oscillations, non-physical values creation, etc.) are generated by the high-order approximation.

The originality of this work lies in a new treatment for theses problems. Contrary to classical methods which try to control such undesirable phenomena through an *a priori* limitation, we propose an *a posteriori* treatment approach based on a local scheme order decrementing. In particular, we show that this concept easily provides properties that are usually difficult to prove in a multidimensional unstructured framework (positivity-preserving for instance).

The robustness and quality of the MOOD method have been numerically proved through numerous test cases in 2D and 3D, and a significant reduction of computational resources (CPU and memory storage) needed to get state-of-the-art results has been shown.