A high-order Discontinuous Galerkin discretization for solving two-dimensional Lagrangian hydrodynamics

François Vilar

▶ **To cite this version:**

François Vilar. A high-order Discontinuous Galerkin discretization for solving two-dimensional Lagrangian hydrodynamics. Numerical Analysis [math.NA]. Université Sciences et Technologies - Bordeaux I, 2012. English. NNT : . tel-00765575

HAL Id: tel-00765575
https://theses.hal.science/tel-00765575

Submitted on 14 Dec 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE de DOCTORAT

PRÉSENTÉE à

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

Par **François Vilar**

Utilisation des méthodes Galerkin discontinues pour la résolution de l'hydrodynamique Lagrangienne bi-dimensionnelle

A high-order Discontinuous Galerkin discretization for solving two-dimensional Lagrangian hydrodynamics

---

Soutenue le : 16 novembre 2012

Après avis des rapporteurs :

Chi-Wang Shu	Professeur, George Brown University
Bruno Després	Professeur, Université Paris VI

Devant la commission d'examen composée de :

Rémi Abgrall	Professeur, Institut Polytechnique de Bordeaux	Examinateur
Daniel Bouche	Directeur de Recherche, CEA	Examinateur
Thierry Colin	Professeur, Institut Polytechnique de Bordeaux	Examinateur
Bruno Després	Professeur, Université Paris VI	Rapporteur
Patrick Le Tallec	Professeur, Ecole Polytechnique	Examinateur
Raphaël Loubère	Chargé de Recherche, Institut de Math. de Toulouse	Examinateur
Pierre-Henri Maire	Ingénieur-chercheur, CEA	Examinateur

– 2012 –

À ma mère

Abstracts

Français

Le travail présenté ici avait pour but le développement d'un schéma de type Galerkin discontinu (GD) d'ordre élevé pour la résolution des équations de la dynamique des gaz écrites dans un formalisme Lagrangien total, sur des maillages bi-dimensionnels totalement déstructurés. À cette fin, une méthode progressive a été utilisée afin d'étudier étape par étape les difficultés numériques inhérentes à la discrétisation Galerkin discontinue ainsi qu'aux équations de la dynamique des gaz Lagrangienne. Par conséquent, nous avons développé dans un premier temps des schémas de type Galerkin discontinu jusqu'à l'ordre trois pour la résolution des lois de conservation scalaires mono-dimensionnelles et bi-dimensionnelles sur des maillages déstructurés. La particularité principale de la discrétisation GD présentée est l'utilisation des bases polynomiales de Taylor. Ces dernières permettent, dans le cadre de maillages bi-dimensionnels déstructurés, une prise en compte globale et unifiée des différentes géométries. Une procédure de limitation hiérarchique, basée aux noeuds et préservant les extrema réguliers a été mise en place, ainsi qu'une forme générale des flux numériques assurant une stabilité globale L_2 de la solution. Ensuite, nous avons tâché d'appliquer la discrétisation Galerkin discontinue développée aux systèmes mono-dimensionnels de lois de conservation comme celui de l'acoustique, de Saint-Venant et de la dynamique des gaz Lagrangienne. Nous avons noté au cours de cette étude que l'application directe de la limitation mise en place dans le cadre des lois de conservation scalaires, aux variables physiques des systèmes mono-dimensionnels étudiés provoquait l'apparition d'oscillations parasites. En conséquence, une procédure de limitation basée sur les variables caractéristiques a été développée. Dans le cas de la dynamique des gaz, les flux numériques ont été construits afin que le système satisfasse une inégalité entropique globale. Fort de l'expérience acquise, nous avons appliqué la discrétisation GD mise en place aux équations bi-dimensionnelles de la dynamique des gaz, écrites dans un formalisme Lagrangien total. Dans ce cadre, le domaine de référence est fixe. Cependant, il est nécessaire de suivre l'évolution temporelle de la matrice jacobienne associée à la transformation Lagrange-Euler de l'écoulement, à savoir le tenseur gradient de déformation. Dans le travail présent, la transformation résultant de l'écoulement est discrétisée de manière continue à l'aide d'une base Éléments Finis. Cela permet une approximation du tenseur gradient de déformation vérifiant l'identité essentielle de Piola. La discrétisation des lois de conservation physiques sur le volume spécifique, le moment et l'énergie totale repose sur une méthode Galerkin discontinu. Le schéma est construit de sorte à satisfaire de manière exacte la loi de conservation géométrique (GCL). Dans le cas du schéma d'ordre trois, le champ de vitesse étant quadratique, la géométrie doit pouvoir se courber. Pour ce faire, des courbes de Bézier sont utilisées pour la paramétrisation des bords des cellules du maillage. Nous illustrons la robustesse et la précision des schémas mis en place à l'aide d'un grand nombre de cas tests pertinents, ainsi que par une étude de taux de convergence.

English

The intent of the present work was the development of a high-order discontinuous Galerkin scheme for solving the gas dynamics equations written under total Lagrangian form on two-dimensional unstructured grids. To achieve this goal, a progressive approach has been used to study the inherent numerical difficulties step by step. Thus, discontinuous Galerkin schemes up to the third order of accuracy have firstly been implemented for the one-dimensional and two-dimensional scalar conservation laws on unstructured grids. The main feature of the presented DG scheme lies on the use of a polynomial Taylor basis. This particular choice allows in the two-dimensional case to take into general unstructured grids account in a unified framework. In this frame, a vertex-based hierarchical limitation which preserves smooth extrema has been implemented. A generic form of numerical fluxes ensuring the global stability of our semi-discrete discretization in the L_2 norm has also been designed. Then, this DG discretization has been applied to the one-dimensional system of conservation laws such as the acoustic system, the shallow-water one and the gas dynamics equations system written in the Lagrangian form. Noticing that the application of the limiting procedure, developed for scalar equations, to the physical variables leads to spurious oscillations, we have described a limiting procedure based on the characteristic variables. In the case of the one-dimensional gas dynamics case, numerical fluxes have been designed so that our semi-discrete DG scheme satisfies a global entropy inequality. Finally, we have applied all the knowledge gathered to the case of the two-dimensional gas dynamics equation written under total Lagrangian form. In this framework, the computational grid is fixed, however one has to follow the time evolution of the Jacobian matrix associated to the Lagrange-Euler flow map, namely the gradient deformation tensor. In the present work, the flow map is discretized by means of continuous mapping, using a finite element basis. This provides an approximation of the deformation gradient tensor which satisfies the important Piola identity. The discretization of the physical conservation laws for specific volume, momentum and total energy relies on a discontinuous Galerkin method. The scheme is built to satisfying exactly the Geometric Conservation Law (GCL). In the case of the third-order scheme, the velocity field being quadratic we allow the geometry to curve. To do so, a Bezier representation is employed to define the mesh edges. We illustrate the robustness and the accuracy of the implemented schemes using several relevant test cases and performing rate convergences analysis.

Introduction

My thesis took place both in the CELIA laboratory (Centre Lasers Intenses et Applications) at Bordeaux University and at the CEA-CESTA (Commissariat à l'Energie Atomique et aux énergies alternatives - Centre d'Etudes Scientifiques et Techniques d'Aquitaine) research center, under the co-supervision of Dr. Pierre-Henri Maire and Pr. Rémi Abgrall.

The research covered during the three years of this Ph.D was part of the huge project willing to domesticate the energy generated by the nuclear fusion. One of the possible means to ignite this extreme physical phenomenon relies on the the use of high power laser facility. This approach is called the Inertial Confinement Fusion (ICF), stemming from the hydrodynamic confinement obtained by high velocity implosion of fusible material. The physical phenomena involved in the implosion of an ICF target are plentiful and extremely complex, we give hereafter a non-exhaustive list of such phenomena:

- interaction between the LASER and the target and creation of a plasma,
- energy transport by electronic conduction and by radiations,
- material ablation,
- formation of very intense shock waves and acceleration of the target,
- focalization of the material at the target center,
- propagation of a diverging thermonuclear wave.

It is very difficult to study these physical processes experimentally, the experiments being very expensive and the results quite difficult to analyze. This is one of the reason why the numerical simulation is so important in nowadays physical research, so is the numerical analysis in this ICF case of the compressible gas dynamics equations.

The hydrodynamic flow resulting from the implosion of an ICF target is a source of very intense shock and rarefaction waves. The size of the domain occupied by the fluid strongly varies over time. Furthermore, the target being composed of several distinct materials, the type of flow generally contains several interfaces that need to be followed with accuracy during the implosion. All these constraints incite the use of the Lagrangian description instead of the Eulerian one. In the Eulerian formalism, the fluid dynamics conservation laws are written using a fixed reference frame whereas in the Lagrangian description they are written through the use of a moving reference frame that follows the fluid motion. Lagrangian representation is particularly well adapted to describe the time evolution of fluid flows contained in regions undergoing large shape changes due to strong compressions or expansions. Moreover, in this representation, there is no mass flux across the boundary surface of a control volume moving with the fluid velocity. Therefore, Lagrangian formalism provides a natural framework to accurately track material

interfaces present in multi-material compressible fluid flows. It also avoids the accuracy loss of the numerical schemes due to the numerical diffusion inherent to the advection terms present in the Eulerian formalism.

Concerning the purpose of my thesis, the subject has been motivated by the work of Pierre-Henri Maire on the development of cell-centered Lagrangian schemes. His work on sub-cell force-based discretization for cell-centered Lagrangian hydrodynamics on polygonal grids, or in the frame of Arbitrary Lagrangian-Eulerian (ALE) method and Reconnection Arbitrary Lagrangian-Eulerian method (ReALE), with an extension to multi-material flows using Volume Of Fluid (VOF) and Moment Of Fluid (MOF) interface reconstruction has permitted significant scientific innovations and progresses in the area of the numerical simulation related to the ICF physics. This scientific research, done with researchers as Rémi Abgrall, Mikhail Shashkov, Raphaël Loubère, Jérome Breil, Stéphane Galéra and others, led to the publication of significant articles in international scientific journals, see [75, 76, 78, 74, 73, 80, 38, 79]. Thus, the aim of this thesis is to investigate the possibility of the extension of the cell-centered Lagrangian scheme developed by Pierre-Henri Maire at the CELIA laboratory to a scheme with a higher order of accuracy. To this end, we have chosen to assess the capability of the discontinuous Galerkin discretization.

The discontinuous Galerkin (DG) methods are locally conservative, stable and high-order accurate. They represent one of the most promising current trends in computational fluid dynamics [26, 27]. They can be viewed as a natural high-order extension of the classical Finite Volume methods. This extension is constructed by means of a local variational formulation in each cell, which makes use of a piecewise polynomial approximation of the unknowns.

The intent of this work was to present a high-order discontinuous Galerkin scheme for solving the gas dynamics equations written under total Lagrangian form on two-dimensional unstructured grids. Thus, it can be seen as the natural extension of the research presented by Raphël Loubère during his Ph.D thesis, supervised by Rémi Abgrall and Jean Ovadia [70]. To achieve the implementation of a high-order DG scheme solving Lagrangian hydrodynamics, we have developed a progressive approach to study the inherent numerical difficulties step by step. Thus, the remaining of this document is organized as follows.

In Chapter 1, we firstly describe discontinuous Galerkin methods up to third-order for one and two-dimensional scalar conservation laws on general unstructured grids. The main feature of the presented DG scheme lies on the use of a polynomial Taylor basis [71]. This particular choice allows in the two-dimensional case to take into general unstructured grids account in a unified framework. In this frame, a vertex-based hierarchical limitation which preserves smooth extrema has been implemented [60, 109]. A generic form of numerical fluxes ensuring the global stability of our semi-discrete discretization in the L_2 norm has also been designed, following [52, 24, 98, 49].

Then, we present in Chapter 2 the application of this DG discretization to the one-dimensional system of conservation laws such as the acoustic system, the shallow-water one and the gas dynamics equations system written in the Lagrangian form. Noticing that the application of the limiting procedure, developed for scalar equations, to the physical variables leads to spurious oscillations, we have described a limiting procedure based on the characteristic variables following [29]. In the case of the one-dimensional gas dynamics case, numerical fluxes have been designed so that our semi-discrete DG scheme satisfies a global entropy inequality.

In Chapitre 3, we start by briefly recalling the evolution of the researches based on the implementation of numerical schemes for the Lagrangian hydrodynamics, as well as the important concepts related to kinematics of fluid motion such as the Lagrangian and Eulerian formalisms. Then, we concisely present the classical Finite Volume EUCCLHYD scheme (Explicit Unstructured Lagrangian HYDrodynamics) [76], which has been the corner stone of our DG formulation. We emphasize that in its first-order version, our DG method naturally recovers the first-order EUCCLHYD scheme.

Finally, in Chapter 4, we apply all the knowledge gathered to the case of the two-dimensional gas dynamics equation written under total Lagrangian form. First, we present the general framework of the discontinuous Galerkin discretization of the gas dynamics equations in the Lagrangian formalism, written using the Lagrangian coordinates. In this framework, the computational grid is fixed, however one has to follow the time evolution of the Jacobian matrix associated to the Lagrange-Euler flow map, namely the gradient deformation tensor. The flow map is discretized by means of continuous mapping, using a finite element basis. This provides an approximation of the deformation gradient tensor which satisfies the important Piola identity. The discretization of the physical conservation laws for specific volume, momentum and total energy relies on a discontinuous Galerkin method. The scheme is built to satisfying exactly the Geometric Conservation Law (GCL). At the end, the particular cases of the second-order and third-order schemes are detailed. In this last case, the velocity field being quadratic we allow the geometry to curve. To do so, a Bezier representation is employed to define the mesh edges. We illustrate the robustness and the accuracy of the implemented schemes using several relevant test cases and performing rate convergences analysis.

Our work during these three years on the discontinous Galerkin discretization in the frame of scalar conservation laws and the Lagrangian hydrodynamics led to the publication of two articles in international journals and one in preparation.

Publications

- F. Vilar, P.-H. Maire and R. Abgrall, *Third order Cell-Centered DG scheme for Lagrangian hydrodynamics on general unstructured Bezier grids.* Journal of Computational Physics. Article in preparation.
- F. Vilar, *Cell-Centered Discontinuous Galerkin discretization for two-dimensional Lagrangian hydrodynamics.* Computers and Fluids, 64:64-73, 2012. Available from `http://www.celia.u-bordeaux1.fr/~vilar/Telechargement/CCDG-2012.pdf`
- F. Vilar, P.-H. Maire and R. Abgrall, *Cell-centered discontinuous Galerkin discretizations for two-dimensional scalar conservation laws on unstructured grids and for one-dimensional Lagrangian hydrodynamics.* Computers and Fluids, 46(1):498-504, 2011. Available from `http://www.celia.u-bordeaux1.fr/~vilar/Telechargement/CCDG-2010.pdf`

Contents

Abstracts **iii**

Introduction **v**

1 High-order discretizations of scalar conservation laws **1**
1.1 One-dimensional scalar conservation laws . . . 2
1.1.1 First-order Finite Volume discretization . . . 2
1.1.2 First-order numerical results . . . 4
1.1.3 Second-order Finite Volume discretization . . . 6
1.1.4 Slope limiters . . . 9
1.1.5 Numerical results . . . 11
1.1.6 Discontinuous Galerkin discretization . . . 14
1.1.7 Numerical flux and L_2 stability . . . 18
1.1.8 Taylor basis . . . 20
1.1.9 Slope limiters for high-order DG methods . . . 22
1.1.10 Numerical results . . . 27
1.2 Two-dimensional scalar conservation laws . . . 32
1.2.1 Discontinuous Galerkin discretization . . . 32
1.2.2 Numerical flux and L_2 stability . . . 34
1.2.3 Two-dimensional Taylor basis . . . 39
1.2.4 Two-dimensional hierarchical vertex-based slope limiter . . . 41
1.2.5 Numerical results for two-dimensional scalar conservations laws . . . 47

2 High-order DG discretization of one-dimensional systems of conservation laws **57**
2.1 Linear acoustics . . . 58
2.1.1 DG discretization of the linear acoustics . . . 58
2.1.2 Linear Riemann problem and numerical fluxes for linear acoustics . . . 59
2.1.3 Limiting procedure based on the characteristic variables . . . 60
2.1.4 Numerical results for linear acoustics . . . 61
2.2 Shallow water equations . . . 64
2.2.1 Governing equations . . . 64
2.2.2 Lagrangian form of the shallow water equations . . . 64
2.2.3 DG discretization of the shallow water equations . . . 66
2.2.4 Approximate Riemann solver and numerical fluxes for the shallow water equations . . . 67
2.2.5 Limiting procedure based on the characteristic variables . . . 69
2.2.6 Numerical results . . . 70
2.3 Lagrangian gas dynamics . . . 73
2.3.1 Discontinuous Galerkin discretization of Lagrangian gas dynamics equations 74

2.3.2 Taylor basis modification for Lagrangian gas dynamics 75
2.3.3 Entropy inequality and numerical fluxes 76
2.3.4 Limiting procedure based on the characteristic variables for the Lagrangian gas dynamics equations 79
2.3.5 Numerical results 81

3 Cell-centered Finite Volume Lagrangian scheme on moving mesh 87
3.1 Cell-Centered Lagrangian scheme overview 87
3.2 Lagrangian and Eulerian descriptions 92
3.2.1 Notations 92
3.2.2 Material time derivative 93
3.2.3 Deformation gradient tensor 94
3.2.4 Transformation formulas 95
3.2.5 Reynolds transport formula 96
3.2.6 Eulerian and Lagrangian form of the gas dynamics equations 98
3.3 First-order cell-centered Lagrangian scheme: EUCCLHYD 102
3.3.1 Control volume form of the gas dynamics equations 102
3.3.2 Compatible cell-centered discretization 102
3.3.3 Sub-cell force-based discretization 107
3.3.4 Thermodynamic consistency of the semi-discrete scheme 109
3.3.5 Summary of the semi-discrete scheme 110
3.3.6 Node-centered solver for the grid velocity 111
3.3.7 Second-order MUSCL scheme 116
3.3.8 Time step control 119

4 High-order DG Cell-centered Lagrangian schemes 121
4.1 DG discretization general framework 122
4.1.1 Variational formulations 124
4.1.2 Deformation gradient tensor discretization 126
4.1.3 Entropic analysis 130
4.1.4 Limiting procedure based on the characteristic variables 131
4.2 Second-order DG scheme 135
4.2.1 Deformation gradient tensor discretization 135
4.2.2 Variational formulations 138
4.2.3 Nodal solvers 142
4.2.4 Numerical results 144
4.3 Third-order DG scheme 159
4.3.1 Geometric consideration 159
4.3.2 Deformation gradient tensor discretization 161
4.3.3 Variational formulations 163
4.3.4 Nodal solvers 166
4.3.5 Deformed initial mesh 170
4.3.6 Numerical results 172

5 Conclusion and perspectives 183

A Notations and reminders related to vectors and tensors **185**
A.1 Introduction to second-order tensors . . . 185
A.1.1 Definitions . . . 185
A.1.2 Properties of the dyadic product of two vectors . . . 187
A.1.3 Inner product of second-order tensors . . . 187
A.2 Eigenvalues and eigenvectors of a tensor . . . 188
A.3 Tensor analysis . . . 188
A.3.1 Derivative of tensor with respect to a scalar . . . 188
A.3.2 Derivative of scalar valued functions of vectors . . . 188
A.3.3 Derivative of vector valued functions of vectors . . . 189
A.3.4 Derivative of scalar valued functions of second-order tensors . . . 189
A.3.5 Derivative of the determinant of a tensor . . . 189
A.3.6 Divergence of a vector and a tensor . . . 190
A.3.7 Tensor identities . . . 190
A.4 Integral transformation formulas . . . 190

Bibliography **193**

Chapter 1

High-order discretizations of scalar conservation laws

The aim of this chapter is to introduce high-order discretizations for linear and non-linear scalar conservation laws. First, in Section 1.1, we start by recalling the classical second-order Finite Volume scheme devoted to the numerical discretization of one-dimensional scalar conservation laws, which can be found in many textbooks [65, 42, 31]. In this well known approach, the numerical solution is described by a piecewise constant approximation. The time rate of change of the cell-averaged value is balanced by the difference of the cell interfaces numerical fluxes. At each cell interface the numerical flux is expressed in terms of the left and right cell-averaged values on both sides of the interface by means of an approximate Riemann solver. The second-order extension is achieved by performing a piecewise linear reconstruction of the numerical solution from the cell-averaged values. The numerical flux at the cell interface is computed by utilizing the extrapolated values on both sides of the interface coming from the piecewise linear reconstruction. The monotonicity of the piecewise linear reconstruction is achieved thanks to a slope limiter of minmod type. The second-order time discretization is obtained employing a two-step predictor-corrector scheme.

To achieve a higher order accuracy, we pursue by presenting the discontinuous Galerkin (DG) discretization of the one-dimensional scalar conservation laws. Theoretically, DG methods allow to reach any arbitrary order of accuracy. Since their introduction in 1973 by Reed and Hill [90] in the framework of neutron transport (steady state linear hyperbolic equations), the discontinuous Galerkin methods turn out to be particularly well adapted to solve system of scalar conservation laws with a high order of accuracy. A major development of the DG method was carried out by Cockburn, Shu *et al.* in a series of papers [28, 27, 26, 25, 29], in which they established a framework to easily solve nonlinear time dependent hyperbolic conservation laws (1.1) using explicit, nonlinear but stable high-order Runge-Kutta time discretizations [99] and DG discretization in space with exact or approximate Riemann solvers as interface fluxes. A total variation bounded (TVB) limiter [97] is used to achieve non-oscillatory properties for strong shocks. These schemes are termed RKDG methods. More recently, Zhang and Shu [110] have constructed uniformly high-order accurate schemes satisfying a strict maximum principle for scalar conservation laws. They proposed a general framework to construct limiter for DG methods with first order Euler forward in time discretization solving one-dimensional scalar conservation laws. Strong stability preserving high-order time discretizations are employed to assure preservation of the maximum principle. A pedagogical presentation of the DG discretization can be found in [24, 98]. A numerical study for the performance of the RKDG method based on different numerical fluxes is performed in [88]. The extension of the local discontin-

uous Galerkin methods to high-order time-dependent partial differential equations is described in the review paper [108] wherein one can find a complete set of references on this topic. It is worth mentioning that the DG methods combine noteworthy advantages from the Finite Element methods and the Finite Volume ones. On the one hand, the high accuracy comes from the high-order polynomial approximation of the solution. This approximation results from a local variational formulation on each cell. On the other hand, this polynomial description of the solution being discontinuous from one cell to another, the fluxes at the interfaces are evaluated by means of numerical flux functions similar to the ones used in the Finite Volume method frame. Thus, the DG discretization can be viewed as an extension of the Finite Volume method to an arbitrary order of accuracy, wherein a piecewise polynomial approximation of the unknown is used.

1.1 One-dimensional scalar conservation laws

This section is devoted to the description of numerical methods for solving the following problem. Let $u = u(x,t)$, for $x \in C$, where $C \subset \mathbb{R}$, and $t \in [0,T]$, be the solution of the following one-dimensional scalar conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \tag{1.1a}$$

$$u(x,0) = u^0(x), \tag{1.1b}$$

where u^0 is the initial data and $f(u)$ is the flux function. For the subsequent discretizations, let us introduce the following notation. Let $\bigcup_{i=1,I} C_i$ be a partition of the computational domain C. Here, $C_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ denotes a generic computational cell of size Δx_i. We also introduce a partition of the time domain $0 = t^1 < t^2 < ... < t^n < ... < t^N = T$ and the time step $\Delta t^n = t^{n+1} - t^n$.

1.1.1 First-order Finite Volume discretization

The Finite Volume discretization is based on the integration of the concerned equation (1.1a) in space and time on $C_i \times [t^n, t^{n+1}]$

$$\int_{t^n}^{t^{n+1}} \int_{C_i} \left(\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} \right) \mathrm{d}x \, \mathrm{d}t = 0.$$

Immediately, this equation rewrites

$$\int_{C_i} \left(u(x,t^{n+1}) - u(x,t^n) \right) \mathrm{d}x + \int_{t^n}^{t^{n+1}} \left(\overline{f}(u(x_{i+\frac{1}{2}},t)) - \overline{f}(u(x_{i-\frac{1}{2}},t)) \right) \mathrm{d}t = 0.$$

Let us introduce $u_i^n = \frac{1}{\Delta x_i} \int_{C_i} u(x,t^n)\,\mathrm{d}x$ the mean value of our unknown over the cell C_i at time t^n, and $\overline{F}_{i+\frac{1}{2}} = \frac{1}{\Delta t^n} \int_{t^n}^{t^{n+1}} \overline{f}(u(x_{i+\frac{1}{2}},t))\,\mathrm{d}t$ with $\overline{f}(u)$ the numerical flux. The Finite Volume scheme obtained writes

$$\Delta x_i \, (u_i^{n+1} - u_i^n) + \Delta t^n \, (\overline{F}_{i+\frac{1}{2}} - \overline{F}_{i-\frac{1}{2}}) = 0. \tag{1.2}$$

The principle of the Finite Volume methods, and from a more general point of view of the discontinuous Galerkin methods (see Section 1.1.6), is the use of a piecewise approximation of the unknown over the cells of the mesh. More precisely, the piecewise approximation of the

unknown would be discontinuous at the interfaces between cells, for example at interface $x_{i+\frac{1}{2}}$. These discontinuities and the conservation property over the domain impose to the single value $\overline{F}_{i+\frac{1}{2}}$ to be shared by the two cells C_i and C_{i+1} surrounding the interface $x_{i+\frac{1}{2}}$ and so to be calculated from the values of the two cells

$$\overline{F}_{i+\frac{1}{2}} = \frac{1}{\Delta t^n} \int_{t^n}^{t^{n+1}} h(u_i, u_{i+1})\, \mathrm{d}t,$$

where h is the numerical flux characterizing the approximation. The evaluation of this time integral can be performed exactly. Another alternative is the use of a quadrature rule in accordance to the expecting accuracy order. For the first-order, the simple left rectangle rule will be used to get an explicit method

$$\overline{F}_{i+\frac{1}{2}} = h(u_i^n, u_{i+1}^n).$$

The approximation flux h is consistent with the physical flux $f(u)$ if it satisfies the property

$$h(\overline{u}, \overline{u}) = f(\overline{u}). \tag{1.3}$$

With this property, the numerical flux reduces to $f(\overline{u})$ in the case of a constant solution where $u(x,t) = \overline{u}$. We also make the assumption that the numerical flux function h is a Lipschitz continuous function with respect to its arguments at point $\overline{u}$, *i.e.*, $\exists\, K > 0$ such as

$$|h(v,w) - f(\overline{u})| \leq K \max(|v - \overline{u}|, |w - \overline{u}|), \tag{1.4}$$

for all v and w with $|v - \overline{u}|$ and $|w - \overline{u}|$ small enough. This last property implies that when the two arguments of the numerical flux function tend toward the value $\overline{u}$ then h tends toward $f(\overline{u})$ smoothly. Moreover, we suppose that the numerical flux function, h, is non-decreasing in the first argument and non-increasing in the second argument. These assumptions leads to a numerical flux which belongs to the class of monotone flux, refer to [42]. In what follows, we shall use the well known Local Lax-Friedrichs flux which satisfies the aforementioned properties. This numerical flux is defined by

$$h(u,v) = \frac{1}{2}[f(u) + f(v) - \beta\,(v - u)]. \tag{1.5}$$

Here, $\beta > 0$ is a parameter which has the dimension of a velocity. Introducing the characteristic speed as $a(u) = \frac{df}{du}$ leads to define the Lax-Friedrichs flux by $\beta = \max\left(|a(w)|\right)$ for $w \in [\inf(u^0(x)), \sup(u^0(x))]$ whereas the Local Lax-Friedrichs flux is given by

$$\beta = \max\left(|a(w)|\right) \text{ for } w \in [\min(u,v), \max(u,v)].$$

In the case of a convex flux function, β expression boils down to $\beta = \max\left(|a(u)|, |a(v)|\right)$. We notice that in the linear case, *i.e.*, $f(u) = au$, where a is the constant convection speed, the Local Lax-Friedrichs flux is nothing but the classical upwind flux since in this case $\beta = |a|$.
Finally, we recall that the stability of the explicit Finite Volume scheme (1.2) is ensured provided that the time step satisfies the following Courant-Friedrichs-Levy (CFL) condition [42]

$$\Delta t^n \leq \min_i \frac{\Delta x_i}{|a(u_i^n)|}. \tag{1.6}$$

In this section, the first-order Finite Volume discretization of the one-dimensional scalar conservation law (1.1) has been briefly recalled. It appears to be very simple to implement. In the next subsection, we are going to verify the robustness of this scheme and to assess its accuracy on different test cases.

1.1.2 First-order numerical results

To assess the accuracy of a numerical scheme, we evaluate the difference between the numerical solution and the analytical solution employing a sequence of successively refined grids characterized by a uniform mesh spacing $h = \Delta x$. Let u_h^n be the numerical solution obtained from the numerical scheme at time t^n, *i.e.*, $u_h^n(x) = u_i^n$ for $x \in C_i$. Denoting by $u_{ex} = u_{ex}(x, t^n)$ the analytical solution, we define the numerical error at time t^n as

$$E_{L_\alpha} = ||u_h^n - u_{ex}||_{L_\alpha}, \tag{1.7}$$

where $\|\cdot\|_{L_\alpha}$ is a given norm characterized by the subscript α. We will assess the numerical errors for the three following classical norms: L_1, L_2 and L_∞. For these norms the corresponding numerical errors read

$$E_{L_1} = \sum_i \int_{C_i} |u_i^n - u_{ex}(x, t^n)| \, \mathrm{d}x. \tag{1.8}$$

$$E_{L_2} = \sqrt{\sum_i \int_{C_i} [u_i^n - u_{ex}(x, t^n)]^2 \, \mathrm{d}x}. \tag{1.9}$$

$$E_{L_\infty} = \max_i [\max_{x \in C_i} (|u_i^n - u_{ex}(x, t^n)|)]. \tag{1.10}$$

For a grid characterized by the constant mesh spacing Δx, we postulate that the numerical error E_{L_α} has the following behavior

$$E_{L_\alpha}^{\Delta x} = D \, (\Delta x)^{q_{L_\alpha}}, \tag{1.11}$$

where D is a non-negative constant independent of Δx and q_{L_α} is a real which corresponds to the rate of convergence of the numerical scheme. The value $q_{L_\alpha} = 1$ characterizes a first-order numerical scheme, for which the error decreases linearly with respect to the mesh spacing. To compute the rate of convergence, we evaluate the numerical errors corresponding to two grids characterized by Δx_1 and Δx_2 with $\Delta x_1 > \Delta x_2$. Using the relation (1.11) yields

$$q_{L_\alpha} = \frac{\log(E_{L_\alpha}^{\Delta x_1}) - \log(E_{L_\alpha}^{\Delta x_2})}{\log(\Delta x_1) - \log(\Delta x_2)}. \tag{1.12}$$

For the numerical applications, we consider two classical test cases. The computational domain is $C = [0, 1]$ and we prescribe periodic boundary conditions. The first test case corresponds to a linear conservation law characterized by the initial condition $u^0(x) = \sin(2\pi x)$ and the linear flux $f(u) = au$, where the constant velocity a is given by $a = 1$. For this case, the analytical solution reads $u_{ex}(x, t) = u^0(x - at)$. The second test case corresponds to a non-linear conservation law characterized by the initial condition $u^0(x) = \sin(2\pi x)$, the non-linear flux $f(u) = \frac{u^2}{2}$ and the characteristic speed $a(u) = u$. With this flux, the conservation law (1.1) is called the Burgers equation. In this case, it is possible to compute the analytical solution up to a critical time by means of the method of characteristics. This critical time, t_c, is the time at which the characteristics first cross and a shock forms as it can be seen in Figure 1.1, refer to [65]. For the initial data under consideration the critical time is given by $t_c = \frac{1}{2\pi}$. Let us point out that before this critical time, the analytical solution is smooth.
We have displayed in Table 1.1 the convergence results for the first test case. The results show clearly a first-order convergence rate.

	L_1		L_2		L_∞	
Δx	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{100}$	6.09E-2	0.96	6.77E-2	0.97	9.69E-2	0.97
$\frac{1}{200}$	3.12E-2	0.98	3.46E-2	0.98	4.96E-2	0.98
$\frac{1}{400}$	1.58E-2	0.99	1.75E-2	0.99	2.51E-2	0.99
$\frac{1}{800}$	7.94E-3	1.00	8.82E-3	1.00	1.26E-2	1.00
$\frac{1}{1600}$	3.98E-3	-	4.42E-3	-	6.33E-3	-

Table 1.1: Rate of convergence for the first-order FV scheme in the case of linear advection with the smooth initial condition $u^0(x) = \sin(2\pi x)$ on the $[0, 1]$ domain, at time $t = 1$ with a CFL= 0.5.

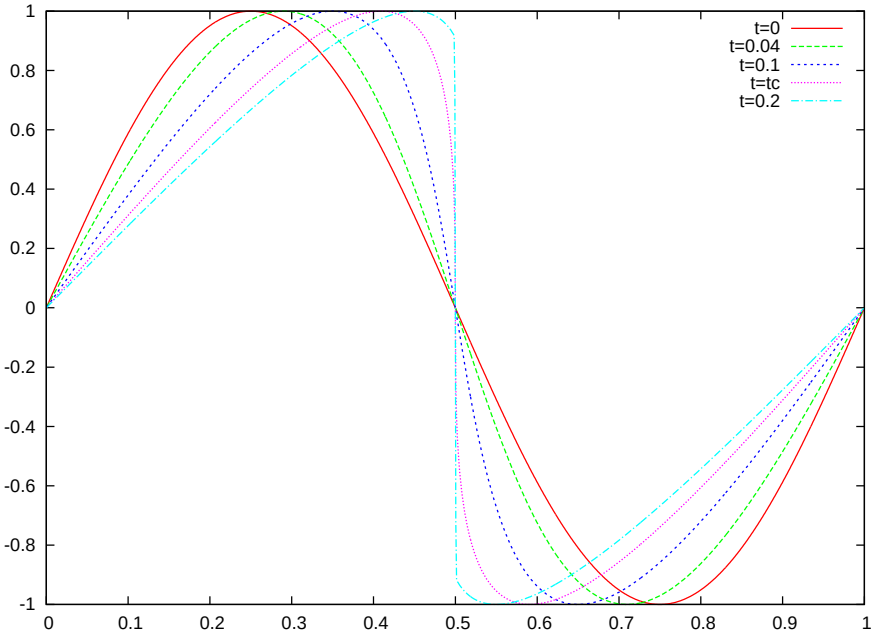


Figure 1.1: Shock formation for the Burgers equation for the smooth initial condition $u^0(x) = \sin(2\pi x)$. The numerical solutions have been obtained using the first-order scheme with a very fine grid.

	L_1		L_2		L_∞	
Δx	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{200}$	6.08E-3	1.00	8.49E-3	0.98	4.16E-2	0.97
$\frac{1}{400}$	3.05E-3	1.00	4.31E-3	0.99	2.20E-2	0.98
$\frac{1}{800}$	1.53E-3	1.00	2.17E-3	0.99	1.13E-2	0.99
$\frac{1}{1600}$	7.65E-4	1.00	1.09E-3	1.00	5.73E-2	1.00
$\frac{1}{3200}$	3.83E-4	-	5.46E-4	-	2.87E-3	-

Table 1.2: Rate of convergence for the first-order Finite Volume scheme used in the case of Burgers equation with the smooth initial condition $u^0(x) = \sin(2\pi x)$ on the $[0, 1]$ domain, at time $t = \frac{t_c}{2}$ with a CFL= 0.5.

We have also performed the convergence analysis for the non-linear Burgers problem. The results are displayed in Table 1.2. As expected, we clearly observe a first-order rate of convergence. Let us point out that to assess more precisely the order of the numerical scheme, we keep refining the mesh until asymptotic regime is reached.

The results presented in the above Tables show that the Finite Volume scheme previously described has a first-order accuracy. From a general point of view, the first-order schemes are generally robust and easy to implement. The problem is the low accuracy of these methods. After a while, the numerical dissipation is so huge that the obtained solution does not correspond anymore to the expected solution. To cure this problem, we derive in the next section a classical second-order extension of the present Finite Volume discretization.

1.1.3 Second-order Finite Volume discretization

In the previous section, the first-order Finite Volume scheme wherein a piecewise constant approximation of the solution is used has been presented. In this case, the numerical flux $h(u_L, u_R)$ is computed by means of the mean values u_L and u_R of the solution on the cells located on both sides of the interface. Here, the aim is to decrease the dissipation and to increase the accuracy of the numerical solution by constructing a second-order extension. The method consists in computing a piecewise linear approximation of the solution over each cell. In the numerical flux evaluation, instead of defining the left and right states as the averaged cell values, one computes u_L and u_R as the extrapolated values at the interface, using the piecewise linear approximation of the solution, on both sides of the interface. This procedure leads to decrease the jump $(u_R - u_L)$ in the smooth parts of the solution and thus decrease the viscous contribution of the numerical flux which is proportional to $(u_R - u_L)$. This approach yields a second-order space discretization. The global second-order accuracy requires also a second-order time discretization by means of a classical predictor-corrector scheme.
Let us point out that there exists a huge literature about second-order Finite Volume schemes for conservation laws. The interested reader could refer to the pioneering works of van Leer [101] and Kolgan [102, 58] and also to text books [42, 43, 65, 6].

Piecewise linear reconstruction

Being given u_i^n, the mean value of the solution on the cell C_i, we define the piecewise linear approximation as

$$\widetilde{u}_i(x) = u_i^n + \delta_i\,(x - x_i), \quad x \in C_i \tag{1.13}$$

where $x_i = \frac{1}{2}(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}})$ is the midpoint of the cell and δ_i represents the slope of the solution. We notice that this linear reconstruction is conservative since it conserves the solution mean value, *i.e.*,

$$\frac{1}{\Delta x_i}\int_{C_i} \widetilde{u}_i(x)\,\mathrm{d}x = u_i^n.$$

The slope δ_i is an approximation of the solution derivative $\frac{\partial u}{\partial x}$ inside cell C_i. The slope reconstruction could be obtained from different methods. Here, following [72], we present a slope computation based on a least squares algorithm, which has the advantage of preserving linear fields on non-uniform meshes. The slope δ_i in cell C_i is computed imposing the following conditions

$$\widetilde{u}_i(x_{i-1}) = u_{i-1}^n \quad \text{and} \quad \widetilde{u}_i(x_{i+1}) = u_{i+1}^n,$$

which amounts to seek the best linear fit for (x_{i-1}, u_{i-1}^n), (x_i, u_i^n) and (x_{i+1}, u_{i+1}^n). For this purpose, we minimize the corresponding cost function

$$\begin{aligned} L(\delta_i) &= \frac{1}{2}\left[\widetilde{u}_i(x_{i-1}) - u_{i-1}^n\right]^2 + \frac{1}{2}\left[\widetilde{u}_i(x_{i+1}) - u_{i+1}^n\right]^2. \\ &= \frac{1}{2}\left[u_i^n + \delta_i\,(x_{i-1} - x_i) - u_{i-1}^n\right]^2 + \frac{1}{2}\left[u_i^n + \delta_i\,(x_{i+1} - x_i) - u_{i+1}^n\right]^2. \end{aligned}$$

The derivative of the cost function reads

$$L'(\delta_i) = (x_{i-1} - x_i)\left[u_i^n + \delta_i\,(x_{i-1} - x_i) - u_{i-1}^n\right] + (x_{i+1} - x_i)\left[u_i^n + \delta_i\,(x_{i+1} - x_i) - u_{i+1}^n\right].$$

The minimum is reached for $L'(\delta_i) = 0$. This gives us the expression of the slope

$$\delta_i = \frac{(u_i^n - u_{i-1}^n)(x_i - x_{i-1}) + (u_{i+1}^n - u_i^n)(x_{i+1} - x_i)}{(x_i - x_{i-1})^2 + (x_{i+1} - x_i)^2}. \tag{1.14}$$

In the case of uniform mesh of space step Δx, the slope expression deriving from a least squares algorithm reduces to

$$\delta_i = \frac{u_{i+1}^n - u_{i-1}^n}{2\,\Delta x},$$

which is nothing but the centered approximation of $\frac{\partial u}{\partial x}$. For the computation of the numerical flux $h(u_L, u_R)$ at the interface $x_{i+\frac{1}{2}}$, the left and right states, u_L and u_R, are evaluated as the extrapolated values on both sides of the interface

$$\begin{aligned} u_L &= \widetilde{u}_i(x_{i+\frac{1}{2}}) = u_i^n + \frac{1}{2}\Delta x_i\,\delta_i, \\ u_R &= \widetilde{u}_{i+1}(x_{i+\frac{1}{2}}) = u_{i+1}^n - \frac{1}{2}\Delta x_{i+1}\,\delta_{i+1}. \end{aligned}$$

This extrapolation is depicted on Figure 1.2. We note that if the slope is set to zero, u_L and u_R correspond to the mean values on both sides of the interface.

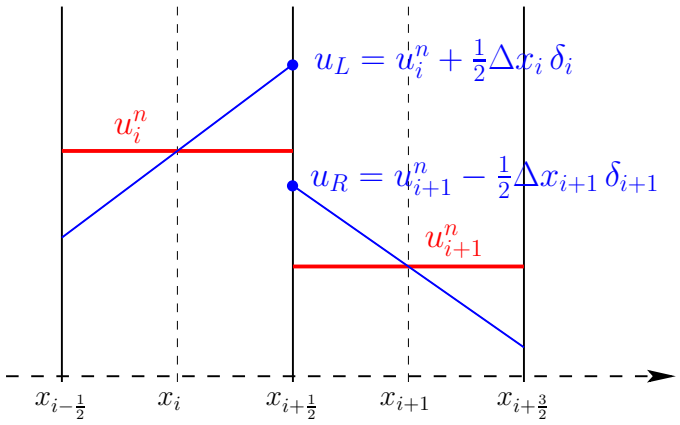


Figure 1.2: Extrapolation values at the interface $x_{i+\frac{1}{2}}$ resulting from the linear reconstructions in cells C_i and C_{i+1}.

In what follows, we identify $u_{i+\frac{1}{2}}^{-,n}$ and $u_{i+\frac{1}{2}}^{+,n}$ to the extrapolated values $\widetilde{u}_i(x_{i+\frac{1}{2}})$ and $\widetilde{u}_{i+1}(x_{i+\frac{1}{2}})$ expressed before.

Time discretization

We recall the Finite Volume scheme introduced in the previous section

$$\Delta x_i \, (u_i^{n+1} - u_i^n) + \Delta t^n \, (\overline{F}_{i+\frac{1}{2}} - \overline{F}_{i-\frac{1}{2}}) = 0.$$

This time, the numerical flux is not computed with the mean values of the function on the cells surrounding the selected interface but with the extrapolated values of the linear reconstruction on the cells located at the interface. The corresponding Finite Volume scheme reads

$$\Delta x_i \, (u_i^{n+1} - u_i^n) + \Delta t^n \left(h(u_{i+\frac{1}{2}}^{-,n}, u_{i+\frac{1}{2}}^{+,n}) - h(u_{i-\frac{1}{2}}^{-,n}, u_{i-\frac{1}{2}}^{+,n}) \right) = 0, \tag{1.15}$$

where $h(u_{i+\frac{1}{2}}^{-,n}, u_{i+\frac{1}{2}}^{+,n})$ refers to the approximation of the time integrated term $\overline{F}_{i+\frac{1}{2}}$. Proceeding like this, the scheme (1.15) would be second-order accurate in space, but only first-order accurate in time. To reach a global second-order accuracy, we have to use at least a second-order quadrature rule in the evaluation of the time integral. Using the midpoint approximation leads to the Predictor-Corrector scheme

$$\Delta x_i \, (u_i^{n+\frac{1}{2}} - u_i^n) + \frac{\Delta t^n}{2} \left(h(u_{i+\frac{1}{2}}^{-,n}, u_{i+\frac{1}{2}}^{+,n}) - h(u_{i-\frac{1}{2}}^{-,n}, u_{i-\frac{1}{2}}^{+,n}) \right) = 0, \qquad \text{predictor} \tag{1.16a}$$

$$\Delta x_i \, (u_i^{n+1} - u_i^n) + \Delta t^n \left(h(u_{i+\frac{1}{2}}^{-,n+\frac{1}{2}}, u_{i+\frac{1}{2}}^{+,n+\frac{1}{2}}) - h(u_{i-\frac{1}{2}}^{-,n+\frac{1}{2}}, u_{i-\frac{1}{2}}^{+,n+\frac{1}{2}}) \right) = 0, \quad \text{corrector} \tag{1.16b}$$

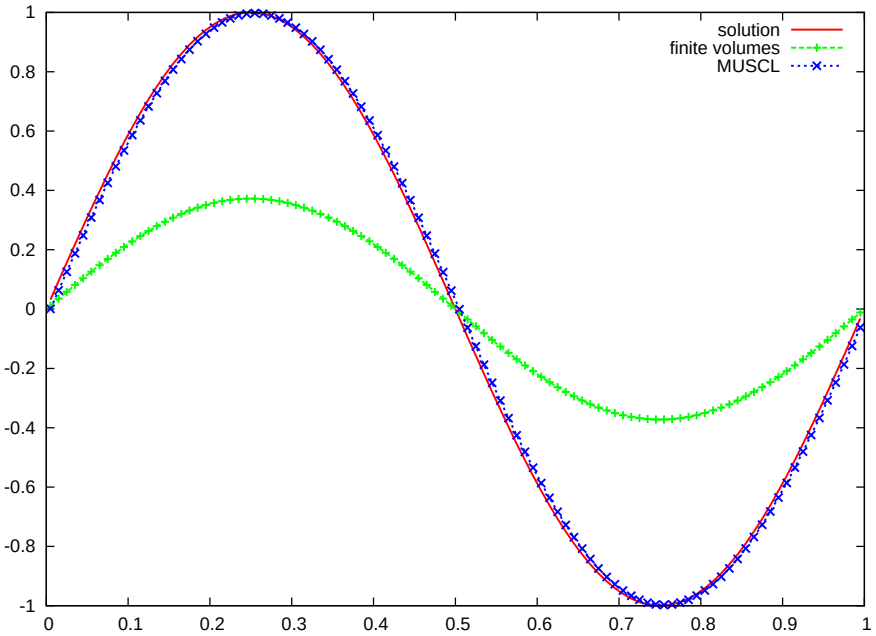


Figure 1.3: Linear advection ($a = 1$) for the initial condition $u^0(x) = \sin(2\pi x)$, $x \in [0, 1]$ with periodic boundaries. Results at the end of ten periods with a CFL=0.5 and a grid with 100 equally spaced cells. Comparison between the analytical solution and the numerical results obtained with the first-order and second-order Finite Volume schemes.

This so called unlimited MUSCL scheme (Monotone Upstream-centered Schemes for Conservation Laws) is second-order accurate and is consequently more accurate than the first-order

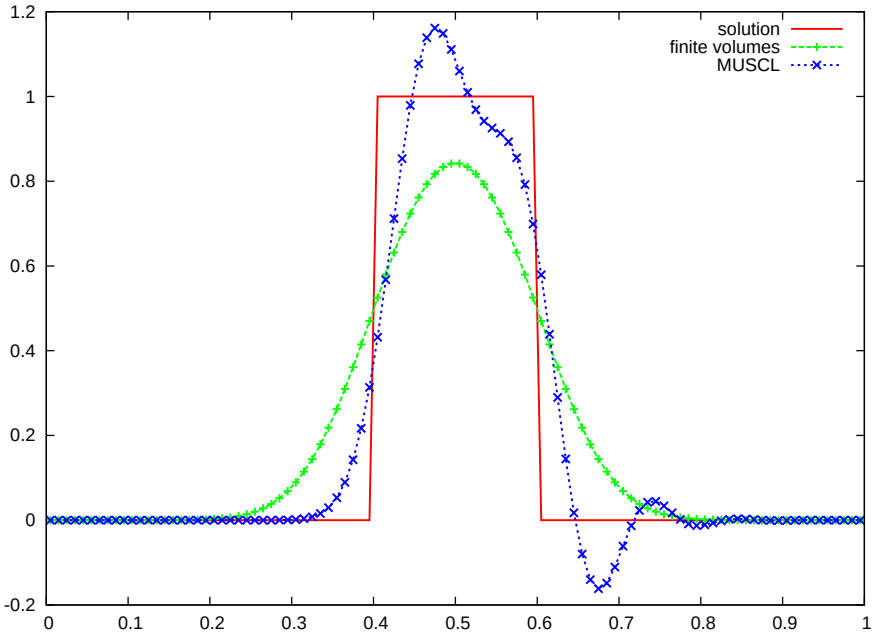


Figure 1.4: Linear advection ($a = 1$) for the discontinuous initial condition $u^0(x) = 1$ if $x \in [0.4, 0.6]$ and null anywhere else, with periodic boundaries. Results at the end of one period with a CFL=0.5 and a grid with 100 equally spaced cells. Comparison between the analytical solution and the numerical results with the Finite Volume scheme and the MUSCL scheme.

Finite Volume scheme presented before, as show Figure 1.3 and Figure 1.4. The numerical flux being computed by means of the extrapolated values at the interface, using the piecewise linear approximation of the solution, the viscous contribution of the numerical flux generating dissipation decreases. However, we clearly observe the apparition of oscillations on the second-order solution in the discontinuous case. To correct this flaw and restore monotonicity, we introduce slope limiters.

1.1.4 Slope limiters

The piecewise linear reconstruction over cell C_i reads

$$\widetilde{u}_i(x) = u_i^n + \delta_i\,(x - x_i),$$

where x_i is the midpoint of the cell C_i and δ_i is the slope obtained by the least squares approach. The modification of the slopes to ensure monotonicity is implemented following [6]. Geometrically speaking, the limitation procedure consists of the minimal change of the slope needed to ensure the following "five-point monotonicity": If $\{u_{i-1}^n, u_i^n, u_{i+1}^n\}$ form a monotonic sequence, then so must the five values $\{u_{i-1}^n, u_{i-\frac{1}{2}}^{+,n}, u_i^n, u_{i+\frac{1}{2}}^{-,n}, u_{i+1}^n\}$. This monotonicity condition applied to the piecewise linear reconstruction leads to a limitation of the slope δ_i. Let us investigate the two different cases wherein the sequence $\{u_{i-1}^n, u_i^n, u_{i+1}^n\}$ is either increasing or decreasing. In the first case, we seek a condition on δ_i imposing that the sequence $\{u_{i-1}^n, u_{i-\frac{1}{2}}^{+,n}, u_i^n, u_{i+\frac{1}{2}}^{-,n}, u_{i+1}^n\}$ is also increasing. Introducing $\alpha_i \in [0, 1]$, this condition amounts to impose

$$u_i^n - \frac{1}{2}\Delta x_i\,\delta_i \geq (1 - \alpha_i)u_i^n + \alpha_i u_{i-1}^n,$$
$$u_i^n + \frac{1}{2}\Delta x_i\,\delta_i \leq (1 - \alpha_i)u_i^n + \alpha_i u_{i+1}^n.$$

Hence, we deduce

$$\delta_i \leq 2\alpha_i \frac{u_i^n - u_{i-1}^n}{\Delta x_i} \quad \text{and} \quad \delta_i \leq 2\alpha_i \frac{u_{i+1}^n - u_i^n}{\Delta x_i}.$$

Consequently, the limited slope $\delta_i^{\lim}$ is defined as

$$\delta_i^{\lim} = \min(\delta_i, 2\alpha_i \frac{u_i^n - u_{i-1}^n}{\Delta x_i}, 2\alpha_i \frac{u_{i+1}^n - u_i^n}{\Delta x_i}),$$

where $\alpha_i \in [0, 1]$.
In the second case, where the sequence $\{u_{i-1}^n, u_i^n, u_{i+1}^n\}$ is decreasing ($\delta_i \leq 0$), the same reasoning yields

$$\delta_i^{\lim} = \max(\delta_i, 2\alpha_i \frac{u_i^n - u_{i-1}^n}{\Delta x_i}, 2\alpha_i \frac{u_{i+1}^n - u_i^n}{\Delta x_i}).$$

The last case to investigate results from a non-monotonic sequence $\{u_{i-1}^n, u_i^n, u_{i+1}^n\}$. Here, we set the slope to zero. For the sake of conciseness, we introduce the minmod function

$$\text{minmod}(x, y, z) = \min(0, \max(x, y, z)) + \max(0, \min(x, y, z)). \tag{1.17}$$

Thus, the limited slope writes

$$\delta_i^{\lim} = \text{minmod}(\delta_i, 2\alpha_i \frac{u_i^n - u_{i-1}^n}{\Delta x_i}, 2\alpha_i \frac{u_{i+1}^n - u_i^n}{\Delta x_i}). \tag{1.18}$$

After limitation, the linear reconstruction on the cell C_i reads

$$\widetilde{u}_i^{\lim}(x) = u_i^n + \delta_i^{\lim}(x - x_i).$$

The limited linear reconstruction is used to compute the extrapolated values at the interfaces needed by the numerical flux. The role of the slope limiter is displayed in Figure 1.5 for $\alpha_i = 1$.

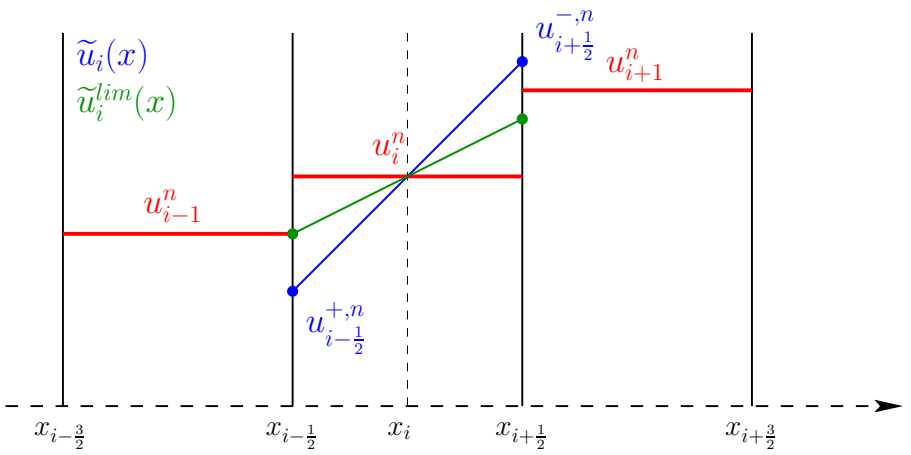


Figure 1.5: Extrapolation values at the interface $x_{i+\frac{1}{2}}$ of the linear reconstructions in the cells C_i and C_{i+1}.

Let us point out that the choice $\alpha_i = 0$ sets the limited slope to zero and thus we recover the first-order scheme. In the numerical applications, we will set $\alpha_i = 1$. This particular value allows to

preserve the linear fields [72]. This is the less diffusive limitation. However, this choice allows for a non-monotonic behavior of the set of values $\{u^n_{i-1}, u^{-,n}_{i-\frac{1}{2}}, u^{+,n}_{i-\frac{1}{2}}, u^n_i, u^{-,n}_{i+\frac{1}{2}}, u^{+,n}_{i+\frac{1}{2}}, u^n_{i+1}\}$. Choosing $\alpha_i = \frac{1}{2}$ yields a more restrictive limiter which is easily seen to lead to the monotonicity of the previous seven-point sequence. In this case, we recover the classical limitation which ensures that the second-order Finite Volume scheme is TVD, that is its total variation is non-increasing as time evolves, refer to [65, 42].
The efficiency of the limitation procedure for $\alpha_i = 1$ is displayed in Figure 1.6 wherein are compared the solutions with and without the use of slope limiter for linear advection.

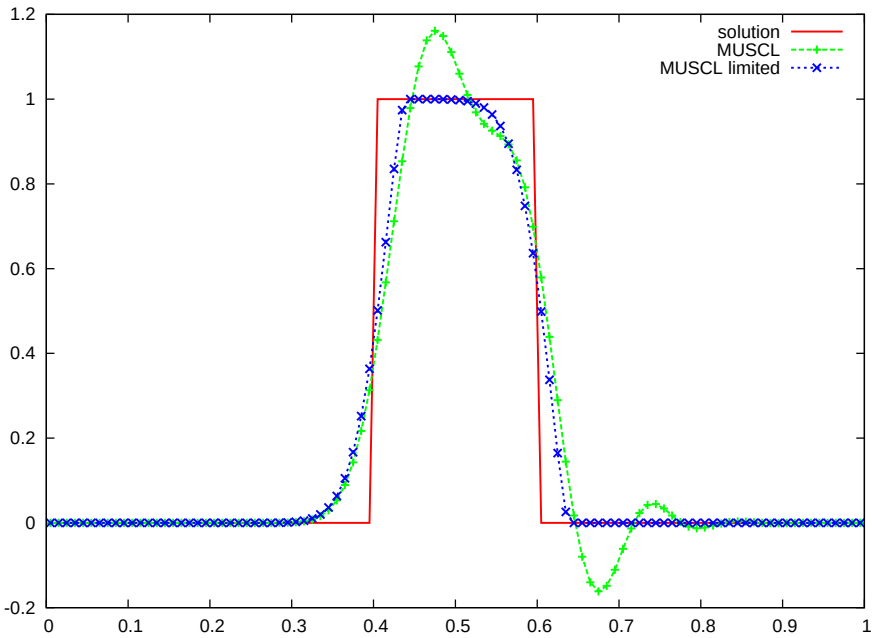


Figure 1.6: Linear advection ($a = 1$) for the discontinuous initial condition $u^0(x) = 1$ if $x \in [0.4, 0.6]$ and null anywhere else, with periodic boundaries, at the end of one period with a CFL=0.5 and a mesh characterized by 100 uniformly spaced cells. Comparison between the analytical solution and the numerical results obtained with the second-order scheme with and without the use of slope limiter.

We observe in Figure 1.6 that all the spurious oscillations have disappeared. In order to confirm that the MUSCL scheme presented in this section is characterized by a second-order accuracy, and that the limitation does not spoil this accuracy, we perform a numerical convergence analysis using the same test cases than before.

1.1.5 Numerical results

We have already observe the gain in accuracy with the linear reconstruction presented in the previous section on Figure 1.3 and Figure 1.4. Now, we perform the same rate convergence analysis than for the Finite Volume scheme, to the linear advection problem, refer to the Table 1.3, and to the non-linear Burgers problem, refer to Table 1.4.
The results presented in these Tables confirm that the MUSCL scheme is indeed second-order accurate. We note in the previous section the apparition of oscillations on the MUSCL scheme solutions, and the necessity of a limitation procedure to force the monotonicity. The efficiency of the previously described slope limiter on the undesirable oscillations has been depicted on Figure 1.6. To assess the accuracy of the MUSCL scheme with limitation, we firstly confront a

	L_1		L_2		L_∞	
Δx	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{100}$	2.01E-3	2.00	2.23E-3	2.00	3.19E-3	2.00
$\frac{1}{200}$	5.01E-4	2.00	5.57E-4	2.00	7.95E-4	2.00
$\frac{1}{400}$	1.25E-4	2.00	1.39E-4	2.00	1.98E-4	2.00
$\frac{1}{800}$	3.13E-5	2.00	3.47E-5	2.00	4.96E-5	2.00
$\frac{1}{1600}$	7.82E-6	-	8.68E-6	-	1.24E-5	-

Table 1.3: Rate of convergence for the second-order MUSCL scheme computed in the case of linear advection with the smooth initial condition $u^0(x) = \sin(2\pi x)$ on the $[0, 1]$ domain, at time $t = 1$ with a CFL= 0.5.

	L_1		L_2		L_∞	
Δx	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{200}$	1.07E-4	1.99	1.34E-4	2.00	3.65E-4	2.05
$\frac{1}{400}$	2.70E-5	2.00	3.36E-5	1.99	8.82E-5	2.02
$\frac{1}{800}$	6.77E-6	2.00	8.43E-6	2.00	2.18E-5	2.01
$\frac{1}{1600}$	1.69E-6	2.00	2.11E-6	2.00	5.41E-6	2.00
$\frac{1}{3200}$	4.23E-7	-	5.28E-7	-	1.35E-6	-

Table 1.4: Rate of convergence for the second-order MUSCL scheme computed in the case of Burgers equation with the smooth initial condition $u^0(x) = \sin(2\pi x)$ on the $[0, 1]$ domain, at time $t = \frac{t_c}{2}$ with a CFL= 0.5.

solution obtained with the first-order Finite Volume scheme and the limited MUSCL scheme. We still see on Figure 1.7 the important gain of accuracy between those two.

To end with this accuracy study, we perform the rates of convergence for the limited MUSCL scheme. The results are gathered in Table 1.5 and Table 1.6. We observe that the rates of convergence are close to 2, even if there are slightly inferior. Indeed, if we confront the numerical errors obtained with the MUSCL scheme with and without slope limiter, we note a slightly loss of accuracy. We end up telling that the slope limiter correctly prevent the apparition of oscillations even though a slightly decrease of accuracy is observed.

	L_1		L_2		L_∞	
Δx	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{100}$	3.64E-3	1.93	4.68E-3	1.70	1.74E-2	1.30
$\frac{1}{200}$	9.63E-4	1.95	1.44E-3	1.71	7.08E-3	1.34
$\frac{1}{400}$	2.49E-4	1.97	4.41E-4	1.70	2.79E-3	1.47
$\frac{1}{800}$	6.37E-5	1.97	1.36E-4	1.69	1.01E-3	1.26
$\frac{1}{1600}$	1.62E-7	-	4.21E-5	-	4.22E-4	-

Table 1.5: Rate of convergence for the second-order MUSCL scheme with limitation computed in the case of linear advection with the smooth initial condition $u^0(x) = \sin(2\pi x)$ on the $[0, 1]$ domain, at time $t = 1$ with a CFL= 0.5.

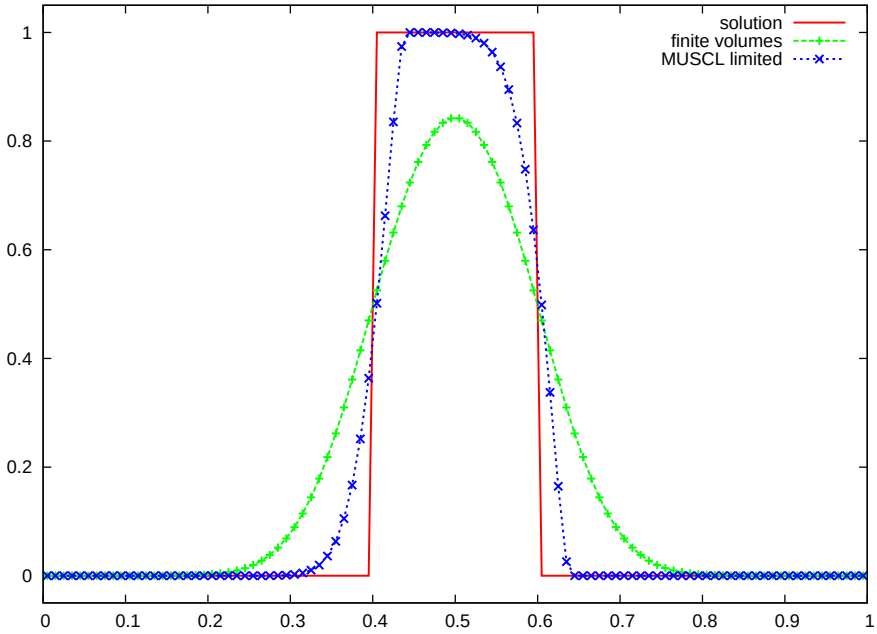


Figure 1.7: Linear advection ($a = 1$) of the discontinuous initial condition $u^0(x) = 1$ if $x \in [0.4, 0.6]$ and null anywhere else, with periodic boundaries, at the end of one period with a CFL= 0.5 and on 100 cells: comparison between the analytical solution and those obtained with the Finite Volume scheme and the MUSCL scheme with slope limiter.

	L_1		L_2		L_∞	
Δx	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{200}$	1.36E-4	1.96	2.20E-4	1.79	1.03E-3	0.95
$\frac{1}{400}$	3.49E-5	1.97	6.36E-5	1.77	5.35E-4	1.61
$\frac{1}{800}$	8.91E-6	1.97	1.86E-5	1.74	1.75E-4	1.21
$\frac{1}{1600}$	2.27E-6	1.98	5.58E-6	1.72	7.56E-5	1.30
$\frac{1}{3200}$	5.75E-7	-	1.69E-6	-	3.07E-5	-

Table 1.6: Rate of convergence for the second-order MUSCL scheme with limitation computed in the case of Burgers equation with the smooth initial condition $u^0(x) = \sin(2\pi x)$ on the $[0, 1]$ domain, at time $t = \frac{t_c}{2}$ with a CFL= 0.5.

Achieving a higher order of accuracy, let say third-order for instance, would require to perform a piecewise polynomial reconstruction from the cell-averaged values supplemented by a high-order time discretization. One possible solution to this problem consists in using the primitive function for $u(x, t)$. This approach was first introduced by Collela and Woodward in their PPM [30](Piecewise Parabolic Method) and has since been employed in a variety of other methods, particularly the ENO [99](Essentially Non Oscillatory) methods developed by Chakravarthy, Enquist, Harten and Osher and its numerous variants. Here, we shall follow another approach by employing the discontinuous Galerkin (DG) method. This is the topic of the next section.

1.1.6 Discontinuous Galerkin discretization

In this section we recall the DG discretization of the one-dimensional conservation law (1.1) following the classical approach introduced by Cockburn and Shu [27]. Our presentation slightly differs from the previous one in the sense that the piecewise polynomial space we are employing is spanned by a Taylor basis instead of a Legendre basis. This particular choice has no importance for solving one-dimensional conservation laws. We are making this choice, following [71], in the perspective of the two-dimensional extension. In this case, the choice of a Taylor basis is justified by the fact that this basis will be independent of the cell form. Namely, we shall employ the same basis regardless the shape of the cell: triangular quadrangular, polygonal. This choice will allow to treat general unstructured grids with the same framework. The price to pay will be that we will have to cope with non-orthogonal basis and thus non-diagonal mass matrix on general polygonal grids. In what follows, we will present DG discretization up to third-order. To preserve the monotonicity of the numerical solution and at the same time its accuracy, we will utilize the vertex-based limiter described in [60]. This vertex-based limiter belongs to the class of moment limiter, initially introduced by Biswas et al. [8], which is itself a generalization of the second-order accurate minmod limiter of van Leer [101].

Variational formulation

Our starting point is the one-dimensional scalar conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \tag{1.19}$$

where $u = u(x,t)$ for $x \in \mathbb{R}$ and $t \in [0,T]$.
Let us recall that $C_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ a is generic cell of size Δx_i and let $\mathbb{P}^K(C_i)$ be the set of polynomials of degree up to K. If u_h denotes the polynomial approximation of the solution, let u_h^i be the restriction of u_h, over the cell C_i

$$u_h^i(x,t) = \sum_{k=0}^{K} u_k^i(t)\,\sigma_k^i(x), \quad x \in C_i \tag{1.20}$$

where u_k^i are the successive moments of u_h and $\{\sigma_k^i\}_{k=0,\dots,K}$ is a basis of $\mathbb{P}^K(C_i)$. The coefficients $u_k^i(t)$ present in (1.20) are determined by writing the local variational formulation of (1.19), for $q = 0, \dots, K$

$$\begin{aligned}\int_{C_i} \frac{\partial u}{\partial t}\,\sigma_q^i \,\mathrm{d}x &= -\int_{C_i} \sigma_q^i \,\frac{\partial f(u)}{\partial x}\,\mathrm{d}x,\\ &= \int_{C_i} f(u)\,\frac{\mathrm{d}\,\sigma_q^i}{\mathrm{d}x}\,\mathrm{d}x - [\overline{f}(u)\,\sigma_q^i]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}.\end{aligned}$$

Here, $[\overline{f}(u)\,\sigma_q^i]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} = \overline{f}_{i+\frac{1}{2}}\,\sigma_q^i(x_{i+\frac{1}{2}}^-) - \overline{f}_{i-\frac{1}{2}}\,\sigma_q^i(x_{i-\frac{1}{2}}^+)$ where $\overline{f}_{i+\frac{1}{2}}$ is, as in the Finite Volume method, the numerical flux which is a single valued function defined at the cell interfaces and in general depends on the trace of the numerical solution on both sides of the interface. Interchanging the function u by its polynomial approximation u_h leads to $K+1$ semi-discrete equations for the same number of unknowns

$$\sum_{k=0}^{K}\left(\int_{C_i} \sigma_k^i\,\sigma_q^i\,\mathrm{d}x\right)\frac{\mathrm{d}\,u_k^i}{\mathrm{d}t} = \int_{C_i} f(u_h^i)\,\frac{\mathrm{d}\,\sigma_q^i}{\mathrm{d}x}\,\mathrm{d}x - [\overline{f}(u)\,\sigma_q^i]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}. \tag{1.21}$$

Before going any further, let $\boldsymbol{U}^i = (u_0^i, ..., u_l^i, ..., u_K^i)^{\mathrm{t}}$ be the unknown vector and $\mathsf{M}_{qk}^i = \int_{C_i} \sigma_k^i \, \sigma_q^i \, \mathrm{d}x$ be the mass matrix coefficients. We notice that this mass matrix is locally defined on each cell and is time independent. The introduction of the scalar product

$$\langle u, v \rangle = \int_{C_i} u(x) \, v(x) \, \mathrm{d}x, \quad , \tag{1.22}$$

allows the following mass matrix coefficients definition

$$\mathsf{M}_{qk}^i = \left\langle \sigma_q^i, \sigma_k^i \right\rangle .$$

This notation shows that M^i is the Gram matrix associated to the chosen polynomial basis and to the scalar product defined previously. Hence, this matrix is symmetric positive definite and thus invertible. Moreover, this matrix being time independent, it can be computed once for all, stored and re-used during the calculation. The choice of an orthogonal basis, *i.e.*, $\left\langle \sigma_q^i, \sigma_k^i \right\rangle = 0$ for $q \neq k$, leads to a diagonal mass matrix which simplifies greatly the scheme and leads to have all the moments u_k^i equations being independent of each others. With this notation, the left-hand side of (1.21) can be identified as the qth row of the matrix-vector product $\mathsf{M}^i \dfrac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{U}^i$.

Projection onto the approximation space

Being given a function $\phi = \phi(x)$, we compute its polynomial approximation restricted to the cell C_i as $\phi_h^i(x) = \sum_{k=0}^K \phi_k^i \, \sigma_k^i$. To this end we define the projection operator Π as

$$\begin{aligned} \Pi \, : \, L^1(C_i) &\longrightarrow \mathbb{P}^K(C_i) \\ \phi &\longrightarrow \Pi(\phi) = \phi_h^i, \end{aligned}$$

where the polynomial approximation $\Pi(\phi)$ is defined by its successive moments ϕ_k^i as

$$\int_{C_i} \Pi(\phi) \, \sigma_q^i \, \mathrm{d}x = \int_{C_i} \phi \, \sigma_q^i \, \mathrm{d}x, \quad \text{for } q = 0, \ldots, K. \tag{1.23}$$

Let us note that (1.23) is equivalent to minimizing the cost function $\|\Pi(\phi) - \Phi\|^2$ where $\| \cdot \|$ refers to the norm induced by the scalar product (1.22). Substituting $\Pi(\phi)$ by $\phi_h^i = \sum_{k=0}^K \phi_k^i \, \sigma_k^i$, leads to a linear system of size $K+1$. The qth row of this linear system reads

$$\sum_{k=0}^{K} \left(\int_{C_i} \sigma_k^i \, \sigma_q^i \, \mathrm{d}x \right) \phi_k^i = \int_{C_i} \phi \, \sigma_q^i \, \mathrm{d}x, \quad \text{for } q = 0, \ldots, K. \tag{1.24}$$

The function ϕ being known at each point of C_i, the right hand-side of (1.24) can be evaluated with an adapted quadrature formula, which can be of Gauss-Legendre type, refer to [27]. Introducing the vectors $\boldsymbol{\Phi}^i = (\phi_0^i, \ldots, \phi_K^i)^{\mathrm{t}}$ and $\widetilde{\boldsymbol{\Phi}}^i = (\langle \phi, \sigma_0^i \rangle, \ldots, \langle \phi, \sigma_K^i \rangle)^{\mathrm{t}}$, system (1.24) can be written under the compact form

$$\mathsf{M}^i \, \boldsymbol{\Phi}^i = \widetilde{\boldsymbol{\Phi}}^i. \tag{1.25}$$

The mass matrix being invertible, this system always admits a unique solution. Moreover, M^i being symmetric positive definite, one can use the Cholesky decomposition algorithm to compute its inverse.

This projection operator is also used to construct the polynomial approximation of the initial data $u^0(x)$

$$u_h^{0,i} = \sum_{k=0}^{K} u_k^{0,i}\, \sigma_k,$$

where the $u_k^{0,i}$ coefficients are obtained by solving the linear system

$$\mathsf{M}^i\, \boldsymbol{U}^{0,i} = \widetilde{\boldsymbol{U}}^{0,i}.$$

In the same spirit, we develop the internal term $\int_{C_i} f(u_h^i)\, \frac{\mathrm{d}\,\sigma_q^i}{\mathrm{d}x}\, \mathrm{d}x$, to get a compact form of the $K+1$ semi-discrete equations.

Compact form of the semi-discrete equations

Knowing the polynomial approximation (1.20) of the function u in the cell C_i, we are able to evaluate at each point of the cell the function $f(u_h^i(x,t))$. The use of the previous projection algorithm with an appropriate quadrature formula leads to set

$$f_h^i(x,t) = \Pi(f(u_h^i)) = \sum_{k=0}^{K} f_k^i(t)\, \sigma_k^i(x),$$

where the f_k^i coefficients are obtained solving the system

$$\mathsf{M}^i\, \boldsymbol{F}^i = \widetilde{\boldsymbol{F}}^i.$$

Here, the vectors $\boldsymbol{F}^i \in \mathbb{R}^{K+1}$ and $\widetilde{\boldsymbol{F}}^i \in \mathbb{R}^{K+1}$ are given by

$$\begin{aligned}
\boldsymbol{F}^i &= (f_0^i, \ldots, f_K^i)^{\mathrm{t}}, \\
\widetilde{\boldsymbol{F}}^i &= (\left\langle f(u_h^i), \sigma_0^i \right\rangle, \ldots, \left\langle f(u_h^i), \sigma_K^i \right\rangle)^{\mathrm{t}}.
\end{aligned}$$

Consequently, with these definitions, the internal term rewrites

$$\int_{C_i} f(u_h^i)\, \frac{\mathrm{d}\,\sigma_q^i}{\mathrm{d}x}\, \mathrm{d}x = \sum_{k=0}^{K} \left(\int_{C_i} \sigma_k^i\, \frac{\mathrm{d}\,\sigma_q^i}{\mathrm{d}x}\, \mathrm{d}x \right)\, f_k^i.$$

Introducing the matrix D^i defined as

$$\mathsf{D}_{qk}^i = \int_{C_i} \sigma_k^i\, \frac{\mathrm{d}\,\sigma_q^i}{\mathrm{d}x}\, \mathrm{d}x,$$

the internal term can been identified as the qth row of the matrix-vector product $\mathsf{D}^i\, \boldsymbol{F}^i$. And finally, the $K+1$ semi-discrete equations (1.21) can be written in the following compact form

$$\mathsf{M}^i\, \frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{U}^i = \mathsf{D}^i\, \boldsymbol{F}^i - \left[\overline{f}_{i+\frac{1}{2}}\, \boldsymbol{\sigma}^i(x_{i+\frac{1}{2}}^-) - \overline{f}_{i-\frac{1}{2}}\, \boldsymbol{\sigma}^i(x_{i-\frac{1}{2}}^+) \right], \tag{1.26}$$

where $\boldsymbol{\sigma}^i(x) = (\sigma_0^i(x), \ldots, \sigma_K^i(x))^{\mathrm{t}}$.
This compact semi-discrete discretization has a theoretical $(K+1)^{\mathrm{th}}$ order of accuracy in space. In the same way that for the MUSCL scheme, we have to apply a time discretization accurate enough to have a global $(K+1)^{\mathrm{th}}$ order of accuracy.

Time discretization

The above semi-discrete DG scheme corresponds to a system of ordinary differential equations which writes

$$\frac{\mathrm{d}\, u_h}{\mathrm{d}t} = L(u_h, t), \tag{1.27}$$

where $L(u_h, t)$ is the operator corresponding to the space discretization, which is generally non-linear. Thus, to solve (1.27), we have to implement a time discretization remaining explicit and with an accuracy adapted to the space discretization. Those reasons conduce us to choose a Runge-Kutta time discretization. On the time interval $[t^n, t^{n+1}]$, knowing u_h^n the piecewise polynomial solution at time t^n, the general algorithm reads

1. Initialize $u_0^h = u_h^n$.
2. For $j = 1, \ldots, J$ perform the intermediary states

$$u_h^{(j)} = \sum_{l=0}^{j-1} \left[\alpha_{jl}\, u_h^{(l)} + \beta_{jl}\, \Delta t^n\, L(u_h^{(l)}, t^n + d_l\, \Delta t^n) \right].$$

3. Increment $u_h^{n+1} = u_h^{(J)}$.

The coefficients α_{jl}, β_{jl} and d_l will be given subsequently. The index J refers to the order of the time discretization. The TVD Runge-Kutta method under consideration is characterized by the following properties given by [24]

$$\begin{aligned} &\alpha_{jl} \geq 0, \\ &\sum_{l=0}^{J-1} \alpha_{jl} = 1, \\ &\text{If } \beta_{jl} \neq 0 \text{ thus } \alpha_{jl} \neq 0. \end{aligned}$$

With these properties, this class of Runge-Kutta methods corresponds to the convex combination of first-order explicit Euler time discretizations. Therefore, the explicit Euler stability would be automatically transported to the high-order time discretization. Besides, for a $(K+1)^{\text{th}}$ space and time discretization in the linear advection case $f(u) = a\, u$, a von Neumann stability analyze leads to the following time step limitation [24]

$$\Delta t^n \leq \min_i \frac{\Delta x_i}{|a|(2K+1)}. \tag{1.29}$$

For the sake of clearness, let us explicit the time discretization schemes for $J = 1, 2, 3$

- First-order

$$u_h^{n+1} = u_h^n + \Delta t^n\, L(u_h^n, t^n).$$

- Second-order

$$\begin{aligned} u_h^{(1)} &= u_h^n + \Delta t^n\, L(u_h^n, t^n), \\ u_h^{n+1} &= \frac{1}{2}(u_h^n + u_h^{(1)}) + \frac{\Delta t^n}{2}\, L(u_h^{(1)}, t^n + \Delta t^n), \end{aligned}$$

- Third-order

$$
\begin{aligned}
u_h^{(1)} &= u_h^n + \Delta t^n\, L(u_h^n, t^n),\\
u_h^{(2)} &= \frac{1}{4}(3\, u_h^n + u_h^{(1)}) + \frac{\Delta t^n}{4}\, L(u_h^{(1)}, t^n + \Delta t^n),\\
u_h^{n+1} &= \frac{2}{3}(u_h^n + 2\, u_h^{(2)}) + \frac{2\,\Delta t^n}{3}\, L(u_h^{(2)}, t^n + \frac{\Delta t^n}{2}),
\end{aligned}
$$

To complete our scheme, the numerical fluxes at the interfaces $\overline{f}_{i+\frac{1}{2}}$ have to be determined. This task will be achieved in the next section by studying the L_2 stability of the semi-discrete DG scheme.

1.1.7 Numerical flux and L_2 stability

In this part, following [52, 24, 98, 49], we want to design a generic form of numerical fluxes which ensures the global stability of our semi-discrete discretization in the L_2 norm. To this end, let us consider the local variational formulation written using u_h^i as a test function

$$
\int_{C_i} u_h^i\, \frac{\partial\, u_h^i}{\partial t}\, \mathrm{d}x = \int_{C_i} f(u_h^i)\, \frac{\partial\, u_h^i}{\partial x}\, \mathrm{d}x - [\overline{f}(u)\, u_h^i]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}.
$$

We can immediately infer

$$
\frac{1}{2}\, \frac{\mathrm{d}}{\mathrm{d}t} \int_{C_i} (u_h^i)^2\, \mathrm{d}x = \int_{C_i} f(u_h^i)\, \frac{\partial\, u_h^i}{\partial x}\, \mathrm{d}x - [\overline{f}(u)\, u_h^i]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}. \tag{1.30}
$$

Here, we make use of the function F which denotes a primitive of the flux function defined as $F(u) = \int_0^u f(s)\, \mathrm{d}s$. Noticing that $\frac{\partial\, F(u_h^i)}{\partial x} = \frac{\mathrm{d}\, F(u_h^i)}{\mathrm{d}u}\, \frac{\partial\, u_h^i}{\partial x} = f(u_h^i)\, \frac{\partial\, u_h^i}{\partial x}$, equation (1.30) writes

$$
\frac{1}{2}\, \frac{\mathrm{d}}{\mathrm{d}t} \int_{C_i} (u_h^i)^2\, \mathrm{d}x = [F(u_h^i)]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} - [\overline{f}(u)\, u_h^i]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}. \tag{1.31}
$$

Let us set $R_i = [\overline{f}(u) u_h^i]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} - [F(u_h^i)]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}$. For periodic boundary conditions, the sum of (1.31) over all the cells writes

$$
\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u_h\|_{L_2}^2 = \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t} \sum_{i,cells} \int_{C_i} (u_h^i)^2\, \mathrm{d}x = -\sum_{i,cells} R_i. \tag{1.32}
$$

At this point, we claim that the global stability in L_2 norm for our semi-discrete scheme amounts to impose

$$
\sum_{i,cells} R_i \geq 0. \tag{1.33}
$$

Next, we determine the form of the numerical flux so that (1.33) is enforced. By interchanging the sum from cells to nodes, (1.33) rewrites

$$
\begin{aligned}
\sum_{i,cells} R_i &= \sum_{i,cells} \left[\overline{f}_{i+\frac{1}{2}} u_h^i(x_{i+\frac{1}{2}}^-, t) - \overline{f}_{i-\frac{1}{2}} u_h^i(x_{i-\frac{1}{2}}^+, t) - F(u_h^i(x_{i+\frac{1}{2}}^-, t)) + F(u_h^i(x_{i-\frac{1}{2}}^+, t)) \right],\\
&= \sum_{i,nodes} \left[\overline{f}_{i+\frac{1}{2}} (u_h^i(x_{i+\frac{1}{2}}^-, t) - u_h^{i+1}(x_{i+\frac{1}{2}}^+, t)) - F(u_h^i(x_{i+\frac{1}{2}}^-, t)) + F(u_h^{i+1}(x_{i+\frac{1}{2}}^+, t)) \right],\\
&= \sum_{i,nodes} \left[\overline{f}_{i+\frac{1}{2}} (u_L - u_R) - F(u_L) + F(u_R) \right].
\end{aligned} \tag{1.34}
$$

where u_L and u_R denote the extrapolated values at the interface of the left and right states on both sides of the interface, *i.e.*, $u_L = u_h^i(x^-_{i+\frac{1}{2}}, t)$ and $u_R = u_h^{i+1}(x^+_{i+\frac{1}{2}}, t)$. Recalling the definition of the primitive function $F(u) = \int_0^u f(s)\,\mathrm{d}s$, equation (1.34) yields

$$\sum_{i,cells} R_i = \sum_{i,nodes} (u_L - u_R)\left(\overline{f}_{i+\frac{1}{2}} - \frac{1}{u_R - u_L}\int_{u_L}^{u_R} f(s)\,\mathrm{d}s\right). \tag{1.35}$$

Finally, the global stability of the semi-discrete scheme in L_2 norm is ensured provided that the numerical flux is written

$$\overline{f}_{i+\frac{1}{2}} = \frac{1}{u_R - u_L}\int_{u_L}^{u_R} f(s)\,\mathrm{d}s - C_{i+\frac{1}{2}}(u_R - u_L), \tag{1.36}$$

where $C_{i+\frac{1}{2}}$ is a non-negative scalar characterizing the numerical viscosity of the scheme. Finally, the sufficient condition used to determine the numerical flux (1.36) gives the global L_2 stability of the semi-discrete scheme

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u_h\|^2_{L_2} = -\sum_{i,nodes} C_{i+\frac{1}{2}}(u_R - u_L)^2 \leq 0. \tag{1.37}$$

The numerical dissipation of the semi-discrete discontinuous Galerkin scheme is directly related to the square of the polynomial approximation jumps located at the interfaces. Thanks to this dissipation, we can hope that the scheme will remain L_2 stable (with a correct time discretization). An obvious dimensional analysis of (1.37), shows that $C_{i+\frac{1}{2}}$ has the dimension of a velocity.
In the linear case, $f(u) = a\,u$, where a is the constant advection velocity. Recalling (1.36) in this case, one gets

$$\overline{f}_{i+\frac{1}{2}} = \frac{a}{2}(u_L + u_R) - C_{i+\frac{1}{2}}(u_R - u_L) \tag{1.38}$$

We recognize two different parts in this flux, the centered one $\frac{a}{2}(u_L + u_R)$, and the viscous one $C_{i+\frac{1}{2}}(u_R - u_L)$ which brings dissipation and consequently stability. Moreover, if we set $C_{i+\frac{1}{2}} =\mid a \mid$ in the above formula, we recover the well known upwind flux.
In the non-linear case, we can use a quadrature formula to evaluate $\int_{u_L}^{u_R} f(s)\,\mathrm{d}s$. If we choose the trapezoidal rule and take $C_{i+\frac{1}{2}} = \frac{1}{2}\max(|f'(u_L)|, |f'(u_R)|)$, we recover the local Lax-Friedrichs flux

$$\overline{f}_{i+\frac{1}{2}} = \frac{f(u_L) + f(u_R)}{2} - \max(|f'(u_L)|, |f'(u_R)|)(\frac{u_R - u_L}{2}). \tag{1.39}$$

The use of this centered term $\frac{f(u_L)+f(u_R)}{2}$ instead of $\frac{1}{u_R-u_L}\int_{u_L}^{u_R} f(s)\,\mathrm{d}s$ modifies very slightly the numerical solution, the dissipative error coming from the viscous term. For instance, in the case of the Burgers equation with the initial data $u^0(x) = \sin(2\pi x)$, the relative error difference between the two numerical solutions obtained with the first-order scheme is about 0.032%.

Comment 1 *Let us point out that there is a strong link between the L_2 stability and the entropy inequality. As noticed in [98], the weak solutions of the one-dimensional scalar conservation laws* (1.19) *may not be unique, and the unique, physically relevant weak solution, named entropy solution, satisfies an entropy inequality, refer to [43, 6]. It is very important to design numerical scheme that mimics at the discrete level an entropy inequality. In the framework of DG discretization, it is possible to prove that the DG scheme satisfies a cell entropy inequality, refer to [98, 52]. This result does not depend on the accuracy of the scheme. Moreover, it is shown in [98] that the cell entropy inequality trivially implies an L_2 stability of the numerical solution.*

1.1.8 Taylor basis

Following [71] we have made the choice to use a Taylor basis to construct the polynomial approximation of our numerical solution. The main interest of this basis lies in the fact that for two-dimensional extension it does not depend of the shape of the cell. Namely, we shall employ the same basis for triangular and quadrangular cells. Moreover, with this basis, we will be able to construct DG discretization over general polygonal grids. The Taylor basis is constructed by means of the Taylor expansion at the centroid of the cells. Let us consider a function $u : \mathbb{R} \to \mathbb{R}$ be K times differentiable at point x_i, the centroid of the cell C_i. Under this hypothesis, we can write the Taylor expansion of u at x_i up to K^{th} order for all $x \in C_i$.

$$\begin{aligned} u(x) &= u(x_i) + u'(x_i)(x - x_i) + \frac{u''(x_i)}{2}(x - x_i)^2 + \cdots + \frac{u^{(K)}(x_i)}{K!}(x - x_i)^K + o((x - x_i)^K), \\ &= \sum_{k=0}^{K} \frac{u^{(k)}(x_i)}{k!} (x - x_i)^k + o((x - x_i)^K), \end{aligned} \tag{1.40}$$

where $u^{(K)}$ is K^{th} derivative of the function u. If we set the polynomial function u_h^i to be $\sum_{k=0}^{K} \frac{u^{(k)}(x_i)}{k!} (x - x_i)^k$, this approximation would be K^{th} order accurate. We can immediately identify the basis functions and the successive moments of the polynomial approximation function u_h^i associated to this definition

$$u_k^i = u^{(k)}(x_i) \, (\Delta x_i)^k \quad \text{and} \quad \sigma_k^i = \frac{1}{k!} \left(\frac{x - x_i}{\Delta x_i} \right)^k, \qquad \text{for } k = 0, \ldots, K, \, \forall \, x \in C_i.$$

Here, the coefficient Δx_i is utilized to normalize the basis functions. In this approximation, the first moment associated to the first basis function 1 is equal to $u(x_i)$. We introduce $\langle u \rangle_i$, the mean value of the function u over the cell C_i

$$\langle u \rangle_i = \frac{1}{\Delta x_i} \int_{C_i} u(x) \, \mathrm{d}x.$$

The integration of (1.40) leads to

$$\begin{aligned} \langle u \rangle_i &= \sum_{k=0}^{K} \frac{u^{(k)}(x_i)}{k!} \left\langle (x - x_i)^k \right\rangle_i + \left\langle o((x - x_i)^K) \right\rangle_i, \\ &= \sum_{k=0}^{K} u^{(k)}(x_i) \, (\Delta x_i)^k \, \frac{1}{k!} \left\langle \left(\frac{x - x_i}{\Delta x_i} \right)^k \right\rangle_i + o((x - x_i)^K). \end{aligned} \tag{1.41}$$

Performing the difference between (1.40) and (1.41), one gets

$$\begin{aligned} u(x) - \langle u \rangle_i &= \sum_{k=0}^{K} u^{(k)}(x_i) \, (\Delta x_i)^k \, \frac{1}{k!} \left[\left(\frac{x - x_i}{\Delta x_i} \right)^k - \left\langle \left(\frac{x - x_i}{\Delta x_i} \right)^k \right\rangle_i \right] + o((x - x_i)^K), \\ &= \sum_{\boldsymbol{k=1}}^{K} u^{(k)}(x_i) \, (\Delta x_i)^k \, \frac{1}{k!} \left[\left(\frac{x - x_i}{\Delta x_i} \right)^k - \left\langle \left(\frac{x - x_i}{\Delta x_i} \right)^k \right\rangle_i \right] + o((x - x_i)^K), \end{aligned} \tag{1.42}$$

because $\langle 1 \rangle_i = 1$ and so $1 - \langle 1 \rangle_i = 0$.

Thanks to relation (1.42), we can define a new K^{th} order polynomial approximation u_h^i of the function u

$$\begin{aligned} u_h^i(x) &= \sum_{k=0}^{K} u_k^i \, \sigma_k^i, \\ &= \langle u \rangle_i + \sum_{k=1}^{K} u^{(k)}(x_i) \, (\Delta x_i)^k \; \frac{1}{k!} \left[\left(\frac{x - x_i}{\Delta x_i} \right)^k - \left\langle \left(\frac{x - x_i}{\Delta x_i} \right)^k \right\rangle_i \right]. \end{aligned}$$

This leads to the space approximation $\mathbb{P}^K(C_i)$ to be spanned by the basis $\{\sigma_k^c\}_{k=0\ldots K}$, where

$$\sigma_k^i = \frac{1}{k!} \left[\left(\frac{x - x_i}{\Delta x_i} \right)^k - \left\langle \left(\frac{x - x_i}{\Delta x_i} \right)^k \right\rangle_i \right]. \tag{1.43}$$

This is the chosen Taylor basis. The moment related to the first basis function 1 corresponds to $\langle u \rangle_i$, the mean value of the function u over cell C_i. If we do not go any further in the approximation, we recover the classical Finite Volume discretization. In the case of higher order schemes, the successive moments correspond to the successive derivatives of the function u located at the centroid C_i, times a normalizing factor $(\Delta x_i)^k$. From (1.43), it immediately comes that, for $k = 0, \ldots, K$

$$\mathsf{M}_{0k}^i = \mathsf{M}_{k0}^i = \int_{C_i} \sigma_0^i \, \sigma_k^i \, \mathrm{d}x = \int_{C_i} \sigma_k^i \, \mathrm{d}x = \Delta x_i \, \delta_{0k},$$

where δ_{ij}, the Kronecker symbol, is equal to 1 if $i = j$ and is null if $i \neq j$. Consequently, the equation corresponding to the function mean value $u_0^i = \langle u \rangle_i$ is independent of the other polynomial basis components equations.

In what follows, we describe the numerical schemes we have developed, using a DG discretization, up to the third-order. We explicit now the polynomial basis functions for the third-order scheme, *i.e.*, for $K = 2$

$$\sigma_0^i = 1, \; \sigma_1^i = \frac{x - x_i}{\Delta x_i}, \; \sigma_2^i = \frac{1}{2} \left[\left(\frac{x - x_i}{\Delta x_i} \right)^2 - \frac{1}{12} \right]. \tag{1.44}$$

These functions are depicted on Figure 1.8.

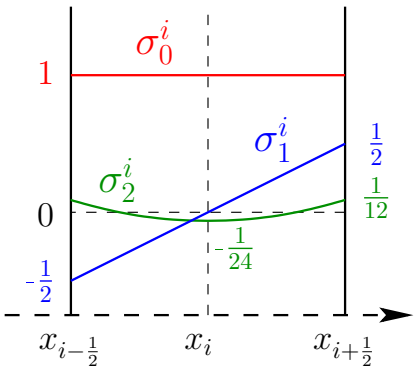


Figure 1.8: Taylor basis of $\mathbb{P}^K(C_i)$ over the cell C_i, in the third-order case.

We point out that the only geometry contributions present in (1.43) and (1.44) are the centroid position x_i and the scaling factors Δx_i. This remark will have a huge importance regarding the two-dimensional extension.

So far, we have made explicit the DG space discretization with the use of the Taylor basis, the Runge-Kutta time discretization and the numerical flux formula to impose a global L_2 stability to the solution. Thus, we are now able to completely implement this RKDG scheme and observe the accuracy gain between schemes ranging from first-order to third-order, refer to Figure 1.9.

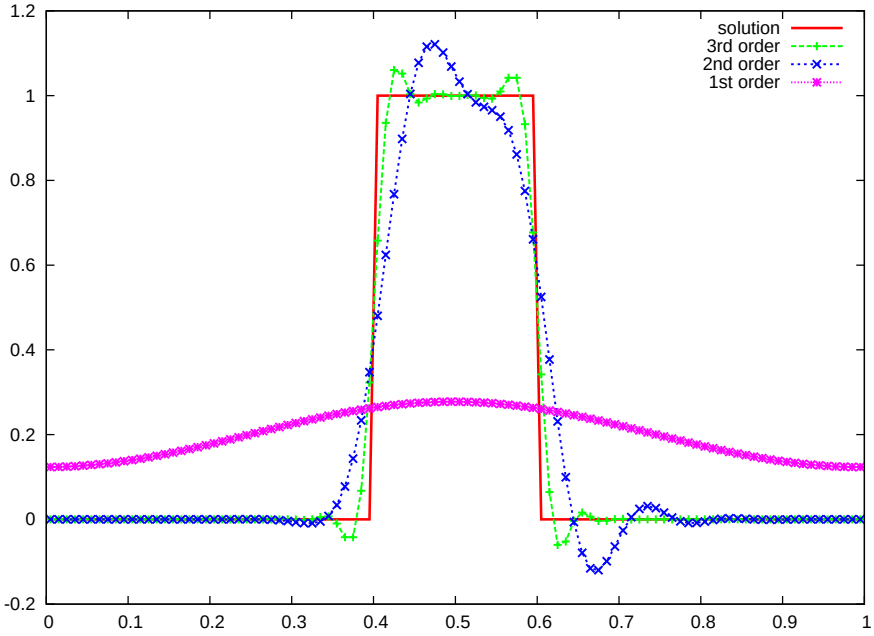


Figure 1.9: Linear advection ($a = 1$) with the discontinuous initial condition $u^0(x) = 1$ if $x \in [0.4, 0.6]$ and null anywhere else, with periodic boundaries, at the end of ten periods with a CFL=0.2 and grid characterized by 100 uniformly spaced cells. Comparison between the analytical solution and the numerical solutions obtained with the first, second and third-order DG schemes.

Observing Figure 1.9, we clearly see the huge gain of accuracy between the three different solutions. Nevertheless, as for the second-order MUSCL scheme, spurious oscillations are located in the vicinity of the discontinuity points, for the numerical solutions obtained with the second-order DG scheme and the third-order DG scheme. To restore monotonicity, we have to implement a high-order generalization of the slope limiter method presented in Section 1.1.4.

1.1.9 Slope limiters for high-order DG methods

In this paragraph we describe a slope limiter for the DG method that retains as high an order as possible and does not automatically reduce to first-order in case of smooth extrema. This slope limiter, which belongs to the class of moment limiter has been adapted from the works described in [60, 109]. It has the main feature to preserve the smooth extrema for parabolic approximations. Moreover, its extension to two-dimensional unstructured grids is feasible, as it will be shown later. Bearing this in mind, we start by recalling the classical limitation procedure originally introduced by Cockburn and Shu [27].

The generalized slope limiter of Cockburn and Shu

Knowing the third order polynomial development of the solution in cell C_i at time t

$$u_h^i(x,t) = u_0^i + u_1^i \left(\frac{x-x_i}{\Delta x_i}\right) + u_2^i \frac{1}{2}\left[\left(\frac{x-x_i}{\Delta x_i}\right)^2 - \frac{1}{12}\right],$$

we deduce the extrapolated values at the interfaces of the cell C_i, *i.e.*, $u^+_{i-\frac{1}{2}} = u_h^i(x_{i-\frac{1}{2}},t)$ and $u^-_{i+\frac{1}{2}} = u_h^i(x_{i+\frac{1}{2}},t)$, which write

$$\begin{aligned} u^+_{i-\frac{1}{2}} &= u_0^i - \frac{1}{2}u_1^i + \frac{1}{12}u_2^i, \\ u^-_{i+\frac{1}{2}} &= u_0^i + \frac{1}{2}u_1^i + \frac{1}{12}u_2^i. \end{aligned}$$

To ensure the monotonicity of the solution, we seek a limitation of the development coefficients so that

$$\left\{ \begin{array}{l} u^+_{i-\frac{1}{2}} \in [\min(u_0^{i-1}, u_0^i), \max(u_0^{i-1}, u_0^i)], \\ u^-_{i+\frac{1}{2}} \in [\min(u_0^i, u_0^{i+1}), \max(u_0^i, u_0^{i+1})]. \end{array} \right. \tag{1.45}$$

To this end, we introduce $\widetilde{u}_i$ and $\widetilde{\widetilde{u}}_i$ defined as

$$\left\{ \begin{array}{l} u^+_{i-\frac{1}{2}} = u_0^i - \widetilde{\widetilde{u}}_i, \\ u^-_{i+\frac{1}{2}} = u_0^i + \widetilde{u}_i. \end{array} \right. \tag{1.46}$$

And to satisfy (1.45), we limit using the minmod function as

$$\begin{aligned} \widetilde{\widetilde{u}}_i^{\lim} &= \text{minmod}(\widetilde{\widetilde{u}}_i, u_0^i - u_0^{i-1}, u_0^{i+1} - u_0^i), \\ \widetilde{u}_i^{\lim} &= \text{minmod}(\widetilde{u}_i, u_0^i - u_0^{i-1}, u_0^{i+1} - u_0^i). \end{aligned}$$

By means of these quantities, we rebuild the extrapolated values at the interfaces setting

$$\begin{aligned} u^+_{i-\frac{1}{2}} &= u_0^i - \widetilde{\widetilde{u}}_i^{\lim}, \\ u^-_{i+\frac{1}{2}} &= u_0^i + \widetilde{u}_i^{\lim}. \end{aligned}$$

According to (1.46), we have

$$\begin{aligned} \widetilde{\widetilde{u}}_i^{\lim} &= \frac{1}{2}u_1^i - \frac{1}{12}u_2^i, \\ \widetilde{u}_i^{\lim} &= \frac{1}{2}u_1^i + \frac{1}{12}u_2^i. \end{aligned}$$

And finally, we are able to deduce the limited values of the polynomial components

$$\begin{aligned} u_1^{i,\lim} &= \widetilde{u}_i^{\lim} + \widetilde{\widetilde{u}}_i^{\lim}, \\ u_2^{i,\lim} &= 6\,(\widetilde{u}_i^{\lim} - \widetilde{\widetilde{u}}_i^{\lim}). \end{aligned}$$

This limitation is easy to set up and efficient. As it is shown in [27], it ensures that the resulting DG scheme is TVDM, that is the scheme is characterized by a total variation diminishing in the

means. However, like any other TVD scheme, accuracy degenerates to first-order at extrema regardless its smoothness. We can observe the impact of this limitation procedure on smooth extrema in Figure 1.10, where the definition of the convected parabolic signal is given in the equation (1.58) presented in the next section devoted to the numerical results. To overcome this difficulty, Shu has proposed a modification of the minimod function which requires a knowledge of the bounds of the second derivative of the initial data. This modification, which we will not develop here, is described in details in [27].

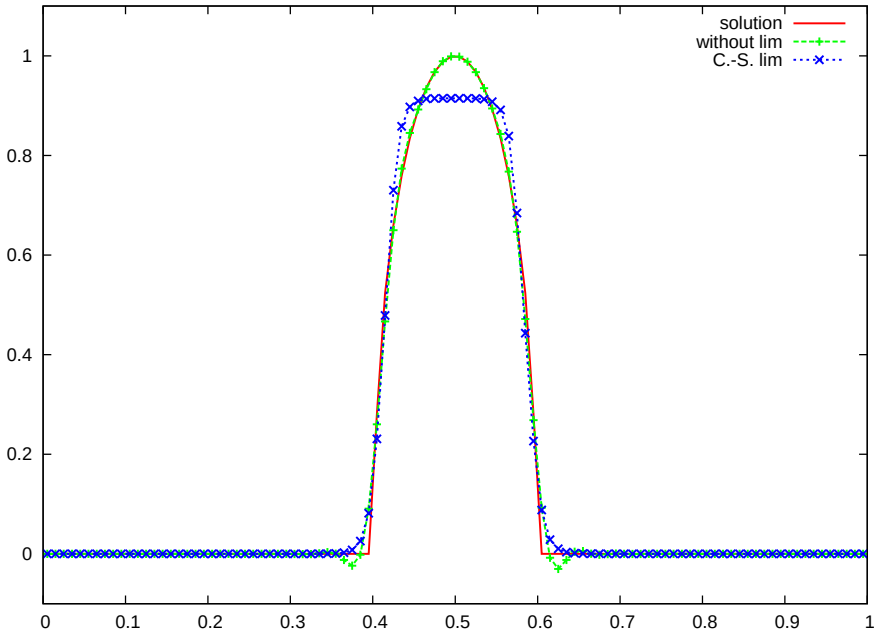


Figure 1.10: Linear advection ($a = 1$) of a parabolic signal, with periodic boundary conditions, at the end of ten periods with a CFL=0.2 and on grid characterized by 100 uniformly spaced cells. Comparison between the analytical solution and the numerical solution obtained with the third-order DG scheme without and with the Cockburn and Shu limitation.

Hierarchical vertex-based slope limiter

We now describe a hierarchical vertex-based slope limiter which is an adaptation of the works presented in [60, 109]. For our third-order DG discretization, this method has the main feature to preserve the local smooth extrema of the numerical solution. Being based on a hierarchical procedure limitation of the polynomial components of the solution, this method is theoretically generalizable to an arbitrary order of accuracy, refer to [60]. In the following paragraph, we start by introducing this limitation procedure in the frame of second-order DG discretization. First, let us recall that the second-order polynomial development of the solution over the cell C_i at time t reads

$$u_h^i(x,t) = u_0^i + u_1^i \left(\frac{x - x_i}{\Delta x_i} \right).$$

To succeed in limiting this polynomial function to ensure its monotonicity we introduce the limitation coefficient $\alpha_{(1)}^i \in [0,1]$ such as the limited solution writes

$$u_h^{i,\lim}(x,t) = u_0^i + \alpha_{(1)}^i \, u_1^i \left(\frac{x - x_i}{\Delta x_i} \right).$$

Let us set $u_{i\pm\frac{1}{2}}^{\min} = \min(u_0^i, u_0^{i\pm1})$ and $u_{i\pm\frac{1}{2}}^{\max} = \max(u_0^i, u_0^{i\pm1})$, which correspond to the minimum and maximum mean values of the solution, in the cells surrounding the interfaces $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$. With these definitions, we seek for limitation imposing the extrapolated value of the solution at an interface to be bounded by the minimum and maximum mean values surrounding this interface, *i.e.*,

$$\left\{ \begin{array}{l} u_{i-\frac{1}{2}}^{\min} \leq u_{i-\frac{1}{2}}^{+} \leq u_{i-\frac{1}{2}}^{\max}, \\ u_{i+\frac{1}{2}}^{\min} \leq u_{i+\frac{1}{2}}^{-} \leq u_{i+\frac{1}{2}}^{\max}, \end{array} \right. \tag{1.47}$$

with the same definitions of $u_{i-\frac{1}{2}}^{+} = u_h^i(x_{i-\frac{1}{2}}, t)$ and $u_{i+\frac{1}{2}}^{-} = u_h^i(x_{i+\frac{1}{2}}, t)$ given previously.

To satisfy (1.47), we introduce the coefficient $\alpha_{(1)}^{i,i-\frac{1}{2}}$ defined by

$$\alpha_{(1)}^{i,i-\frac{1}{2}} = \left\{ \begin{array}{ll} \min\left(1, \dfrac{u_{i-\frac{1}{2}}^{\max} - u_0^i}{u_{i-\frac{1}{2}}^{+} - u_0^i}\right) & \text{if } u_{i-\frac{1}{2}}^{+} - u_0^i > 0, \\ 1 & \text{if } u_{i-\frac{1}{2}}^{+} = u_0^i, \\ \min\left(1, \dfrac{u_{i-\frac{1}{2}}^{\min} - u_0^i}{u_{i-\frac{1}{2}}^{+} - u_0^i}\right) & \text{if } u_{i-\frac{1}{2}}^{+} - u_0^i < 0, \end{array} \right. \tag{1.48}$$

and the coefficient $\alpha_{(1)}^{i,i+\frac{1}{2}}$ defined by

$$\alpha_{(1)}^{i,i+\frac{1}{2}} = \left\{ \begin{array}{ll} \min\left(1, \dfrac{u_{i+\frac{1}{2}}^{\max} - u_0^i}{u_{i+\frac{1}{2}}^{-} - u_0^i}\right) & \text{if } u_{i+\frac{1}{2}}^{-} - u_0^i > 0, \\ 1 & \text{if } u_{i+\frac{1}{2}}^{-} = u_0^i, \\ \min\left(1, \dfrac{u_{i+\frac{1}{2}}^{\min} - u_0^i}{u_{i+\frac{1}{2}}^{-} - u_0^i}\right) & \text{if } u_{i+\frac{1}{2}}^{-} - u_0^i < 0. \end{array} \right. \tag{1.49}$$

Thus, using (1.48) and (1.49) the slope limiter coefficient writes

$$\alpha_{(1)}^{i} = \min(\alpha_{(1)}^{i,i-\frac{1}{2}}, \alpha_{(1)}^{i,i+\frac{1}{2}}). \tag{1.50}$$

Under this condition, the limited function $u_h^{i,\lim}$ respects constraint (1.47). We can also prove that this limitation preserves the linear fields. Thereby, if the function $u(x) = ax + b$ is linear over the domain, the slope limiter satisfies $\alpha_{(1)}^{i} = 1$ for $i = 1, \ldots, I$.

Now, to limit third or even higher order schemes, the previous procedure is adapted to limit the successive components of the polynomial solution. Let us explicit this method for the third-order DG scheme. After recalling the polynomial development of the solution over the cell C_i at time t

$$u_h^i(x,t) = u_0^i + u_1^i \left(\frac{x - x_i}{\Delta x_i}\right) + u_2^i \frac{1}{2}\left[\left(\frac{x - x_i}{\Delta x_i}\right)^2 - \frac{1}{12}\right], \tag{1.51}$$

we introduce the limitation coefficients $\alpha_{(1)}^{i} \in [0,1]$ and $\alpha_{(2)}^{i} \in [0,1]$ such as the limited solution writes

$$u_h^{i,\lim}(x,t) = u_0^i + \alpha_{(1)}^{i}\, u_1^i \left(\frac{x - x_i}{\Delta x_i}\right) + \alpha_{(2)}^{i}\, u_2^i \frac{1}{2}\left[\left(\frac{x - x_i}{\Delta x_i}\right)^2 - \frac{1}{12}\right].$$

The coefficients $\alpha^i_{(2)}$ and $\alpha^i_{(1)}$ are successively determined using the previous second-order DG scheme limitation procedure applied on the following linear reconstructions

$$\widetilde{\widetilde{u}}_h = u_1^i + u_2^i \left(\frac{x - x_i}{\Delta x_i} \right), \tag{1.52}$$

$$\widetilde{u}_h = u_0^i + u_1^i \left(\frac{x - x_i}{\Delta x_i} \right). \tag{1.53}$$

We firstly note that $\widetilde{u}_h$ is the linear restriction of the polynomial development (1.51). Secondly, deriving equation (1.51) with respect to the variable x we directly identify $\widetilde{\widetilde{u}}_h$ to $\Delta x_i \frac{\partial u_h^i}{\partial x}$. Hence, $\widetilde{\widetilde{u}}_h$ corresponds to the partial derivative of the polynomial function $u_h^i(x,t)$ with respect to x. We apply the previous limitation procedure setting

$$\widetilde{\widetilde{u}}_h^{\lim} = u_1^i + \alpha^i_{(2)}\, u_2^i \left(\frac{x - x_i}{\Delta x_i} \right), \tag{1.54}$$

$$\widetilde{u}_h^{\lim} = u_0^i + \alpha^i_{(1)}\, u_1^i \left(\frac{x - x_i}{\Delta x_i} \right). \tag{1.55}$$

With the use of (1.48), (1.49) and (1.50) we firstly compute $\alpha^i_{(2)}$ substituting in these equations the mean value u_0^i for the scaled derivative u_1^i. Afterward, we evaluate $\alpha^i_{(1)}$ by means of (1.48), (1.49) and (1.50) on the linear reconstruction $\widetilde{u}_h^{\lim}$. Finally, we impose

$$\alpha^i_{(1)} = \max(\alpha^i_{(1)}, \alpha^i_{(2)}). \tag{1.56}$$

This last condition is fundamental because it ensures the preservation of smooth extrema. Indeed, the function being approximated by quadratic functions on each cell, this function derivative approximation is linear. Knowing that the limitation procedure defined by (1.48), (1.49) and (1.50) preserves the linear fields, it is clear that $\alpha^i_{(2)} = 1$ and according to (1.56) $\alpha^i_{(1)} = 1$. This slope limiter method preserves the quadratic fields. Indeed, if a function is quadratic and its derivative linear the slope limiter coefficients given by (1.48) and(1.49) are equal to one. This procedure can be generalized to $q+1$ arbitrary order DG scheme. As we did in the case of the second order DG scheme, we apply the limitation characterized by (1.48), (1.49) and (1.50) to the successive linear restrictions corresponding to $(\Delta x_i)^p \frac{\partial^p u_h^i}{\partial x^p}$ where $p = 0, \ldots, q$, starting with the higher order. The procedure consists in constructing a nondecreasing sequence of correction factors $\alpha^i_{(p)}$. And finally, we impose for $p = 1, \ldots, q$

$$\alpha^i_{(p)} = \max_{p \le l \le q} \alpha^i_{(l)}. \tag{1.57}$$

As soon as $\alpha^i_{(l)} = 1$ is encountered, no further limiting is required. In principle, this hierarchical limitation should preserve q^{th} order polynomial fields.
To assess the efficiency of the limitation procedure, we apply this method to the third-order DG scheme in the case of the linear advection of a parabolic signal, as we did with the Cockburn and Shu limitation. We note that the solution is perfectly monotonic and, this time, preserves the smooth extremum better than the Cockburn and Shu limitation did, refer to Figure 1.11.
Finally, we have obtained a discontinuous Galerkin discretization of one-dimensional scalar conservation laws, up to the third-order provided with a hierarchical vertex-based slope limiter that preserves smooth extrema. Now, we are able to perform a numerical analysis of these DG schemes beginning with the second-order one to compare it with the MUSCL scheme presented previously. Then, we compute the rates of convergence for the third-order scheme to confirm numerically its accuracy.

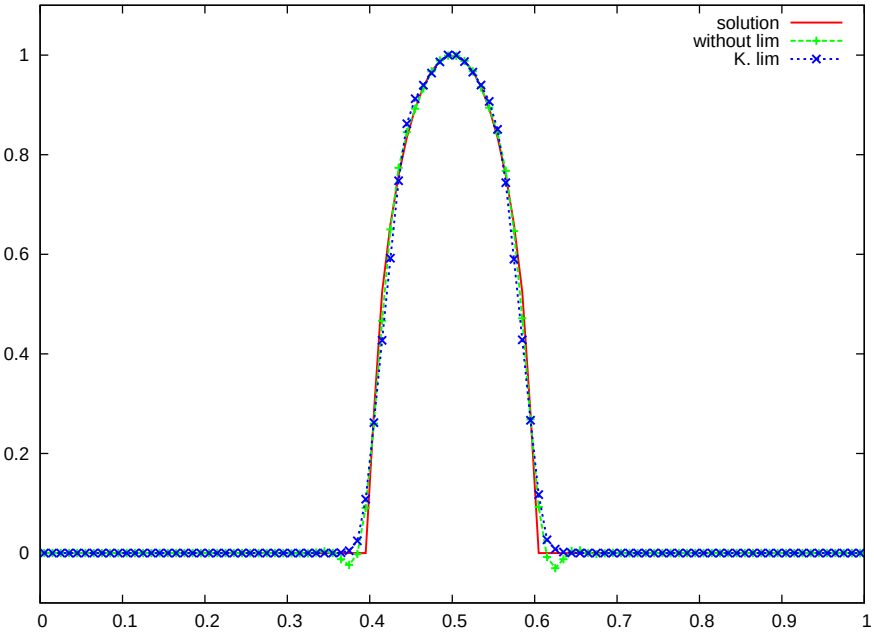


Figure 1.11: Linear advection ($a = 1$) of a parabolic signal, with periodic boundary conditions, at the end of ten periods with a CFL=0.2 using a grid characterized by 100 uniformly spaced cells. Comparison between the analytical solution and the numerical solutions obtained with the third-order DG scheme without limitation and with the hierarchical vertex-based slope limiter.

1.1.10 Numerical results

Firstly, we compare the second-order DG scheme with the second-order MUSCL scheme to put in evidence the accuracy difference between these two methods. This difference is depicted in Figure 1.12. We observe that even if we are in the presence of two second-order schemes, the results are noticeably different the DG scheme being closer to the analytical solution.
To measure the accuracy difference, we compute the numerical errors in the linear case of the advection of a smooth signal with the two second-order schemes presented in the exactly same conditions, meaning at the same final time, with the same CFL condition and without slope limiter.

	L_1		L_2		L_∞	
Δx	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{100}$	1.44E-3	2.00	1.60E-3	2.01	2.29E-3	2.01
$\frac{1}{200}$	3.58E-4	2.00	3.98E-4	2.00	5.69E-4	2.00
$\frac{1}{400}$	8.95E-5	2.00	9.94E-5	2.00	1.42E-4	2.00
$\frac{1}{800}$	2.24E-5	2.00	2.48E-5	2.00	3.54E-5	2.00
$\frac{1}{1600}$	5.59E-6	-	6.21E-6	-	8.84E-6	-

Table 1.7: Rate of convergence for the second-order MUSCL scheme computed in the case of linear advection with the smooth initial condition $u^0(x) = \sin(2\pi x)$ on the $[0, 1]$ domain, at time $t = 1$ with a CFL= 0.2 without slope limiter.

Firstly, Table 1.8 shows that the second-order DG scheme has the expected accuracy. Secondly, comparing the results gathered in Table 1.7 and in Table 1.8, we can conclude that the second-

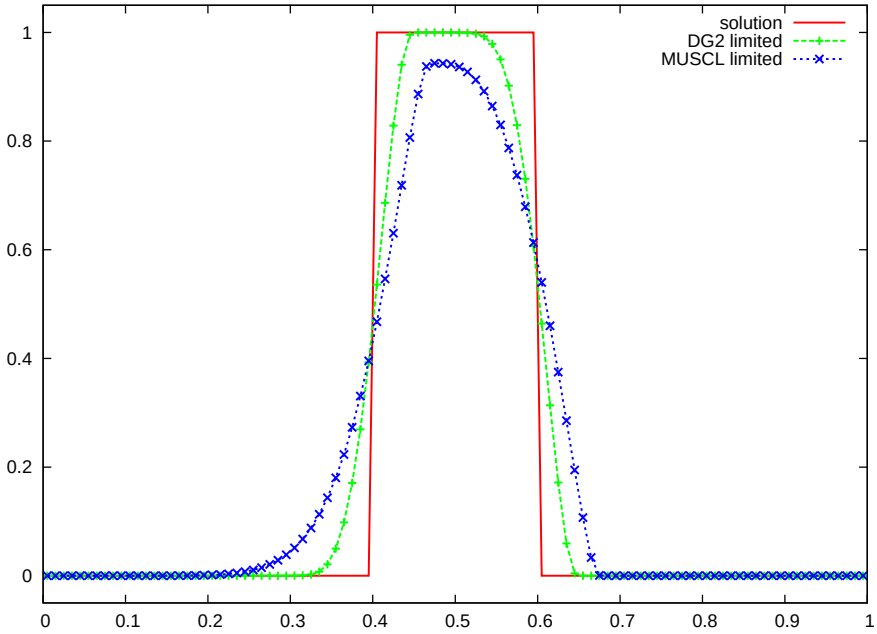


Figure 1.12: Linear advection ($a = 1$) of the discontinuous initial condition $u^0(x) = 1$ if $x \in [0.4, 0.6]$ and null anywhere else, with periodic boundary conditions, at the end of ten periods with a CFL=0.2 and using a grid characterized by 100 uniformly spaced cells. Comparison between the analytical solution and the numerical solutions obtained with the second-order MUSCL scheme with minmod slope limiter and the second-order DG scheme with the vertex-based slope limiter.

	L_1		L_2		L_∞	
Δx	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{100}$	1.81E-4	2.01	2.07E-4	2.00	3.98E-4	1.96
$\frac{1}{200}$	4.47E-5	2.01	5.16E-5	2.00	1.02E-4	1.98
$\frac{1}{400}$	1.11E-5	2.00	1.29E-5	2.00	2.59E-5	1.99
$\frac{1}{800}$	2.77E-6	2.00	3.22E-6	2.00	6.52E-6	2.00
$\frac{1}{1600}$	6.91E-7	-	8.06E-7	-	1.63E-6	-

Table 1.8: Rate of convergence for the second-order DG scheme computed in the case of linear advection with the smooth initial condition $u^0(x) = \sin(2\pi x)$ on the $[0, 1]$ domain, at time $t = 1$ with a CFL= 0.2 without slope limiter.

order DG scheme is more accurate than the MUSCL scheme, five to eight times more in this case.

Now, we perform the convergence analysis for the third-order DG scheme.

Observing Table 1.9, we note the huge gap between the numerical errors obtained with the second-order DG scheme and the third-order DG scheme. We also note that the third-order scheme reaches exactly the expected numerical order. Table 1.10 presents the same results with this difference that the asymptotic regime where the numerical orders for the three norms are equal to 3.00 is reached with more refined meshes ($\Delta x_1 = \frac{1}{6400}$ and $\Delta x_2 = \frac{1}{12800}$).

We end up this numerical results section and this 1D scalar conservation laws part measuring

	L_1		L_2		L_∞	
Δx	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{100}$	6.73E-7	3.00	7.50E-7	3.00	1.16E-6	3.00
$\frac{1}{200}$	8.41E-8	3.00	9.37E-8	3.00	1.45E-7	3.00
$\frac{1}{400}$	1.05E-8	3.00	1.17E-8	3.00	1.81E-8	3.00
$\frac{1}{800}$	1.31E-9	3.00	1.46E-9	3.00	2.26E-9	3.00
$\frac{1}{1600}$	1.64E-10	-	1.83E-10	-	2.83E-10	-

Table 1.9: Rate of convergence for the third-order DG scheme without slope limiter computed in the case of linear advection with the smooth initial condition $u^0(x) = \sin(2\pi x)$ on the $[0, 1]$ domain, at time $t = 1$ with a CFL= 0.2.

	L_1		L_2		L_∞	
Δx	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{200}$	1.77E-7	2.91	3.17E-7	2.85	1.67E-6	2.89
$\frac{1}{400}$	2.35E-8	2.95	4.38E-8	2.90	2.25E-7	2.97
$\frac{1}{800}$	3.05E-9	2.97	5.84E-9	2.94	2.88E-8	2.93
$\frac{1}{1600}$	3.90E-10	2.98	7.59E-10	2.97	3.78E-9	2.95
$\frac{1}{3200}$	4.74E-11	-	9.72E-11	-	4.89E-10	-

Table 1.10: Rate of convergence for the third-order DG scheme without slope limiter computed in the case of Burgers equation with the smooth initial condition $u^0(x) = \sin(2\pi x)$ on the $[0, 1]$ domain, at time $t = \frac{t_c}{2}$ with a CFL= 0.2.

the influence of the presented limitations over the numerical schemes. In order to confront the Cockburn and Shu limitation and the hierarchical vertex-based limitation presented before, we apply our third-order DG scheme on the linear advection with the following initial condition taken from [109], which corresponds to the succession of smooth and non-smooth signals (Gaussian, rectangular, triangular and parabolic signals)

$$u^0(x) = \begin{cases} (G(x, \beta, z - \delta) + G(x, \beta, z + \delta) + 4G(x, \beta, z))/6 & \text{if } x \in [-0.8, -0.6], \\ 1 & \text{if } x \in [-0.4, -0.2], \\ 1 - |10(x - 0.1)| & \text{if } x \in [0, 0.2], \\ (F(x, \alpha, aa - \delta) + F(x, \alpha, aa + \delta) + 4F(x, \alpha, aa))/6 & \text{if } x \in [0.4, 0.6], \\ 0 & \text{else.} \end{cases} \tag{1.58}$$

The parameter values are given by $aa = 0.5$, $z = -0.7$, $\delta = 0.005$, $\alpha = 10$, $\beta = \log 2/(36\delta^2)$, and the F and G functions are defined as

$$G(x, \beta, z) = \exp(-\beta(x - z)^2),$$
$$F(x, \alpha, aa) = \sqrt{\max(1 - \alpha^2(x - aa)^2, 0)}.$$

We have run the third-order DG scheme with the two different limitation in the case of the linear advection of the signal described below. The results are displayed in Figure 1.13. Both solutions are monotonic but we clearly see the difference in the accuracy preservation between these two slope limiter, the solution corresponding to the hierarchical vertex-based limitation being more close to the analytical solution.

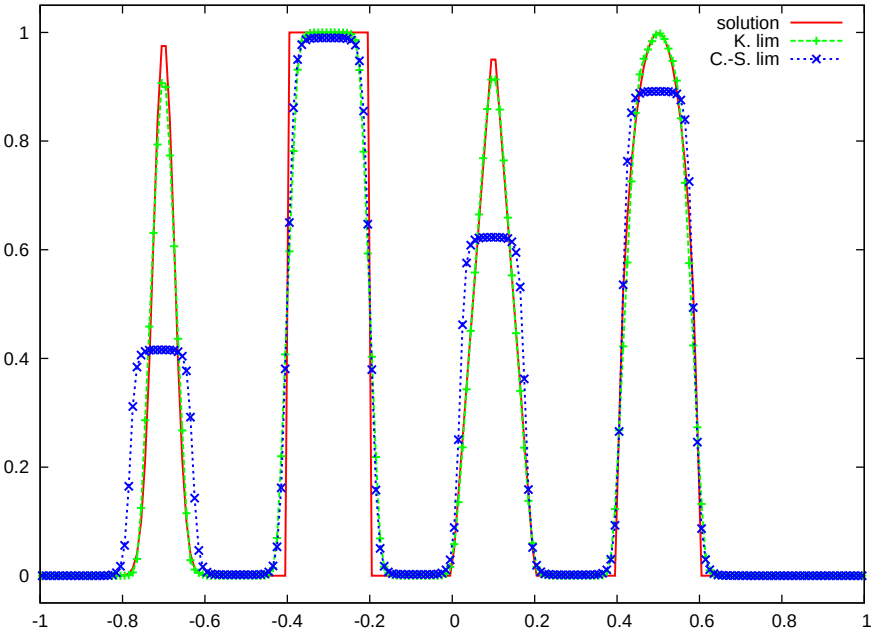


Figure 1.13: Linear advection ($a = 1$) of the composite signal, with periodic boundary conditions, at the end of ten periods with a CFL=0.2 and on $[-1, 1]$ using a grid characterized by 200 uniformly spaced cells. Comparison between the analytical solution and the numerical solutions obtained with the third-order DG scheme with the Cockburn and Shu limitation and with the hierarchical vertex-based limitation.

To confirm the efficiency of the hierarchical vertex-based limitation procedure, we compute the rates of convergence reusing the same test cases than previously, namely, a linear advection case and the Burgers case. In both cases, the numerical orders computed are close to three as expected, the asymptotic regime being reached with more refined meshes. Comparing Table 1.9 and Table 1.10 with Table 1.11 and Table 1.12, we can also note that a small loss of accuracy. This comes from the initial condition which is a sinus function and not a quadratic function. Consequently, the limitation spoils slightly the solution.

	L_1		L_2		L_∞	
Δx	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{100}$	6.31E-6	3.33	1.68E-5	2.97	7.58E-5	2.51
$\frac{1}{200}$	6.26E-7	3.36	2.14E-6	3.00	1.32E-5	2.53
$\frac{1}{400}$	6.07E-8	3.35	2.69E-7	3.03	2.28E-6	2.55
$\frac{1}{800}$	5.95E-9	3.37	3.30E-8	3.06	3.89E-7	2.61
$\frac{1}{1600}$	5.75E-10	-	3.96E-9	-	6.39E-8	-

Table 1.11: Rate of convergence for the third-order DG scheme with hierarchical vertex-based slope limiter computed in the case of linear advection with the smooth initial condition $u^0(x) = \sin(2\pi x)$ on the $[0, 1]$ domain, at time $t = 1$ with a CFL= 0.2.

Finally, to assess the accuracy of the high-order discretization, we have plotted in Figure 1.14 the numerical solutions obtained using respectively the first, second and third order DG discretization for the linear advection test case characterized by the initial condition (1.58).

	L_1		L_2		L_∞	
Δx	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{200}$	2.72E-7	3.21	9.95E-7	3.40	1.07E-5	3.00
$\frac{1}{400}$	2.94E-8	3.10	9.43E-8	3.32	1.34E-6	3.00
$\frac{1}{800}$	3.42E-9	3.05	9.41E-9	3.23	1.68E-7	3.00
$\frac{1}{1600}$	4.13E-10	3.02	1.00E-9	3.15	2.10E-8	3.00
$\frac{1}{3200}$	5.09E-11	-	1.13E-10	-	2.63E-9	-

Table 1.12: Rate of convergence for the third-order DG scheme with hierarchical vertex-based slope limiter computed in the case of Burgers equation with the smooth initial condition $u^0(x) = \sin(2\pi x)$ on the $[0, 1]$ domain, at time $t = \frac{t_c}{2}$ with a CFL= 0.2.

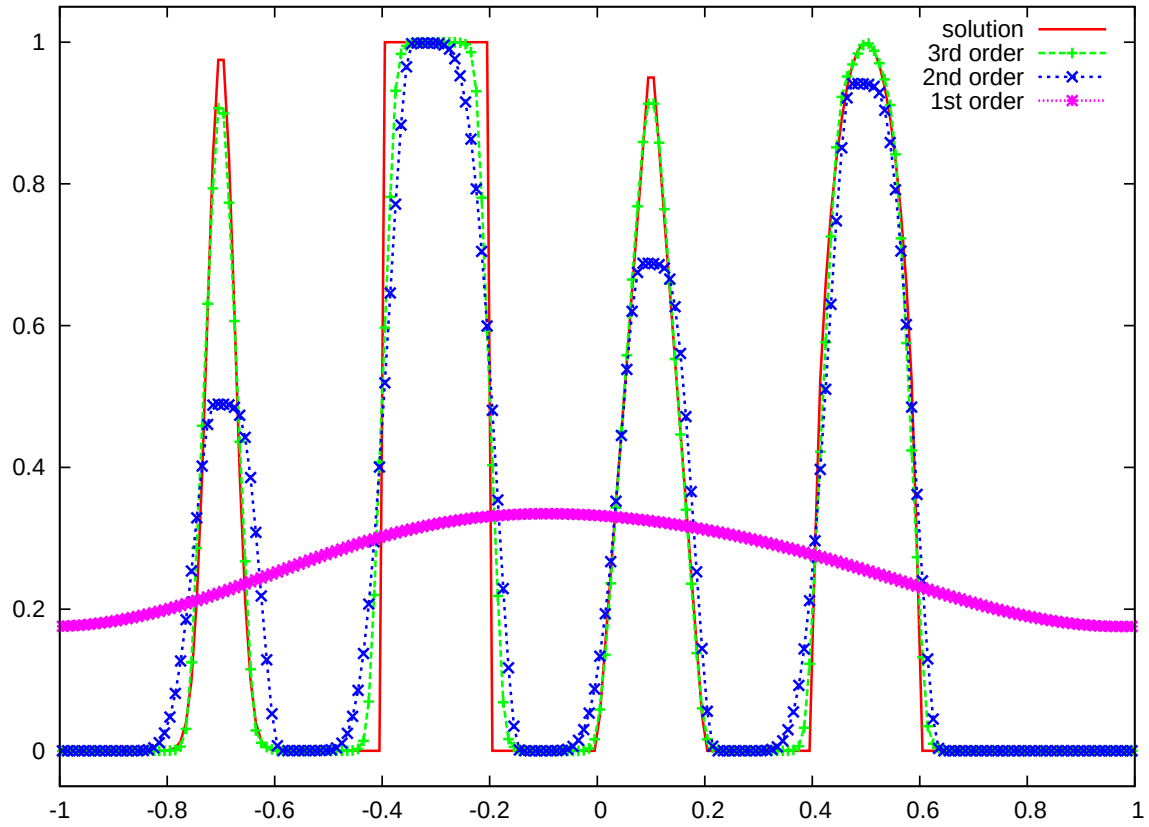


Figure 1.14: Linear advection ($a = 1$) of the composite signal (1.58), with periodic boundary conditions, at the end of ten periods with a CFL=0.2 and on $[-1, 1]$ with a grid characterized by 200 uniformly spaced cells. Comparison between the analytical solution and the numerical solutions obtained with the first, second and third-order DG schemes with the hierarchical vertex-based slope limiter.

1.2 Two-dimensional scalar conservation laws

This section aims at describing the two-dimensional extension on unstructured grids of the DG discretization presented in the last section. In [25] Cockburn and Shu introduced the two-dimensional extension of their RKDG methods to hyperbolic scalar conservation laws. The main difficulty lies in the fact that, on the contrary to the one-dimensional case, it is impossible to devise high-order accurate schemes with the TVD (total variation diminishing) property. Therefore, they content themselves in deriving maximum principles for the RKDG methods which is even a very attractive property for this kind of problems. Moreover, they show that their method does not destroy the high-order accuracy of the scheme and preserves local maximum principle.

In what follows, we shall describe DG discretizations for solving two-dimensional hyperbolic conservation laws on general unstructured grids. The main feature of our DG discretization lies in the use of a polynomial space approximation spanned by a Taylor basis. This type of basis allows to cope with general unstructured grids made of polygonal cells. However, this polynomial basis is in general non-orthogonal and thus leads to a non-diagonal mass matrix but still invertible. We enforce the monotonicity by extending the hierarchical vertex-based slope limiter that has been introduced in the one-dimensional case. Finally, we assess the robustness and the accuracy of our DG discretization by means of various demanding test cases taken from the literature.

We start by introducing some notation. Let ω be an open subset of $\mathbb{R}^2$ and $u = u(\boldsymbol{x}, t)$ be the solution of the following two-dimensional scalar conservation law for $\boldsymbol{x} = (x, y)^{\mathrm{t}} \in \mathbb{R}^2$ and $t \in [0, T]$

$$\frac{\partial u}{\partial t} + \nabla \cdot \boldsymbol{f}(u) = 0, \tag{1.59a}$$

$$u(\boldsymbol{x}, 0) = u^0(\boldsymbol{x}), \tag{1.59b}$$

where u^0 is the initial data and $\boldsymbol{f}(u) = (f_1(u), f_2(u))^{\mathrm{t}}$ with $f_1(u)$ and $f_2(u)$ are the two directional fluxes. For the subsequent discretization, let $\{\omega_c\}_c$ denotes a partition of the domain ω into non-overlapping polygonal cells. Let also $\{t^n\}_n$ be a partition of $[0, T]$ and $\Delta t^n = t^{n+1} - t^n$ the time step.

1.2.1 Discontinuous Galerkin discretization

The DG scheme applied to the two-dimensional scalar conservation laws equation is obtained following the same procedure than the one used in the one-dimensional case. Thus, we perform a local variational formulation of the equation (1.59a). As for the one-dimensional case, we want to develop our function u onto $\mathbb{P}^{\alpha}(\omega_c)$, the set of polynomials of degree up to α. This space approximation leads to a $(\alpha+1)^{th}$ order accurate scheme in space. Let u_h^c be the restriction of u_h, the polynomial approximation of the function u, to cell ω_c at time t

$$u_h^c(\boldsymbol{x}, t) = \sum_{k=0}^{K} u_k^c(t)\, \sigma_k^c(\boldsymbol{x}), \quad \boldsymbol{x} \in \omega_c, \tag{1.60}$$

where the u_k^c are the $K+1$ successive components of u_h^c over the polynomial basis, and σ_k^c the polynomial basis functions. Recalling that the dimension of the polynomial space $\mathbb{P}^{\alpha}(\omega_c)$ is $\frac{(\alpha+1)(\alpha+2)}{2}$, we have to determine the set of the $\frac{(\alpha+1)(\alpha+2)}{2} = K+1$ polynomial function

components. These coefficients $u_k^c(t)$ are determined by writing the local variational formulation of (1.59a) with the test function σ_q^c chosen into the basis, for $q = 0, \ldots, K$

$$\begin{aligned}\int_{\omega_c} \frac{\partial u}{\partial t}\, \sigma_q^c \, \mathrm{d}v &= -\int_{\omega_c} \sigma_q^c \, \nabla \cdot \boldsymbol{f} \, \mathrm{d}v, \\ &= \int_{\omega_c} \boldsymbol{f}(u) \cdot \nabla \sigma_q^c \, \mathrm{d}v - \int_{\partial\omega_c} \sigma_q^c \, \overline{\boldsymbol{f}}(u) \cdot \boldsymbol{n} \mathrm{d}s.\end{aligned}$$

Here, $\overline{\boldsymbol{f}}(u)$ is the numerical flux which is a function defined at the cell interfaces and in general depends on the values of the numerical solution from both sides of the interface. Interchanging the function u by its polynomial approximation u_h^c on the cell ω_c leads to $K+1$ semi-discrete equations for the same number of unknowns

$$\sum_{k=0}^{K} \left(\int_{\omega_c} \sigma_k^c \, \sigma_q^c \, \mathrm{d}v \right) \frac{\mathrm{d}\, u_k^c}{\mathrm{d}t} = \int_{\omega_c} \boldsymbol{f}(u_h^c) \cdot \nabla \sigma_q^c \, \mathrm{d}v - \int_{\partial\omega_c} \sigma_q^c \, \overline{\boldsymbol{f}}(u) \cdot \boldsymbol{n} \mathrm{d}s. \tag{1.61}$$

In equations (1.61), we identify $(\int_{\omega_c} \sigma_k^c \, \sigma_q^c \, \mathrm{d}v)$ as the coefficients of the local mass matrix M^c. For the same reasons than in the one-dimensional discretization part, the mass matrix is symmetric positive definite and thus invertible. We notice that this mass matrix is locally defined over each cell and is time independent.

To get a compact form of the $K+1$ semi-discrete equations (1.61), we firstly rewrite these equations separating the two directions x and y of the flux function $\boldsymbol{f}(u) = (f_1(u), f_2(u))^{\mathrm{t}}$

$$\sum_{k=0}^{K} M_{qk}^c \frac{\mathrm{d}\, u_k^c}{\mathrm{d}t} = \int_{\omega_c} f_1(u_h^c) \frac{\partial \sigma_q^c}{\partial x} \, \mathrm{d}v + \int_{\omega_c} f_2(u_h^c) \frac{\partial \sigma_q^c}{\partial y} \, \mathrm{d}v - \int_{\partial\omega_c} \sigma_q^c \, \overline{\boldsymbol{f}}(u) \cdot \boldsymbol{n} \mathrm{d}s. \tag{1.62}$$

The next step towards a compact form is the use of the projection operator Π over $\mathbb{P}^{\,\alpha}$ defined in Section 1.1.6 by the equation (1.23), (1.24) and (1.25), *i.e.*,

$$\begin{aligned}\Pi \, : \, L^1(\omega_c) &\longrightarrow \mathbb{P}^{\,\alpha}(\omega_c) \\ \phi &\longrightarrow \Pi(\phi) = \phi_h^c.\end{aligned}$$

The components of the polynomial approximation, ϕ_h^c, are the solution of the linear system

$$\mathsf{M}^c \, \boldsymbol{\Phi}^c = \widetilde{\boldsymbol{\Phi}}^c, \tag{1.63}$$

where $\boldsymbol{\Phi}^c = (\phi_0^c, \ldots, \phi_K^c)^{\mathrm{t}}$ and $\widetilde{\boldsymbol{\Phi}}^c = (\langle \phi, \sigma_0^c \rangle, \ldots, \langle \phi, \sigma_K^c \rangle)^{\mathrm{t}}$. The mass matrix being invertible, this system always possesses a unique solution. As we did in the implementation of the one-dimensional DG scheme, we define the polynomial flux functions approximations $f_{1,h}^c$ and $f_{2,h}^c$ by projecting onto the polynomial approximation space the functions $f_1(u_h^c)$ and $f_2(u_h^c)$. We set the vectors $\boldsymbol{F}_1^c \in \mathbb{R}^{K+1}$ and $\boldsymbol{F}_2^c \in \mathbb{R}^{K+1}$ solutions of the systems

$$\mathsf{M}^c \, \boldsymbol{F}_j^c = \widetilde{\boldsymbol{F}}_j^c,$$

where $j = 1$ or $j = 2$, and the vectors $\boldsymbol{F}_j^c$ and $\widetilde{\boldsymbol{F}}_j^i$ are defined as

$$\begin{aligned}\boldsymbol{F}_j^c &= (f_{j,0}^c, \ldots, f_{j,K}^c)^{\mathrm{t}}, \\ \widetilde{\boldsymbol{F}}^c &= (\langle f(u_h^c), \sigma_0^c \rangle, \ldots, \langle f(u_h^c), \sigma_K^c \rangle)^{\mathrm{t}}.\end{aligned}$$

Consequently, with these definitions, the internal terms rewrite

$$\int_{\omega_c} f_1(u_h^c)\,\frac{\partial\,\sigma_q^c}{\partial x}\,\mathrm{d}v + \int_{\omega_c} f_2(u_h^c)\,\frac{\partial\,\sigma_q^c}{\partial y}\,\mathrm{d}v = \sum_{k=0}^{K}\left[\left(\int_{\omega_c}\sigma_k^c\,\frac{\partial\,\sigma_q^c}{\partial x}\,\mathrm{d}v\right) f_{1,k}^c + \left(\int_{\omega_c}\sigma_k^c\,\frac{\partial\,\sigma_q^c}{\partial y}\,\mathrm{d}v\right) f_{2,k}^c\right].$$

Introducing the matrices D_1^c and D_2^c defined as

$$\mathsf{D}_{1,qk}^c = \int_{\omega_c}\sigma_k^c\,\frac{\partial\,\sigma_q^i}{\partial x}\,\mathrm{d}v \quad \text{and} \quad \mathsf{D}_{2,qk}^c = \int_{\omega_c}\sigma_k^c\,\frac{\partial\,\sigma_q^i}{\partial y}\,\mathrm{d}v,$$

the internal term can be identified as the sum of the qth row of the matrix-vector product $\mathsf{D}_1^c\,\boldsymbol{F}_1^c$ and $\mathsf{D}_2^c\,\boldsymbol{F}_2^c$. And finally, the $K+1$ semi-discrete equations (1.61) can be written in the following compact form

$$\mathsf{M}^c\,\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}^c = \mathsf{D}_1^c\,\boldsymbol{F}_1^c + \mathsf{D}_2^c\,\boldsymbol{F}_2^c - \int_{\partial\omega_c}\boldsymbol{\sigma^c}(\boldsymbol{x})\;\left(\overline{\boldsymbol{f}}(u)\,.\,\boldsymbol{n}\right)\mathrm{d}s, \tag{1.64}$$

where $\boldsymbol{\sigma}^c(\boldsymbol{x}) = (\sigma_0^c(\boldsymbol{x}),\ldots,\sigma_K^c(\boldsymbol{x}))^{\mathrm{t}}$.
To complete our scheme, the numerical fluxes along the interfaces $\overline{\boldsymbol{f}}(u)$ have to be defined. To do so, we use the same procedure that in the one-dimensional frame imposing the global L_2 stability to the solution.

1.2.2 Numerical flux and L_2 stability

Following the approach described in [24], we investigate if it is possible to define the numerical flux in such a way as to ensure the global L_2 stability of our semi-discrete DG method. First, we perform a local variational formulation using u_h^c as the test function

$$\int_{\omega_c} u_h^c\,\frac{\partial\,u_h^c}{\partial t}\,\mathrm{d}v = \int_{\omega_c}\boldsymbol{f}(u_h^c)\,.\,\nabla u_h^c\,\mathrm{d}v - \int_{\partial\omega_c} u_h^c\,\overline{\boldsymbol{f}}(u)\,.\,\boldsymbol{n}\mathrm{d}s.$$

We can immediately infer

$$\frac{1}{2}\,\frac{\mathrm{d}}{\mathrm{d}t}\int_{\omega_c}(u_h^c)^2\,\mathrm{d}v = \int_{\omega_c}\boldsymbol{f}(u_h^c)\,.\,\nabla u_h^c\,\mathrm{d}v - \int_{\partial\omega_c} u_h^c\,\overline{\boldsymbol{f}}(u)\,.\,\boldsymbol{n}\mathrm{d}s. \tag{1.65}$$

Here, we make use of the function $\boldsymbol{F}(u)$ which denotes a primitive of the flux function defined as $\boldsymbol{F}(u) = \int_0^u \boldsymbol{f}(s)\,\mathrm{d}s = (\int_0^u f_1(s)\,\mathrm{d}s,\ \int_0^u f_2(s)\,\mathrm{d}s)^{\mathrm{t}}$. And noticing $\nabla.\boldsymbol{F}(u) = \frac{\mathrm{d}\,\boldsymbol{F}(u)}{\mathrm{d}u}.\nabla u = \boldsymbol{f}(u).\nabla u$, equation (1.65) writes

$$\frac{1}{2}\,\frac{\mathrm{d}}{\mathrm{d}t}\int_{\omega_c}(u_h^c)^2\,\mathrm{d}v = \int_{\partial\omega_c}\left(\boldsymbol{F}(u_h^c) - u_h^c\,\overline{\boldsymbol{f}}(u)\right)\,.\,\boldsymbol{n}\mathrm{d}s. \tag{1.66}$$

Introducing $R_c = \displaystyle\int_{\partial\omega_c}\left(u_h^c\,\overline{\boldsymbol{f}}(u) - \boldsymbol{F}(u_h^c)\right)\,.\,\boldsymbol{n}\mathrm{d}s$ and summing over all the cells, one gets

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u_h\|_{L_2}^2 = \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\sum_c\int_{\omega_c}(u_h^c)^2\,\mathrm{d}v = -\sum_c R_c. \tag{1.67}$$

At this point, we claim that the global stability in L_2 norm for our semi-discrete scheme amounts to impose

$$\sum_c R_c = \sum_c \sum_{f_e \in \mathcal{F}(c)} \int_{f_e} \left(u_h^c \, \overline{\boldsymbol{f}}(u)_{|_{f_e}} - \boldsymbol{F}(u_h^c) \right) \cdot \boldsymbol{n}_{f_e} \mathrm{d}s \geq 0, \tag{1.68}$$

where $\mathcal{F}(c)$ is the set of faces of the cell ω_c, $\overline{\boldsymbol{f}}(u)_{|_{f_e}}$ the numerical flux defined on face f_e and $\boldsymbol{n}_{f_e}$ the unit outward normal of face f_e.

Next, we determine the form of the numerical flux so that (1.68) is enforced. By interchanging the sum from cells to faces as in Figure 1.15, (1.68) rewrites

$$\sum_c R_c = \sum_{f_e} \int_{f_e} \left[\overline{\boldsymbol{f}}(u)_{|_{f_e}} (u_L - u_R) - (\boldsymbol{F}(u_L) - \boldsymbol{F}(u_R)) \right] \cdot \boldsymbol{n}_{f_e} \mathrm{d}s \geq 0. \tag{1.69}$$

where u_L and u_R denote the extrapolated values at the interface f_e of the variable u_h on both sides of the interface. Namely, if $\boldsymbol{x}_{f_e}$ denotes a point located on f_e, then $u_L = \lim_{\lambda \to 0^+} u_h(\boldsymbol{x}_{f_e} - \lambda \boldsymbol{n}_{f_e})$ and $u_R = \lim_{\lambda \to 0^+} u_h(\boldsymbol{x}_{f_e} + \lambda \boldsymbol{n}_{f_e})$.

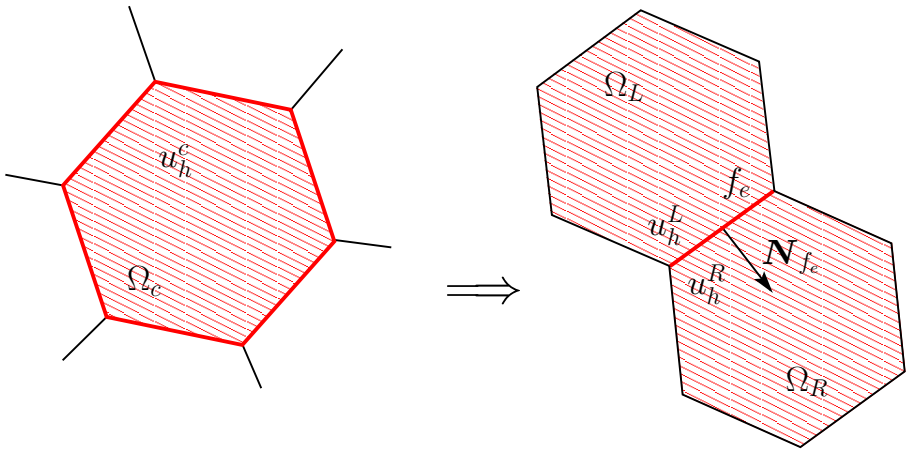


Figure 1.15: Interchange of the sum from cells to faces.

Recalling the definition of the primitive function $\boldsymbol{F}(u) = \int_0^u \boldsymbol{f}(s)\,\mathrm{d}s$, equation (1.69) yields

$$\sum_c R_c = \sum_{f_e} \int_{f_e} (u_L - u_R) \left(\overline{\boldsymbol{f}}(u)_{|_{f_e}} - \frac{1}{u_R - u_L} \int_{u_L}^{u_R} \boldsymbol{f}(s)\,\mathrm{d}s \right) \cdot \boldsymbol{n}_{f_e} \mathrm{d}s. \tag{1.70}$$

Finally, the global stability of the semi-discrete scheme in L_2 norm is ensured provided that the numerical flux is written

$$\overline{\boldsymbol{f}}(u)_{|_{f_e}} = \frac{1}{u_R - u_L} \int_{u_L}^{u_R} \boldsymbol{f}(s)\,\mathrm{d}s - (u_L - u_R) \mathsf{M}_{f_e} \boldsymbol{n}_{f_e}, \tag{1.71}$$

where M_{f_e} is a positive definite matrix which has the physical dimension of a velocity and which characterizes the numerical viscosity of the scheme. Finally, the sufficient condition

used to determined the numerical fluxes (1.71) gives the global L_2 stability of the semi-discrete scheme

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u_h\|_{L_2}^2 = -\sum_{f_e}(u_R - u_L)^2(\mathsf{M}_{f_e}\boldsymbol{n}_{f_e}\,.\,\boldsymbol{n}_{f_e}) \leq 0. \tag{1.72}$$

The numerical dissipation of the semi-discrete discontinuous Galerkin scheme is directly related to the squares of the polynomials approximations jumps located at the interfaces.
In the linear case, $\boldsymbol{f}(u) = \boldsymbol{A}\,u$, where $\boldsymbol{A}$ is the constant advection velocity. Recalling (1.71) in this case, one gets

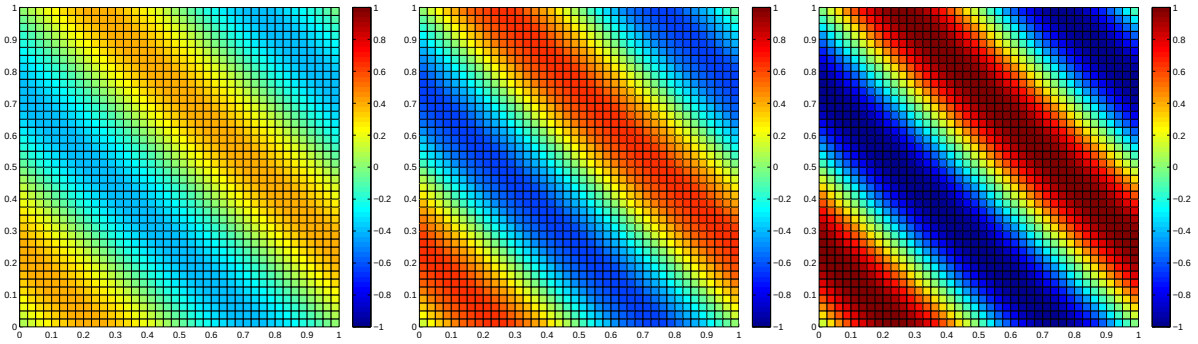
$$\overline{\boldsymbol{f}}(u)_{|_{f_e}} = \frac{u_L + u_R}{2}\,\boldsymbol{A} - (u_R - u_L)\mathsf{M}_{f_e}\boldsymbol{n}_{f_e}. \tag{1.73}$$

Again, we are able to identify two different parts in this flux, the centered one $\frac{u_L+u_R}{2}\boldsymbol{A}$, and the viscous one $(u_R - u_L)\mathsf{M}_{f_e}\boldsymbol{n}_{f_e}$ which brings dissipation and consequently stability. We also note that by setting $\mathsf{M}_{f_e} = \mathsf{M}_{f_e}^{(1)} = \frac{1}{2}|\boldsymbol{A}\,.\,\boldsymbol{n}_{f_e}|\,\mathsf{I}_d$, we recover the classical upwind scheme.
The choice of the matrix M_{f_e} enables us to build a large family of schemes respecting the global L_2 stability of the solution. And according to this matrix, the scheme will be more or less accurate. For example, setting $\mathsf{M}_{f_e} = \mathsf{M}_{f_e}^{(2)} = \frac{1}{2}|\boldsymbol{A}\,.\,\boldsymbol{n}_{f_e}|\,\dfrac{\boldsymbol{A}\otimes\boldsymbol{A}}{\|\boldsymbol{A}\|^2}$, we define a less dissipative scheme since its dissipative contribution is less important than the one resulting from the upwind scheme since

$$\frac{\mathsf{M}_{f_e}^{(2)}\boldsymbol{n}_{f_e}\,.\,\boldsymbol{n}_{f_e}}{\mathsf{M}_{f_e}^{(1)}\boldsymbol{n}_{f_e}\,.\,\boldsymbol{n}_{f_e}} = (\cos\theta)^2 \leq 1,$$

where θ is the angle between the advection velocity $\boldsymbol{A}$ and the normal $\boldsymbol{n}_{f_e}$. This scheme also preserves the non-dissipative property in orthogonal directions to the advection velocity of the upwind scheme.

To compare the two first-order schemes constructed respectively with the matrices $\mathsf{M}_{f_e}^{(1)}$ and $\mathsf{M}_{f_e}^{(2)}$, we shall compute the numerical solution of the linear advection equation characterized by $\boldsymbol{A} = (1,1)^{\mathrm{t}}$ and the initial condition $u^0(x,y) = \sin(2\pi(x+y))$ during one period We clearly observe in Figure 1.16, Figure 1.17 and Figure 1.18 that the scheme resulting from the second dissipation matrix is less diffusive and thus more accurate than the classical upwind scheme.
Let us analyze the results obtained in the frame of this numerical fluxes study. As expected, the scheme characterized by $\mathsf{M}_{f_e}^{(2)}$ is less diffusive and consequently more accurate than the classical upwind scheme. The outstanding difference in the scatter plots Figure 1.17 and Figure 1.19 depicts very well this phenomenon. We could also notice that the accuracy difference between the two schemes is less relevant on the polygonal grid than on the Cartesian grid. The explanation is simple, in the case where the Cartesian grid would be aligned with the linear advection direction the results obtained with the two different schemes would be identical. Here, the angle between the advection velocity $\boldsymbol{A}$ and the grid direction is optimal to observe the difference of the diffusive contribution of the two numerical fluxes, refer to Figure 1.16. In the case of a totally unstructured polygonal grid, refer to Figure 1.18, the diffusive error being uniformly distributed through the mesh the results obtained with the two different schemes are closer than in the previous case. Obviously, other possible choices for the viscous matrix would leads to different results.



(a) First-order scheme using $\mathsf{M}_{f_e}^{(1)}$. (b) First-order scheme using $\mathsf{M}_{f_e}^{(2)}$. (c) Exact solution.

Figure 1.16: Linear advection ($\boldsymbol{A} = (1,1)^{\mathrm{t}}$) of the smooth signal $u^0(\boldsymbol{x}) = \sin(2\pi(x+y))$ where $\boldsymbol{x} = (x,y)^{\mathrm{t}} \in [0,1]^2$, with periodic boundary conditions, at the end of a period on a Cartesian grid made of 40×40 cells. Comparison between the analytical solution and the numerical solutions obtained with the first-order discretization and the numerical fluxes based on $\mathsf{M}_{f_e}^{(1)}$ and $\mathsf{M}_{f_e}^{(2)}$.

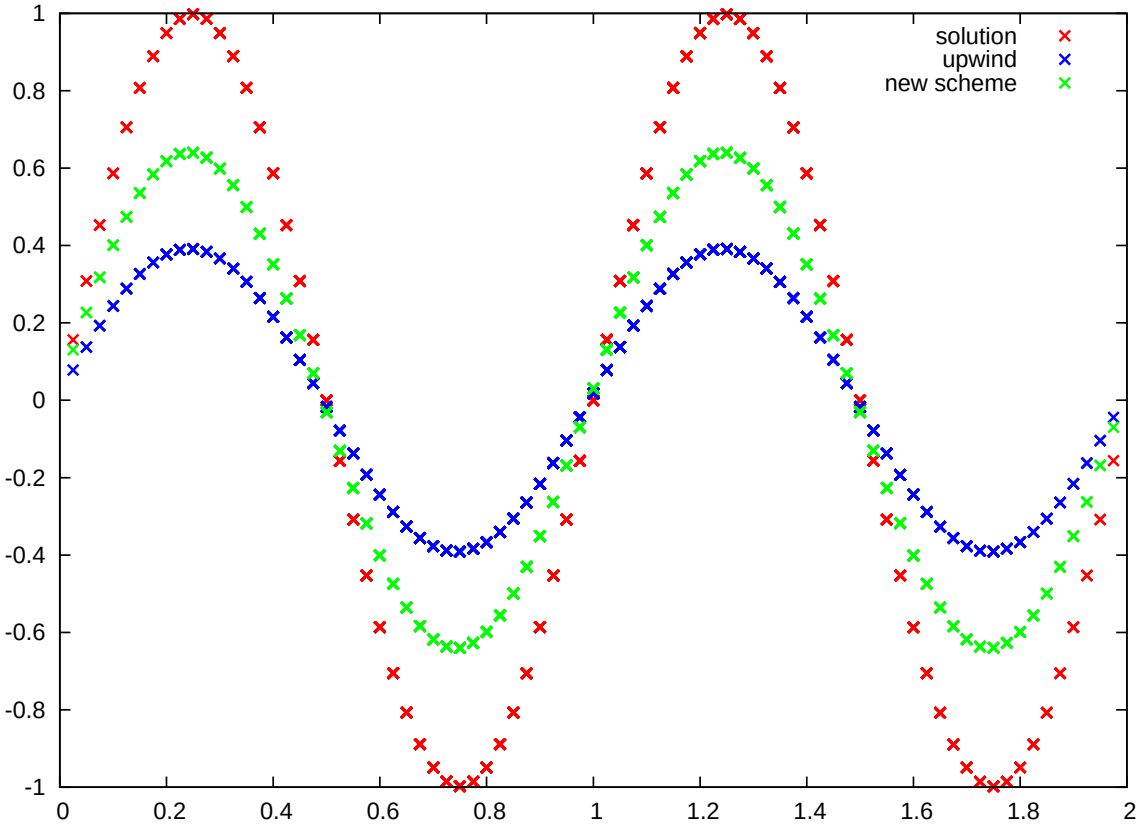
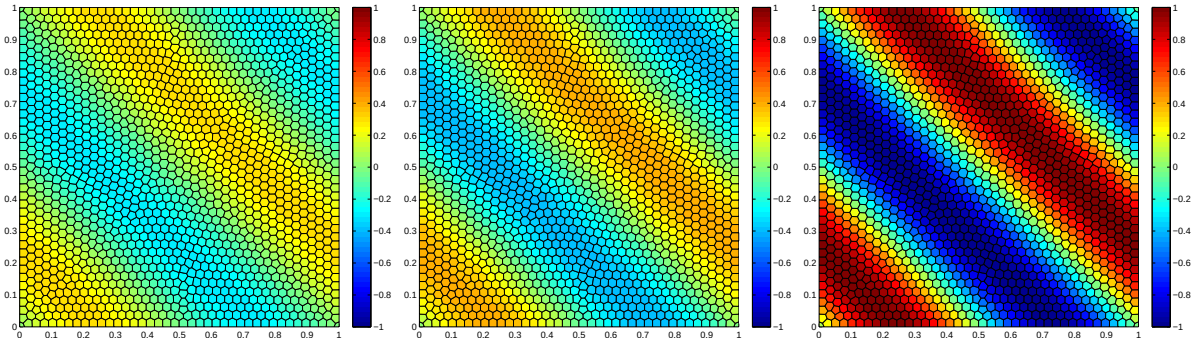


Figure 1.17: Scatter plots of the solutions (ordinate) of the linear advection of the smooth signal $u^0(\boldsymbol{x}) = \sin(2\pi(x+y))$ versus the cumulated coordinate $x+y$ (abscissa). Comparison between the analytical solution and the numerical solutions obtained with the first-order discretization and the numerical fluxes based on $\mathsf{M}_{f_e}^{(1)}$ and $\mathsf{M}_{f_e}^{(2)}$.



(a) First-order scheme using $\mathsf{M}_{f_e}^{(1)}$. (b) First-order scheme using $\mathsf{M}_{f_e}^{(2)}$. (c) Exact solution.

Figure 1.18: Linear advection ($\boldsymbol{A} = (1,1)^{\mathrm{t}}$) of the smooth signal $u^0(\boldsymbol{x}) = \sin(2\pi(x+y))$ where $\boldsymbol{x} = (x,y)^{\mathrm{t}} \in [0,1]^2$, with periodic boundary conditions, at the end of a period on an unstructured grid made of 1600 polygonal cells. Comparison between the analytical solution and the numerical solutions obtained with the first-order discretization and the numerical fluxes based on $\mathsf{M}_{f_e}^{(1)}$ and $\mathsf{M}_{f_e}^{(2)}$.

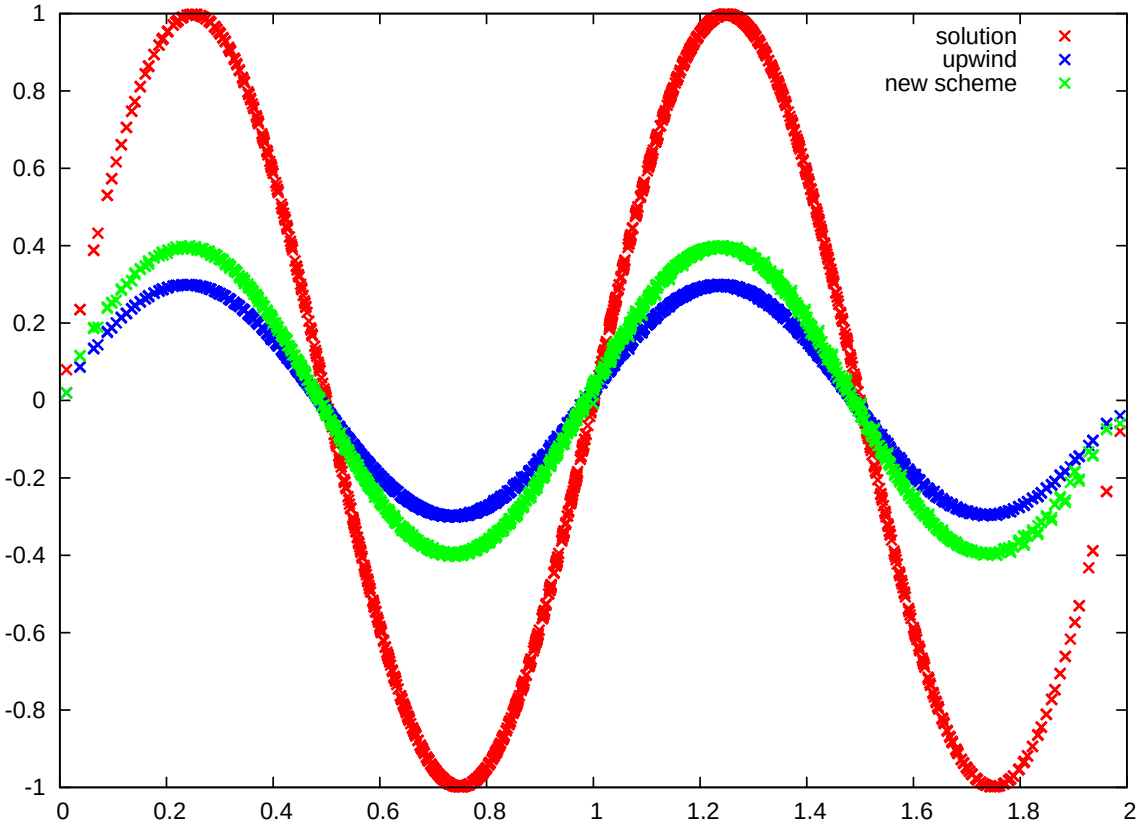


Figure 1.19: Scatter plots of the solutions (ordinate) of the linear advection of the smooth signal $u^0(\boldsymbol{x}) = \sin(2\pi(x+y))$ versus the cumulated coordinate $x+y$ (abscissa). Comparison between the analytical solution and the numerical solutions obtained with the first-order discretization and the numerical fluxes based on $\mathsf{M}_{f_e}^{(1)}$ and $\mathsf{M}_{f_e}^{(2)}$.

In the general non-linear case, as in the one-dimensional study, a quadrature formula can be used to evaluate the integral $\int_{u_L}^{u_R} \boldsymbol{f}(s)\,\mathrm{d}s$. Taking the same trapezoidal rule, setting the viscous matrix to $\mathsf{M}_{f_e} = \frac{1}{2}\max\left(|\boldsymbol{A}(u_L)\cdot\boldsymbol{n}_{f_e}|, |\boldsymbol{A}(u_R)\cdot\boldsymbol{n}_{f_e}|\right)$ we recover the local Lax-Friedrichs scheme

$$\overline{\boldsymbol{f}(u)}_{|f_e}\cdot\boldsymbol{n}_{f_e} = \frac{1}{2}(\boldsymbol{f}(u_L)+\boldsymbol{f}(u_R))\cdot\boldsymbol{n}_{f_e} - \max(|\boldsymbol{A}(u_L)\cdot\boldsymbol{n}_{f_e}|, |\boldsymbol{A}(u_R)\cdot\boldsymbol{n}_{f_e}|)(\frac{u_R-u_L}{2}). \quad (1.74)$$

where $\boldsymbol{A}(u) = \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}u}(u) = (f_1'(u), f_2'(u))^{\mathrm{t}}$.

Due to the simplicity of its implementation, we made the choice of the local Lax-Friedrichs flux for the numerical applications. Again, we notice that if $\boldsymbol{f}(u) = \boldsymbol{A}\,u$, where $\boldsymbol{A}$ is a constant velocity, the local Lax-Friedrichs flux reduces to the classical upwind flux.

1.2.3 Two-dimensional Taylor basis

For each cells ω_c we want to develop our unknowns onto $\mathbb{P}^{\,\alpha}(\omega_c)$, the set of polynomials of degree less than α. This space approximation leads to a $(\alpha+1)^{th}$ space order accurate scheme. Let u_h^c be the restriction of u_h, the polynomial approximation of the function u, over the cell ω_c

$$u_h^c(\boldsymbol{x},t) = \sum_{k=0}^{K} u_k^c(t)\,\sigma_k^c(\boldsymbol{x}),$$

where the u_k^c are the $K+1$ successive components of u_h over the polynomial basis, and σ_k^c the polynomial basis functions. Recalling that the dimension of the polynomial space $\mathbb{P}^{\,\alpha}(\omega_c)$ is $\frac{(\alpha+1)(\alpha+2)}{2}$, we have to determine the set of the $\frac{(\alpha+1)(\alpha+2)}{2} = K+1$ basis functions. We make the choice of the two-dimensional extension of the Taylor basis defined in Section 1.1.8, which comes from a Taylor expansion on the cell, located at the centroid $\boldsymbol{x}_c$ of the cell ω_c defined as following

$$\boldsymbol{x}_c = \frac{1}{|\omega_c|}\int_{\omega_c} \boldsymbol{x}\,\mathrm{d}v, \quad (1.75)$$

where $|\omega_c|$ is the volume of the cell ω_c.
We set the first basis element to 1, *i.e.*, $\sigma_0^c = 1$. Going further in space discretization, the $q+1$ basis functions of degree q, with $0 < q \leq \alpha$, write

$$\sigma^c_{\frac{q(q+1)}{2}+j} = \frac{1}{j!(q-j)!}\left(\frac{x-x_c}{\Delta x_c}\right)^{q-j}\left(\frac{y-y_c}{\Delta y_c}\right)^{j}, \quad \text{for } j = 0\ldots q. \quad (1.76)$$

Here, we have introduced the scaling factors Δx_c and Δy_c in the definition of the polynomial basis in order to improve the condition number of the associated mass matrix. These scaling factors are defined by $\Delta x_c = \frac{x_{\max}-x_{\min}}{2}$ and $\Delta y_c = \frac{y_{\max}-y_{\min}}{2}$ where

$$x_{\max\,/\,\min} = \max\,/\,\min_{p\in\mathcal{P}(c)}\{x_p\},$$
$$y_{\max\,/\,\min} = \max\,/\,\min_{p\in\mathcal{P}(c)}\{y_p\},$$

knowing that $\mathcal{P}(c)$ is the set of vertices of the cell ω_c. Let us point out that the subscript which corresponds to the numbering of the basis function in (1.76) is equal to $\frac{q(q+1)}{2}+j$ for $j = 0,\ldots,q$ since the number of polynomial basis functions of degree strictly less than q is equal to $\frac{q(q+1)}{2}$.

Let us introduce $\langle u \rangle_c$, the mean value of u over the cell ω_c

$$\langle u \rangle_c = \frac{1}{|\omega_c|} \int_{\omega_c} u(\boldsymbol{x}) \, \mathrm{d}v, \tag{1.77}$$

and the associated scalar product

$$\langle u, v \rangle_c = \int_{\omega_c} u(\boldsymbol{X}) \, v(\boldsymbol{X}) \, \mathrm{d}v. \tag{1.78}$$

In our polynomial discretization, we want the mean value to be preserved. Consequently, we identify the first component of u_h^c to $\langle u \rangle_c$, *i.e.*, $u_0^c = \langle u \rangle_c$. This definition leads to a particular constraint on the successive basis functions writing

$$\begin{aligned} u_0^c &= \frac{1}{|\omega_c|} \int_{\omega_c} u_h^c \, \mathrm{d}v, \\ &= \frac{1}{|\omega_c|} \sum_{k=0}^{K} u_k^c \int_{\omega_c} \sigma_k^c \, \mathrm{d}v, \\ &= u_0^c + \sum_{k=1}^{K} u_k^c \, \frac{1}{|\omega_c|} \int_{\omega_c} \sigma_k^c \, \mathrm{d}v, \\ &= u_0^c + \sum_{k=1}^{K} u_k^c \, \langle \sigma_k^c \rangle_c \, . \end{aligned} \tag{1.79}$$

To satisfy equation (1.79), we impose $\langle \sigma_k^c \rangle_c = 0, \ \forall k \neq 0$. Consequently, we set a new definition of the $q+1$ basis functions of degree q, with $0 < q \leq \alpha$

$$\sigma^c_{\frac{q(q+1)}{2}+j} = \frac{1}{j!(q-j)!} \left[\left(\frac{x - x_c}{\Delta x_c} \right)^{q-j} \left(\frac{y - y_c}{\Delta y_c} \right)^{j} - \left\langle \left(\frac{x - x_c}{\Delta x_c} \right)^{q-j} \left(\frac{y - y_c}{\Delta y_c} \right)^{j} \right\rangle_c \right]. \tag{1.80}$$

For the third-order scheme, $\alpha = 2$ and $K = \frac{(\alpha+1)(\alpha+2)}{2} - 1 = 5$

$$\begin{aligned} \sigma_0^c = 1, \, \sigma_1^c = \frac{x - x_c}{\Delta x_c}, \, \sigma_2^c = \frac{y - y_c}{\Delta y_c}, \\ \sigma_3^c = \tfrac{1}{2} \left[\left(\frac{x - x_c}{\Delta x_c} \right)^2 - \left\langle \left(\frac{x - x_c}{\Delta x_c} \right)^2 \right\rangle_c \right], \\ \sigma_4^c = \frac{(x - x_c)(y - y_c)}{\Delta x_c \, \Delta y_c} - \left\langle \frac{(x - x_c)(y - y_c)}{\Delta x_c \, \Delta y_c} \right\rangle_c, \\ \sigma_5^c = \tfrac{1}{2} \left[\left(\frac{y - y_c}{\Delta y_c} \right)^2 - \left\langle \left(\frac{y - y_c}{\Delta y_c} \right)^2 \right\rangle_c \right], \end{aligned} \tag{1.81}$$

since by definition of the centroid, $\langle x - x_c \rangle_c = 0$ and $\langle y - y_c \rangle_c = 0$.
We point out that the only geometry contributions contained in (1.80) and (1.81) are the centroid position $\boldsymbol{x}_c$ and the scaling factors Δx_c and Δy_c. Thus, regardless the shape of the cell, we could use the same class of basis functions. We also note that the first moment u_0 associated to the first basis element 1 is the mean value of the function u over the cell ω_c. If the discretization ends here, the scheme would correspond to the classical Finite Volume method. Going further, the successive moments can be identified as the successive derivatives of the function expressed

at the centroid of the cell. Namely, for the third-order case, we have the following relationships between the components and the derivatives

$$u_1^c \to \Delta x_c \frac{\partial}{\partial x} u_h^c(\boldsymbol{x}_c), \quad u_2^c \to \Delta y_c \frac{\partial}{\partial y} u_h^c(\boldsymbol{x}_c),$$

$$u_3^c \to (\Delta x_c)^2 \frac{\partial^2}{\partial x^2} u_h^c(\boldsymbol{x}_c), \quad u_4^c \to \Delta x_c \Delta y_c \frac{\partial^2}{\partial x \partial y} u_h^c(\boldsymbol{x}_c), \quad u_5^c \to (\Delta y_c)^2 \frac{\partial^2}{\partial y^2} u_h^c(\boldsymbol{x}_c).$$

Mimicking the one-dimensional case, from (1.80), it immediately comes that, for $k = 0, \ldots, K$

$$\langle \sigma_0^c, \sigma_k^c \rangle_c = \int_{\omega_c} \sigma_0^c \, \sigma_k^c \, \mathrm{d}v = \int_{\omega_c} \sigma_k^c \, \mathrm{d}v = |\omega_c| \, \delta_{0k},$$

where δ_{ij} the Kronecker symbol is equal to 1 if $i = j$ and 0 if $i \neq j$. The equation corresponding to the mean value $u_0^c = \langle u \rangle_c$ is independent of the other equations of the polynomial basis components.

We have described the $(\alpha + 1)^{\text{th}}$ order accurate semi-discrete DG scheme with the use of the Taylor basis and the numerical flux formula enforcing a global L_2 stability. To reach a $(K+1)^{\text{th}}$ global accuracy order we shall use the explicit $(K+1)^{\text{th}}$ order Runge-Kutta time discretization presented in Section 1.1.6. We are now able to completely implement this RKDG scheme and assess the accuracy gain between first-order, second-order and third-order, refer to Figure 1.20.

We clearly see on Figure 1.20 and Figure 1.21 a huge gain of accuracy between the three different solutions. Nevertheless, as for the one-dimensional DG schemes, we observe the occurrence of spurious oscillations in the neighborhood of the discontinuities. To cure this problem, we present in the next section a hierarchical vertex-based limitation, which corresponds to the two-dimensional extension of the limitation procedure presented in Section 1.1.9.

1.2.4 Two-dimensional hierarchical vertex-based slope limiter

In this section, we aim at describing the two-dimensional hierarchical vertex-based slope limiter that we use to enforce the monotonicity of our high-order DG scheme. As in the one-dimensional case, we first introduce this slope limiter applied to the second-order DG scheme. In this case the polynomial development of the function u on the cell ω_c at time t reads

$$u_h^c(\boldsymbol{x}, t) = u_0^c + u_1^c \, \frac{x - x_c}{\Delta x_c} + u_2^c \, \frac{y - y_c}{\Delta y_c}.$$

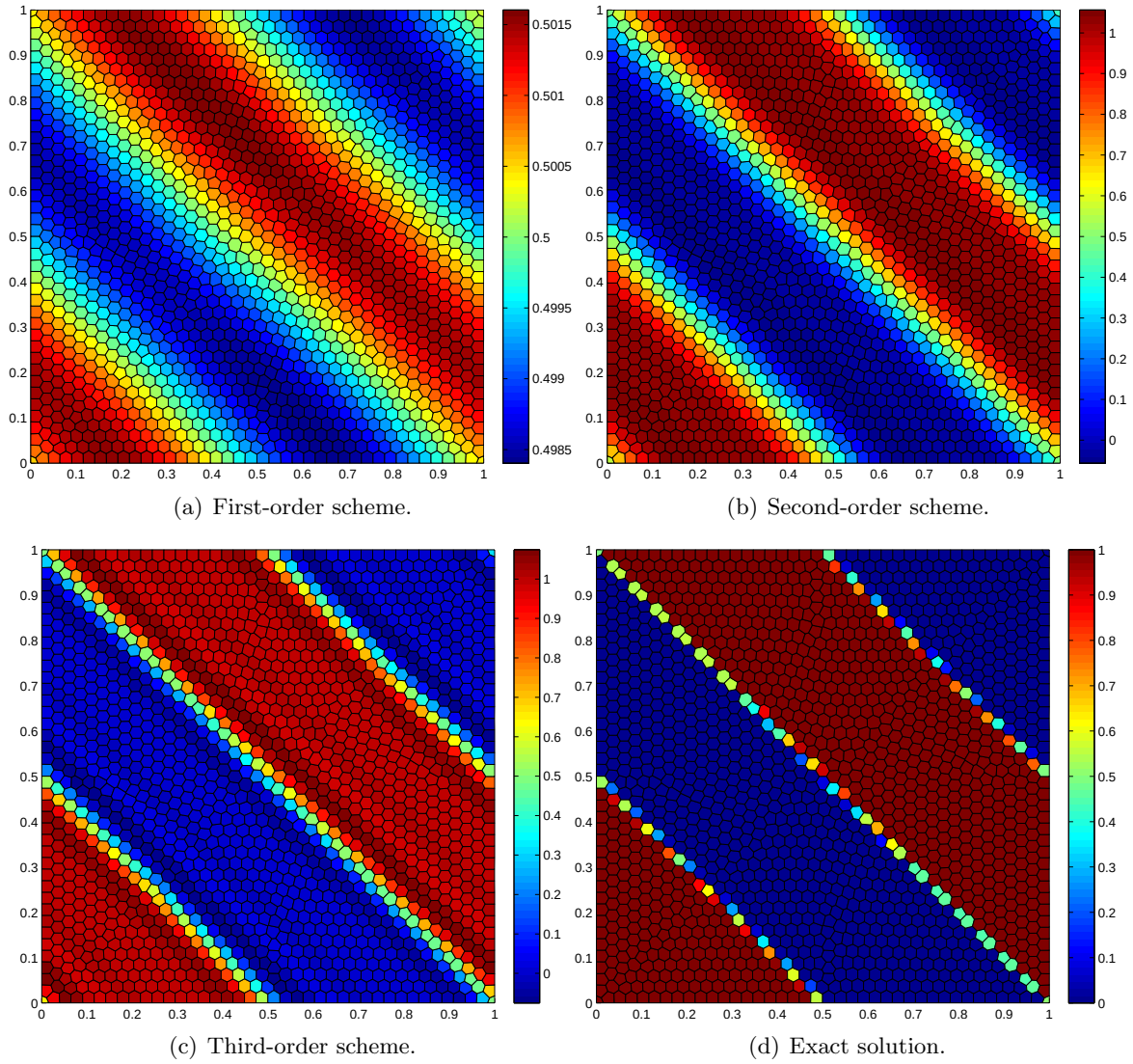
If $\alpha_{(1)}^c \in [0, 1]$ denotes the slope limiter coefficient then the limited solution writes

$$u_h^c(\boldsymbol{x}, t) = u_0^c + \alpha_{(1)}^c \left(u_1^c \, \frac{x - x_c}{\Delta x_c} + u_2^c \, \frac{y - y_c}{\Delta y_c} \right).$$

Barth and Jespersen [5] define the coefficient $\alpha_{(1)}^c$ so that the extrapolated values of u_h^c at the cell vertex $p \in \mathcal{P}(c)$, are bounded by $u_c^{\max}$ and $u_c^{\min}$ which are the maximum and minimum of the cell mean values taken over the neighbors of ω_c sharing a common edge with itself

$$u_c^{\min} \leq u_h^c(\boldsymbol{x}_p) \leq u_c^{\max}, \quad \forall p \in \mathcal{P}(c),$$

where $\boldsymbol{x}_p$ is the position vector of the vertex p.



(a) First-order scheme. (b) Second-order scheme.

(c) Third-order scheme. (d) Exact solution.

Figure 1.20: Linear advection ($\boldsymbol{A} = (1,1)^{\mathrm{t}}$) of the discontinuous signal $u^0(\boldsymbol{x}) = 1$ if $(x+y) \in [0, 0.5] \cup [1.5, 2]$ and null anywhere else, where $\boldsymbol{x} = (x,y)^{\mathrm{t}} \in [0,1]^2$, with periodic boundary conditions, at the end of five periods on an unstructured grid made of 1600 polygonal cells.

Knowing that it is a linear function with respect to the space variable, the solution u_h reaches its local extrema values at the vertices p of ω_c. Hence, a vertex-based limiting strategy is appropriate. This amounts to define the elementwise bounds $u_c^{\max}$ and $u_c^{\min}$ locally at the cell vertex p. To this end, let us define $u_p^{\max}$ and $u_p^{\min}$ being the maximum and minimum mean values in $\mathcal{C}(p)$ the set of the cells surrounding the vertex p

$$u_p^{\max} = \max_{c \in \mathcal{C}(p)} u_0^c,$$
$$u_p^{\min} = \min_{c \in \mathcal{C}(p)} u_0^c.$$

Bearing this modification in mind, we define $\alpha_{(1)}^c$ in such a way that

$$u_p^{\min} \leq u_h^c(\boldsymbol{x}_p) \leq u_p^{\max}, \quad \forall p \in \mathcal{P}(c),$$

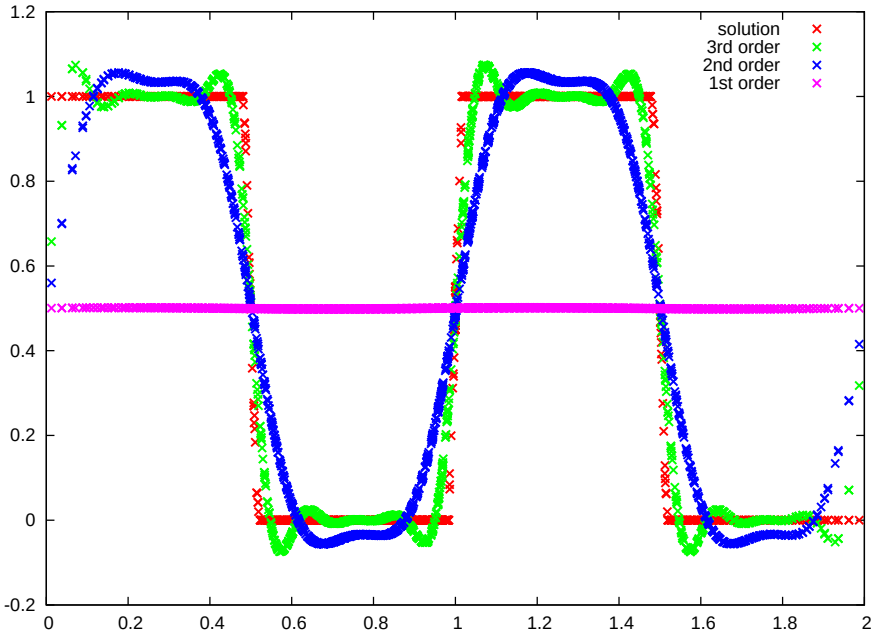


Figure 1.21: Scatter plots of the solutions (ordinate) of the linear advection of the discontinuous signal $u^0(\boldsymbol{x}) = 1$ if $(x+y) \in [0, 0.5] \cup [1.5, 2]$ and null anywhere else, versus the cumulated coordinate $x+y$ (abscissa). Comparison between the analytical solution and those obtained with the first, the second and the third-order DG schemes on an unstructured grid made of 1600 polygonal cells at the end of five periods.

this leads to the formula which is the two-dimensional counterpart of (1.48), (1.49) and (1.50) which reads

$$\alpha_{(1)}^c = \min_{p \in \mathcal{P}(c)} \begin{cases} \min\left(1, \dfrac{u_p^{\max} - u_0^c}{u_h^c(\boldsymbol{x}_p) - u_0^c}\right) & \text{if } u_h^c(\boldsymbol{x}_p) - u_0^c > 0, \\ 1 & \text{if } u_h^c(\boldsymbol{x}_p) = u_0^c, \\ \min\left(1, \dfrac{u_p^{\min} - u_0^c}{u_h^c(\boldsymbol{x}_p) - u_0^c}\right) & \text{if } u_h^c(\boldsymbol{x}_p) - u_0^c < 0. \end{cases} \tag{1.82}$$

Let us point out that the only difference compared to the classical Barth-Jespersen limiter is the use of $u_p^{\max}$ and $u_p^{\min}$ in place of $u_c^{\max}$ and $u_c^{\min}$.

Having defined the slope limiter for the second-order DG method, we pursue by presenting its extension to third-order DG method. Following [109], we multiply all derivatives of order q by a common coefficient $\alpha_{(q)}^c$. For instance, the limited counterpart of the third-order polynomial expansion of the solution reads

$$u_h^c(\boldsymbol{x}, t) = u_0^c + \alpha_{(1)}^c \left(u_1^c \, \sigma_1^c + u_2^c \, \sigma_2^c\right) + \alpha_{(2)}^c \left(u_3^c \, \sigma_3^c + u_4^c \, \sigma_4^c + u_5^c \, \sigma_5^c\right),$$

where $\alpha_{(1)}^c$ and $\alpha_{(2)}^c$ are respectively the limiting coefficients for the first and second-order derivatives. Knowing that u_h^c is a quadratic function with respect to the space variables, its gradient is a linear function with respect to the space variables. Moreover, the components of the gradient of the resulting linear function are nothing but the second-order derivatives of the

initial function

$$\Delta x_c \frac{\partial u_h^c}{\partial x} = u_1^c + \alpha_{(x)}^c \left(u_3^c \frac{x - x_c}{\Delta x_c} + u_4^c \frac{y - y_c}{\Delta y_c} \right), \tag{1.83}$$

$$\Delta y_c \frac{\partial u_h^c}{\partial y} = u_2^c + \alpha_{(y)}^c \left(u_4^c \frac{x - x_c}{\Delta x_c} + u_5^c \frac{y - y_c}{\Delta y_c} \right). \tag{1.84}$$

Here, the limiting coefficients $\alpha_{(x)}^c$ and $\alpha_{(y)}^c$ are determined according to formula (1.82). We notice that we obtain two limiting coefficients for the second-order derivative. Taking the minimum value between these two coefficients leads to define the limiting coefficient as

$$\alpha_{(2)}^c = \min(\alpha_{(x)}^c, \alpha_{(y)}^c)$$

Regarding the limiting coefficient of the first-order derivatives, $\alpha_{(1)}^c$, we compute it by enforcing the monotonicity of the linear function

$$u^{(1)} = u_0^c + \alpha_{(1)}^c \left(u_1^c \frac{x - x_c}{\Delta x_c} + u_2^c \frac{y - y_c}{\Delta y_c} \right). \tag{1.85}$$

To this end, we apply once more formula (1.82). Finally, to ensure the preservation of smooth extrema, we modify this coefficient as follows

$$\alpha_{(1)}^c = \max(\alpha_{(1)}^c, \alpha_{(2)}^c)$$

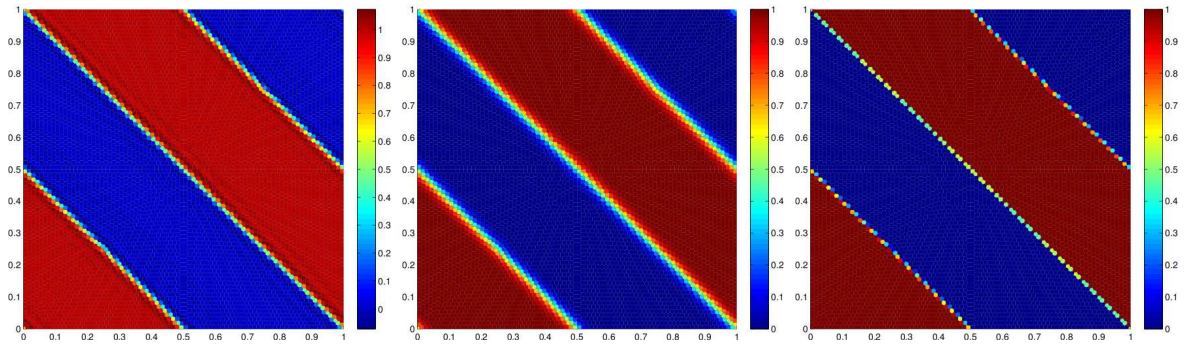
The key point to go to high-order limitation is to implement the limiter as a hierarchical algorithm. In general, we begin with the highest order derivatives and calculate a nondecreasing sequence of correction factors

$$\alpha_{(m)}^c = \max_{m \leq q} \alpha_{(q)}^c, \quad \forall m \geq 1. \tag{1.86}$$

As soon as $\alpha_{(q)}^c = 1$ is encountered, no further limiting is required since definition (1.86) implies that $\alpha_{(m)}^c = 1$ for all $m \leq q$.

Comment 2 *Let us remark that the definition of the correction factor, $\alpha_{(2)}^c$, for the limited quadratic reconstruction is probably too much restrictive. By defining $\alpha_{(2)}^c = \min(\alpha_{(x)}^c, \alpha_{(y)}^c)$, we determine a unique correction factor for the three components of the Hessian matrix of u_h^c, that is the three second-order derivatives $\frac{\partial^2}{\partial x^2} u_h^c$, $\frac{\partial^2}{\partial x \partial y} u_h^c$ and $\frac{\partial^2}{\partial y^2} u_h^c$. Doing so, we are completely ignoring the vectorial structure of the gradient of u_h^c. Indeed, the gradient of u_h^c is a vector valued function which is linear with respect to the space variables. Moreover, its Taylor representation contains a tensor gradient which is nothing but the Hessian matrix of u_h^c evaluated at the centroid $\boldsymbol{x}_c$. Bearing this in mind, it would be interesting to investigate a limiting procedure which takes into account this vectorial structure, refer to [81]. This may improve the accuracy of the high-order DG method particularly in the case of unstructured grids.*

To conclude this section devoted to the hierarchical vertex-based slope limiter, let us investigate its impact on some numerical experiments. We note on Figure 1.22 and Figure 1.23 that with the use of the slope limiter described in this section the solution is perfectly monotonic and preserves the high accuracy of the scheme.
We also compare the Barth-Jespersen limitation and the presented vertex-based hierarchical limitation using a linear advection test case characterized by a velocity field corresponding to a



(a) Without limitation. (b) With limitation. (c) Exact solution.

Figure 1.22: Linear advection ($\boldsymbol{A} = (1,1)^{\mathrm{t}}$) of the discontinuous signal $u^0(\boldsymbol{x}) = 1$ if $(x+y) \in [0, 0.5] \cup [1.5, 2]$ and null anywhere else, where $\boldsymbol{x} = (x,y)^{\mathrm{t}} \in [0,1]^2$, with periodic boundary conditions, at the end of a period on an unstructured grid made of 6400 polygonal cells. Comparison between the analytical solution and the numerical solutions obtained with the third-order DG scheme with and without limitation.

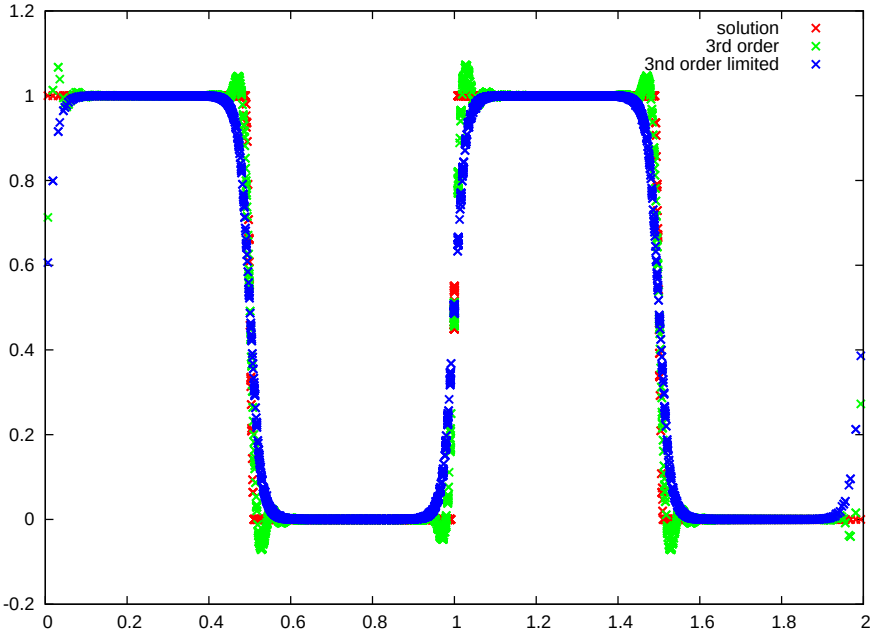


Figure 1.23: Scatter plots of the solutions (ordinate) of the linear advection of the discontinuous signal $u^0(\boldsymbol{x}) = 1$ if $(x+y) \in [0, 0.5] \cup [1.5, 2]$ and null anywhere else, versus the cumulated coordinate $x+y$ (abscissa). Comparison between the analytical solution and the numerical solutions obtained with the third-order DG scheme with and without limitation, on an unstructured grid made of 6400 polygonal cells at the end of a period.

rigid rotation defined by $\boldsymbol{A} = (0.5-y, x-0.5)^{\mathrm{t}}$. We apply this solid body rotation to a smooth hump displayed in Figure 1.24 and defined by

$$u^0(\boldsymbol{x}) = \frac{1}{4}(1 + \cos(\pi r(\boldsymbol{x}))), \tag{1.87}$$

where $r(\boldsymbol{x}) = \min(1, \frac{\sqrt{(x-x_0)^2+(y-y_0)^2}}{r_0})$.

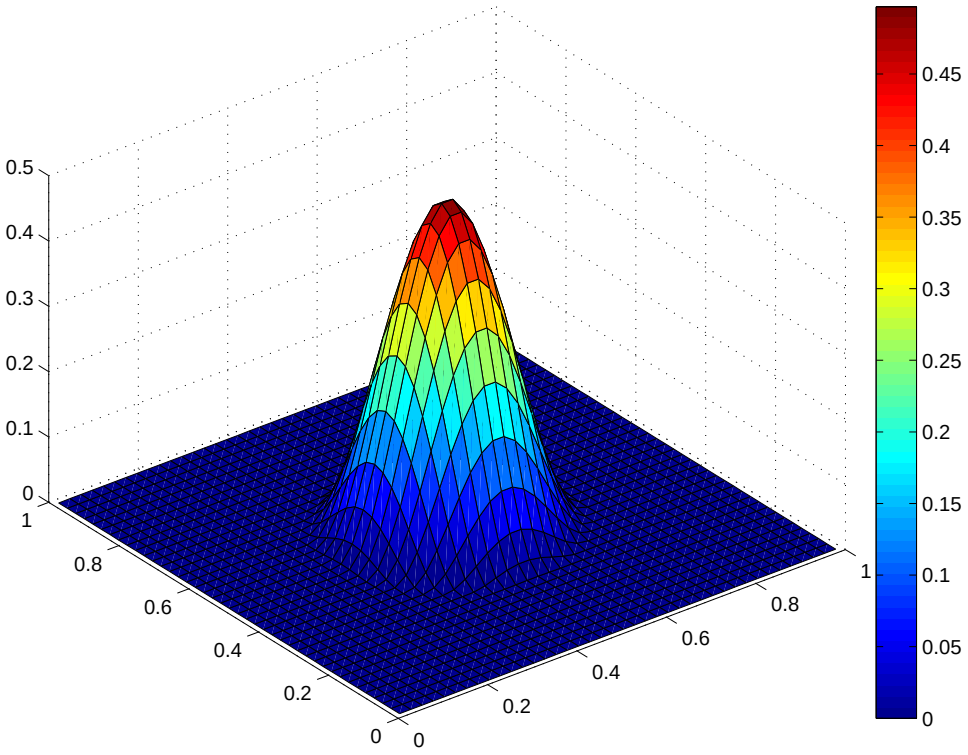


Figure 1.24: Smooth hump signal on a Cartesian grid made of 40×41 cells.

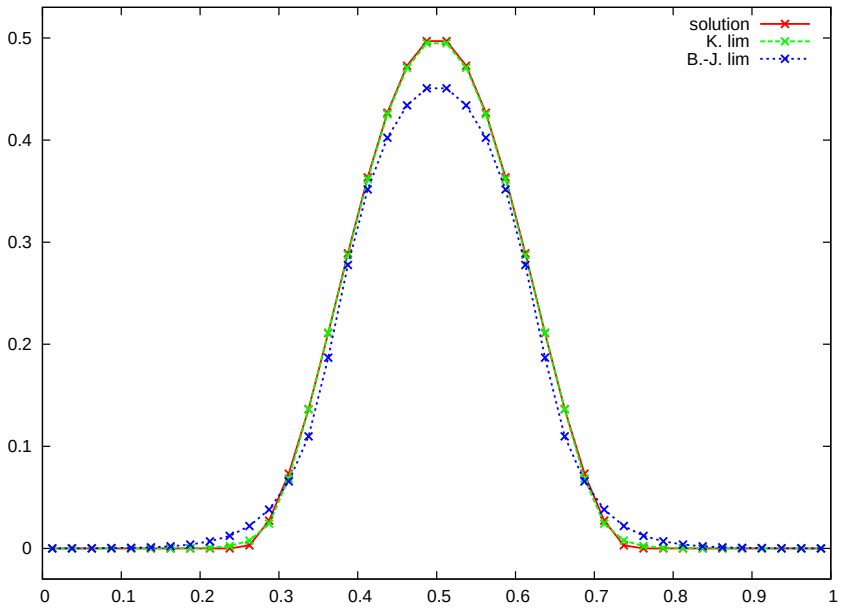


Figure 1.25: Scatter plots of the solutions (ordinate) of the solid body rotation ($\boldsymbol{A} = (0.5 - y, x - 0.5)^{\mathrm{t}}$) of a smooth hump signal, versus the coordinate x (abscissa) for a fixed $y = 0.5$. Comparison between the analytical solution and the numerical solutions obtained with the third-order DG schemes with Barth-Jespersen and vertex-based slope limiters on a Cartesian grid made of 40×40 cells.

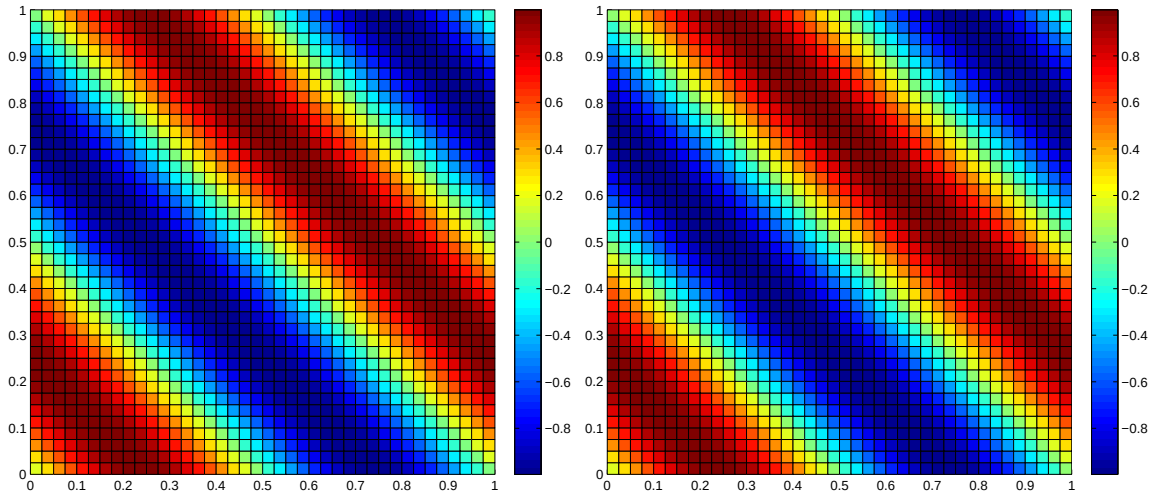
For the computation we use $x_0 = y_0 = 0.5$ and $r_0 = 0.25$, and $t_f = 3\,T$, where t_f is the final time and $T = 2\pi$ is the period of the rotation.
The comparison of the numerical errors shows that the scheme utilizing the vertex-based slope limiter is more accurate than the one using the Barth-Jespersen slope limiter, the errors being respectively $E_{L_1} = 6.31\,10^{-4}$, $E_{L_2} = 1.55\,10^{-3}$, $E_{L_\infty} = 7.26\,10^{-3}$ and $E_{L_1} = 4.78\,10^{-3}$, $E_{L_2} = 8.95\,10^{-3}$, $E_{L_\infty} = 4.92\,10^{-2}$. Moreover, we can observe in Figure 1.25 that the vertex-based slope limiter preserves quite well the smooth extremum.

Finally, we have obtained a DG discretization of two-dimensional scalar conservation laws, up to the third order which employs a hierarchical vertex-based slope limiter. In the next section, we shall perform a convergence analysis of this DG discretization to investigate its accuracy.

1.2.5 Numerical results for two-dimensional scalar conservations laws

Linear test cases

To investigate the accuracy of our third-order DG discretization, we perform various convergence analysis studying the linear advection of the smooth initial condition $u^0(\boldsymbol{x}) = \sin(2\pi(x+y))$ with the velocity field $\boldsymbol{A} = (1,1)^t$. We begin this accuracy study on Cartesian grids such as the ones displayed in Figure 1.26.



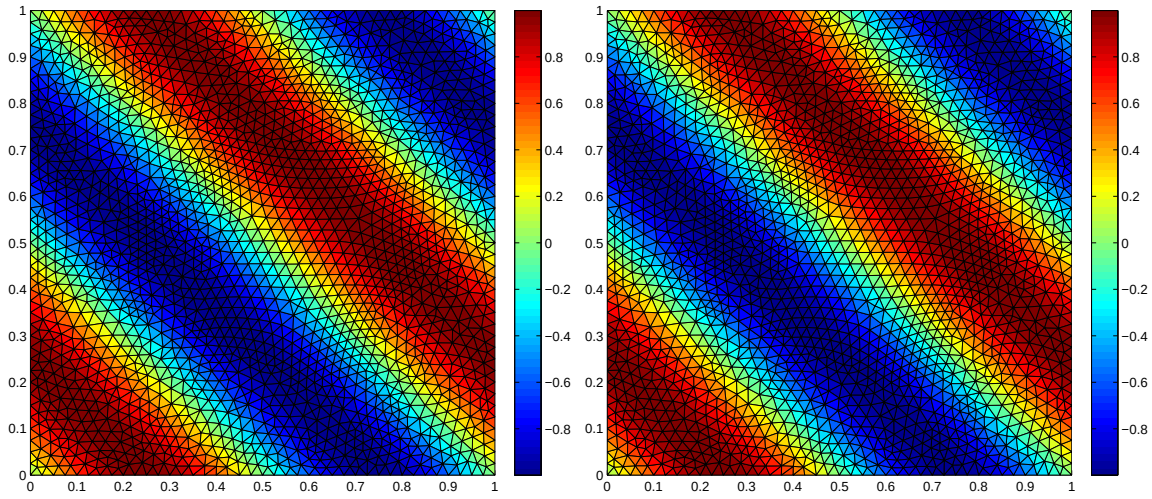
(a) 3rd order scheme. (b) Exact solution.

Figure 1.26: Linear advection ($\boldsymbol{A} = (1,1)^{\mathrm{t}}$) of the smooth initial condition $u^0(\boldsymbol{x}) = \sin(2\pi(x+y))$ where $\boldsymbol{x} = (x,y)^{\mathrm{t}} \in [0,1]^2$, with periodic boundary conditions, at the end of a period on a Cartesian grid made of 40×40 cells.

Observing Table 1.13, we note that our DG scheme reaches exactly the expected order, namely it is characterized by a third-order rate of convergence. Keeping the same test case, we pursue by investigating the accuracy of our DG discretization on unstructured grids made of triangular and polygonal cells, refer respectively to Figure 1.27, and Figure 1.28. The unstructured grids made of triangular cells used in the accuracy analysis not being paved uniformly, the rate of convergence calculation could be less pertinent. It would also depends on the quality of the different meshes. Nevertheless, considering Table 1.14, we clearly see that our DG scheme is numerically third-order accurate, the rates of convergence being very close to three.

	L_1		L_2		L_∞	
$\mathcal{N}_c$	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
10×10	3.19E-3	3.30	4.09E-3	3.10	1.52E-2	2.92
20×20	3.35E-4	3.04	4.77E-4	3.01	2.01E-3	2.99
40×40	4.08E-5	3.01	5.93E-5	3.00	2.53E-4	3.00
80×80	5.07E-6	3.00	7.41E-6	3.00	3.16E-5	3.00
160×160	6.34E-7	-	9.26E-7	-	3.95E-6	-

Table 1.13: Rate of convergence for the third-order DG scheme computed in the case of the linear advection ($\boldsymbol{A} = (1,1)^{\mathrm{t}}$) of the smooth initial condition $u^0(\boldsymbol{x}) = \sin(2\pi(x+y))$ where $\boldsymbol{x} = (x,y)^{\mathrm{t}} \in [0,1]^2$, with periodic boundary condition, at the end of a period on Cartesian grids with a CFL= 0.1.

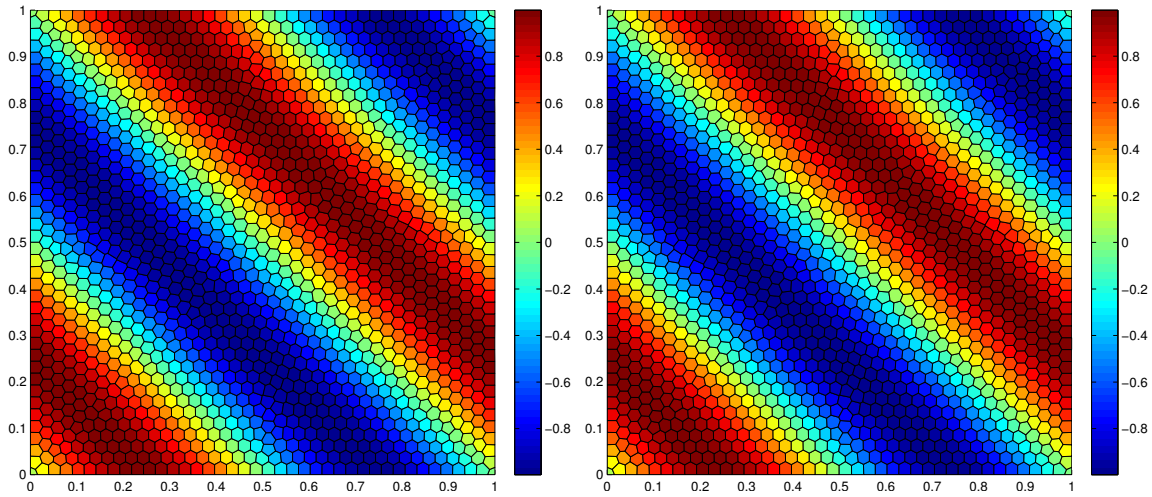


(a) 3rd order scheme. (b) Exact solution.

Figure 1.27: Linear advection ($\boldsymbol{A} = (1,1)^{\mathrm{t}}$) of the smooth initial condition $u^0(\boldsymbol{x}) = \sin(2\pi(x+y))$ where $\boldsymbol{x} = (x,y)^{\mathrm{t}} \in [0,1]^2$, with periodic boundary conditions, at the end of a period on a unstructured grid made of 3860 triangular cells.

	L_1		L_2		L_∞	
$\mathcal{N}_c$	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
230	1.01E-3	3.27	1.28E-3	3.15	4.62E-3	2.61
972	1.05E-4	3.04	1.44E-4	3.08	7.57E-4	2.72
3860	1.28E-5	2.85	1.70E-5	2.85	1.15E-4	3.15
14920	1.77E-6	3.01	2.36E-6	3.01	1.30E-5	2.93
59796	2.19E-7	-	2.93E-7	-	1.71E-6	-

Table 1.14: Rate of convergence for the third-order DG scheme computed in the case of the linear advection ($\boldsymbol{A} = (1,1)^{\mathrm{t}}$) of the smooth initial condition $u^0(\boldsymbol{x}) = \sin(2\pi(x+y))$ where $\boldsymbol{x} = (x,y)^{\mathrm{t}} \in [0,1]^2$, with periodic boundary conditions, at the end of a period on triangular grids with a CFL= 0.1.



(a) 3rd order scheme. (b) Exact solution.

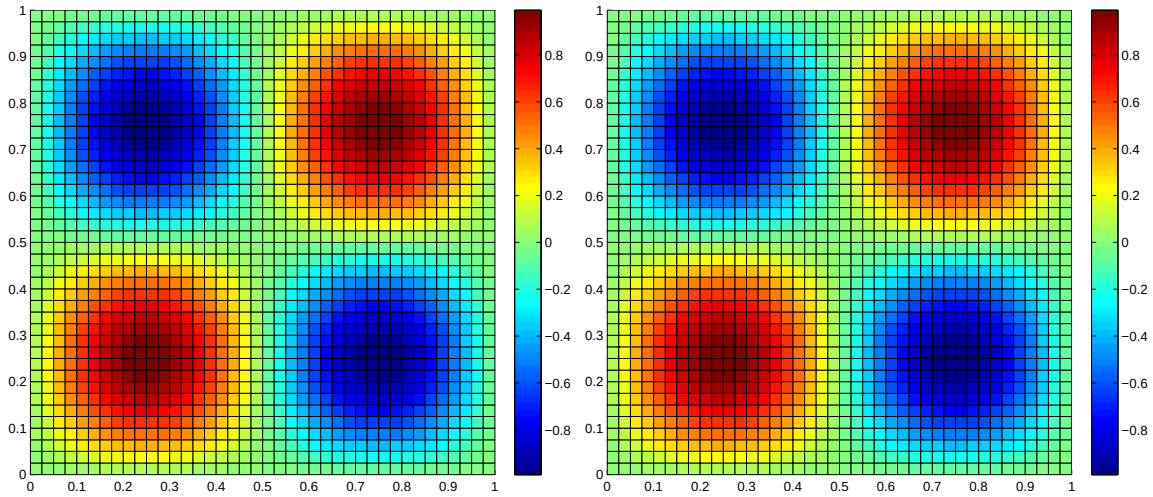
Figure 1.28: Linear advection ($\boldsymbol{A} = (1,1)^{\mathrm{t}}$) of the smooth initial condition $u^0(\boldsymbol{x}) = \sin(2\pi(x+y))$ where $\boldsymbol{x} = (x,y)^{\mathrm{t}} \in [0,1]^2$, with periodic boundary condition, at the end of a period on an unstructured grid made of 1600 polygonal cells.

	L_1		L_2		L_∞	
$\mathcal{N}_c$	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
100	1.76E-3	3.12	2.27E-3	3.10	1.14E-2	2.92
400	2.06E-4	3.06	2.65-4	3.05	1.51E-3	3.08
1600	2.44E-5	2.97	3.19E-5	2.97	1.79E-4	2.96
6400	3.12E-6	3.00	4.07E-6	3.00	2.30E-5	2.93
25600	3.91E-7	-	5.09E-7	-	3.02E-6	-

Table 1.15: Rate of convergence for the third-order DG scheme computed in the case of the linear advection ($\boldsymbol{A} = (1,1)^{\mathrm{t}}$) of the smooth initial condition $u^0(\boldsymbol{x}) = \sin(2\pi(x+y))$ where $\boldsymbol{x} = (x,y)^{\mathrm{t}} \in [0,1]^2$, with periodic boundary conditions, at the end of a period on unstructured polygonal grids with a CFL= 0.1.

Finally, we perform the convergence analysis of our DG scheme using unstructured grids made of general polygonal cells. For polygonal grids, we could make the same remark than previously concerning the impact of the meshes regularity on the computation of the rate of convergence. Table 1.15 confirms the high-order accuracy of our DG scheme. Comparing Table 1.13 and Table 1.15, we notice that for the same number of cells, the third-order DG scheme seems to be more accurate on polygonal cells than on Cartesian cells.This is probably due to the number of faces and thus to the numerical error done on the flux between cells, the polygonal cells having generally more neighbors than the Cartesian cells.

We have confirmed the high accuracy of our DG scheme for both structured and unstructured grids in the case of the linear advection of the smooth initial condition $u^0(\boldsymbol{x}) = \sin(2\pi(x+y))$. This computations have been made without limitation. Now, we will assess the accuracy of our DG scheme using an other linear advection test case characterized by the smooth initial condition $u^0(\boldsymbol{x}) = \sin(2\pi x)\sin(2\pi y)$. The numerical results obtained on a Cartesian grid without limitation are displayed in Figure 1.29.



(a) 3rd order scheme. (b) Exact solution.

Figure 1.29: Linear advection ($\boldsymbol{A} = (1,1)^{\mathrm{t}}$) of the smooth initial condition $u^0(\boldsymbol{x}) = \sin(2\pi x)\sin(2\pi y)$ where $\boldsymbol{x} = (x,y)^{\mathrm{t}} \in [0,1]^2$, with periodic boundary conditions, at the end of a period on a Cartesian grid made of 40×40 cells.

	L_1		L_2		L_∞	
$\mathcal{N}_c$	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
10×10	1.96E-3	3.14	2.55E-3	3.09	8.07E-3	2.90
20×20	2.22E-4	3.01	3.00E-4	3.01	1.08E-3	3.02
40×40	2.75E-5	3.00	3.73E-5	3.00	1.33E-4	3.01
80×80	3.43E-6	3.00	4.67E-6	3.00	1.65E-5	3.01
160×160	4.29E-7	-	5.83E-7	-	2.05E-6	-

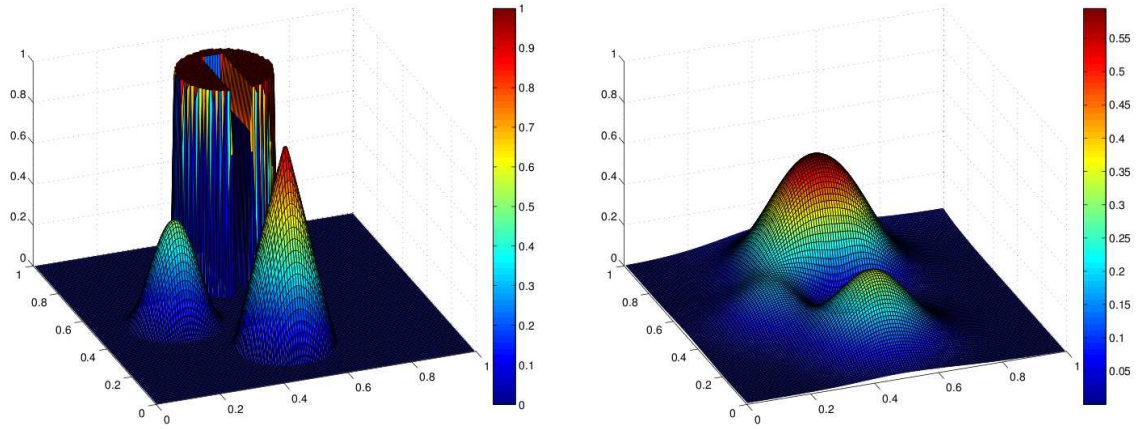
Table 1.16: Rate of convergence for the third-order DG scheme computed in the case of the linear advection ($\boldsymbol{A} = (1,1)^{\mathrm{t}}$) of the smooth initial condition $u^0(\boldsymbol{x}) = \sin(2\pi x)\sin(2\pi y)$ where $\boldsymbol{x} = (x,y)^{\mathrm{t}} \in [0,1]^2$, with periodic boundary conditions, at the end of a period on Cartesian grids with a CFL= 0.1.

The results of the convergence analysis performed in the absence of slope limiter are displayed in Table 1.16. They clearly show that the expected rate of convergence is reached. The impact of the hierarchical vertex-based slope limiter is displayed in Table 1.17. Comparing Table 1.16 (without limitation) and Table 1.17 (with limitation), we clearly observe that the limitation procedure decreases the accuracy without destroying the rate of convergence.

To allow the comparison of our DG discretization to other existing methods, we use the classical test case taken from [66], which corresponds to the two-dimensional extension of the composite signal (1.58) presented in Section 1.1.10, refer to Figure 1.13. We use a velocity field corresponding to a rigid rotation defined previously by $\boldsymbol{A} = (0.5 - y, x - 0.5)^{\mathrm{t}}$, and we apply this solid body rotation to the initial data displayed in Figure 1.30, which includes both a slotted disk, a cone and a smooth hump of the form (1.87). The results displayed in Figure 1.30 show the noteworthy gain in accuracy between the different DG schemes, as well as their robustness. Observing the solutions plotted in Figure 1.30 and the values of the L_2 errors, we can conclude that we obtain similar results to those presented in [60].

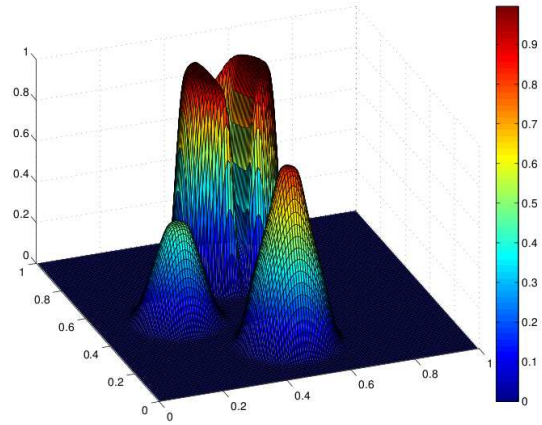
	L_1		L_2		L_∞	
$\mathcal{N}_c$	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
10×10	1.99E-2	3.50	2.49E-2	3.12	8.19E-2	3.00
20×20	1.76E-3	3.84	2.87E-3	3.26	1.03E-2	2.44
40×40	1.23E-4	3.79	2.99E-4	3.31	1.90E-3	2.44
80×80	8.90E-6	3.61	3.00E-5	3.34	3.51E-4	2.48
160×160	7.30E-7	-	2.95E-6	-	6.28E-5	-

Table 1.17: Rate of convergence for the third-order DG scheme with the hierarchical vertex-based slope limiter computed in the case of the linear advection ($\boldsymbol{A} = (1,1)^{\mathrm{t}}$) of the smooth initial condition $u^0(\boldsymbol{x}) = \sin(2\pi(x+y))$ where $\boldsymbol{x} = (x,y)^{\mathrm{t}} \in [0,1]^2$, with periodic boundary conditions, at the end of a period on Cartesian grids with a CFL$= 0.1$.

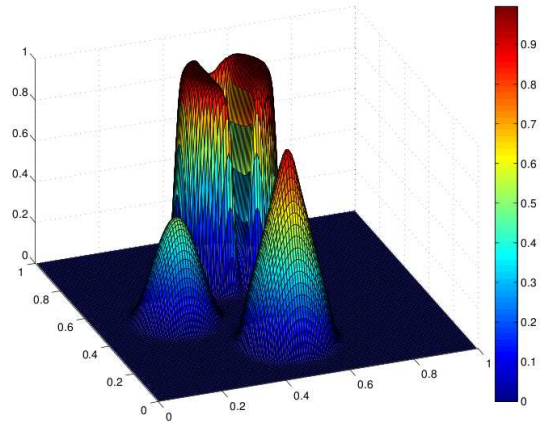


(a) Exact solution.

(b) First-order, $E_{L_2} = 1.80\,10^{-1}$.



(c) Second-order, $E_{L_2} = 7.19\,10^{-2}$.



(d) Third-order, $E_{L_2} = 6.61\,10^{-2}$.

Figure 1.30: Solid body rotation ($\boldsymbol{A} = (0.5-y, x-0.5)^{\mathrm{t}}$) of a composite signal, at the end of a period ($t_f = 2\pi$) on a Cartesian grid made of 128×128 cells. Comparison between the analytical solution and the numerical solutions obtained with the first, second and third-order DG schemes with the vertex-based slope limiter.

Non-linear problems

Burgers problem Here, we consider the inviscid Burgers problem in two space dimensions. In this problem the flux function reads

$$\begin{aligned} \boldsymbol{f}(u) &= (f_1(u), f_2(u))^{\mathrm{t}}, \\ &= \left(\tfrac{1}{2}u^2, \tfrac{1}{2}u^2\right)^{\mathrm{t}}. \end{aligned} \tag{1.88}$$

Let us point out that this problem corresponds to the two-dimensional extension of the one presented in the one-dimensional case, *i.e.*, $f(u) = \frac{1}{2}u^2$. Here, we compute the test case described in [46, 47] for which the initial data is composed of four different constant states

$$u^0(\boldsymbol{x}) = \begin{cases} -0.2 & \text{if } x < 0.5 \text{ and } y > 0.5, \\ -1 & \text{if } x > 0.5 \text{ and } y > 0.5, \\ 0.5 & \text{if } x < 0.5 \text{ and } y < 0.5, \\ 0.8 & \text{if } x > 0.5 \text{ and } y < 0.5. \end{cases} \tag{1.89}$$

We compute the solution on the domain $[0,1]^2$ until $t_f = \frac{1}{2}$, refer to Figure 1.31.
We observe the displacement of three shock waves over the domain and a rarefaction wave in the bottom right corner. Our results are in good agreement with the numerical results displayed in [47].

KPP problem We now turn our attention to nonlinear conservation laws with non-convex flux. To this end we consider the KPP problem which has been proposed by Kurganov, Petrova and Popov [59] to test the convergence properties of some WENO schemes. For this particular non-linear problem, the fluxes are are given by

$$\begin{aligned} \boldsymbol{f}(u) &= (f_1(u), f_2(u))^{\mathrm{t}}, \\ &= (\sin(u), \cos(u))^{\mathrm{t}}. \end{aligned}$$

The initial condition is defined by

$$u^0(\boldsymbol{x}) = \begin{cases} \dfrac{7\,\pi}{2} & \text{if } \sqrt{x^2+y^2} \leq 1, \\ \\ \dfrac{\pi}{4} & \text{if } \sqrt{x^2+y^2} > 1. \end{cases} \tag{1.90}$$

The computational domain, $[-1.55, 1.65] \times [-2.05, 1.15]$, is paved using a 320×320 Cartesian grid made of equally spaced cells, which corresponds to a mesh spacing $\Delta x = \Delta y = \frac{1}{100}$. The computation is run until the stopping time $t_f = 1$. As noticed in [47], this test was proposed since it is challenging to many high-order numerical schemes because the solution has a two-dimensional composite wave structure. The numerical result displayed in Figure 1.32 and obtained , with our third-order DG limited scheme, is quite similar to the results displayed in [59, 47]. It exhibits a quite correct rotating composite wave structure.

Buckley-Leverett problem We consider the classical Buckley-Leverett equation. It corresponds to a simplified model for describing two-phase fluid flow in a porous medium. This example is considered in [47]. As for the KPP problem, the fluxes of this particular non-linear problem are non-convex. There are defined by

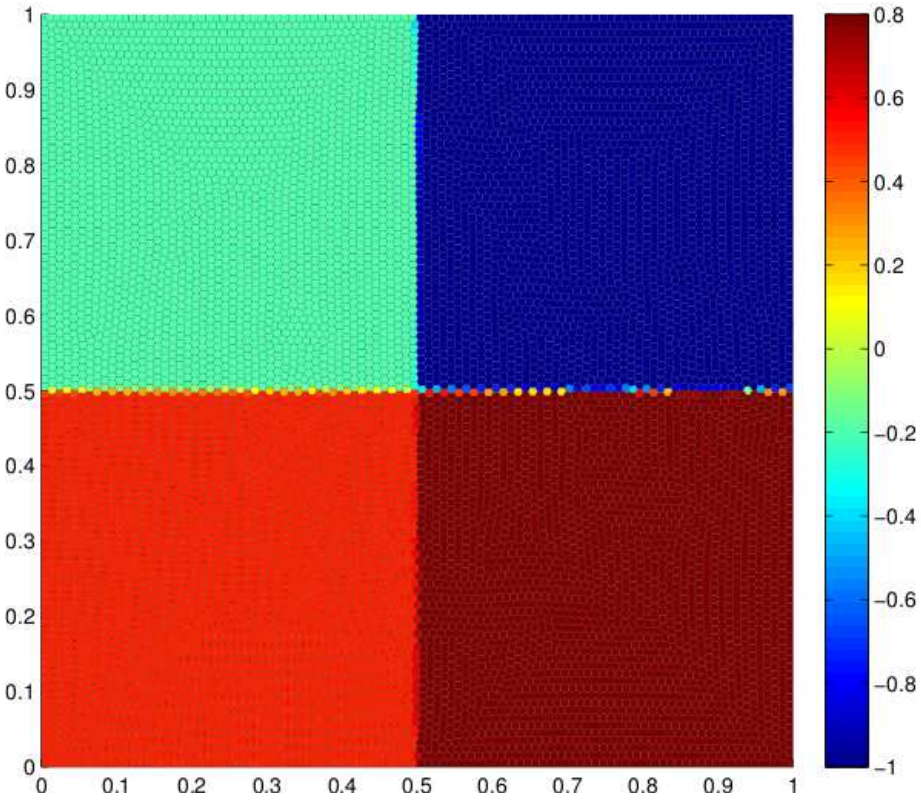
$$\begin{aligned} \boldsymbol{f}(u) &= (f_1(u), f_2(u))^{\mathrm{t}}, \\ &= \left(\frac{u^2}{u^2 + (1-u)^2}, f_1(u)(1 - 5(1-u)^2)\right)^{\mathrm{t}}. \end{aligned} \tag{1.91}$$

The computational domain is $[-1, 1.2]^2$ and the final time is $t_f = 0.5$. The initial condition is defined by

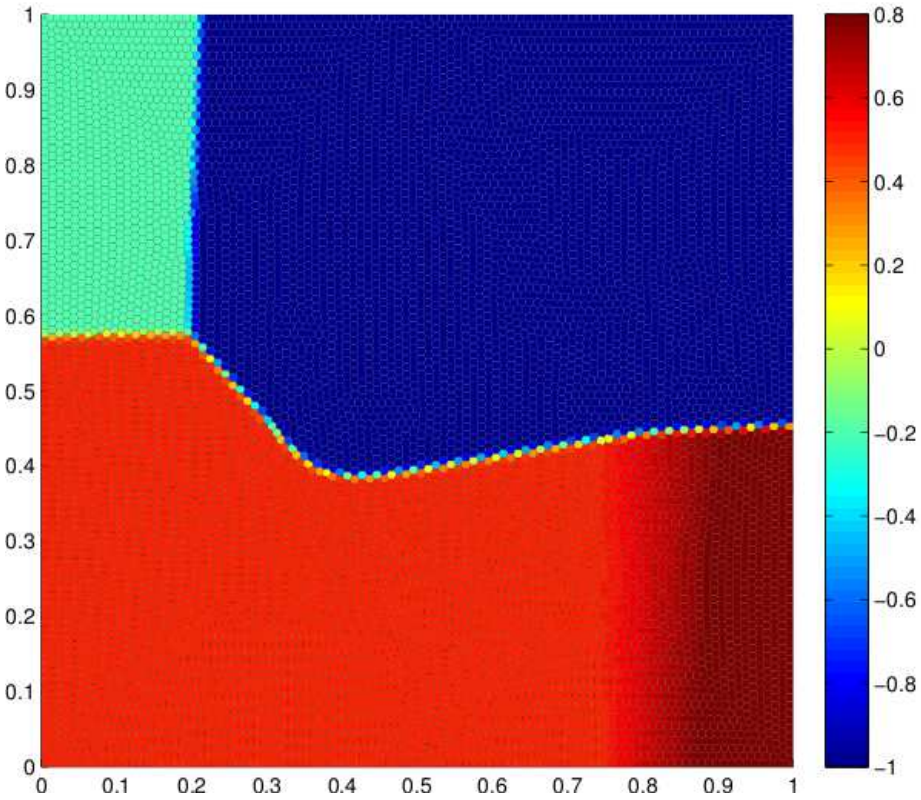
$$u^0(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \sqrt{x^2 + y^2} \leq 0.5, \\ \\ 0 & \text{if } \sqrt{x^2 + y^2} > 0.5. \end{cases} \tag{1.92}$$

The numerical result displayed in Figure 1.33 and obtained using a general unstructured grid made of 10000 polygonal cells, with the third-order DG limited scheme, is quite similar to the result presented in [47].

In this Chapter, we have described the high-order discontinuous Galerkin discretization of the one and two-dimensional scalar conservation laws on general unstructured grids. The main feature of the presented DG scheme lies on the use of a polynomial Taylor basis. This particular choice allows in the two-dimensional case to take into general unstructured grids account in a unified framework. In this frame, a vertex-based hierarchical limitation which preserves smooth extrema has been implemented. A generic form of numerical fluxes ensuring the global stability of our semi-discrete discretization in the L_2 norm has also been designed. We have illustrated the robustness and the accuracy of the implemented schemes using several relevant test cases and performing rate convergences analysis. In the next Chapter, we present the application of this DG discretization to the one-dimensional system of conservation laws.

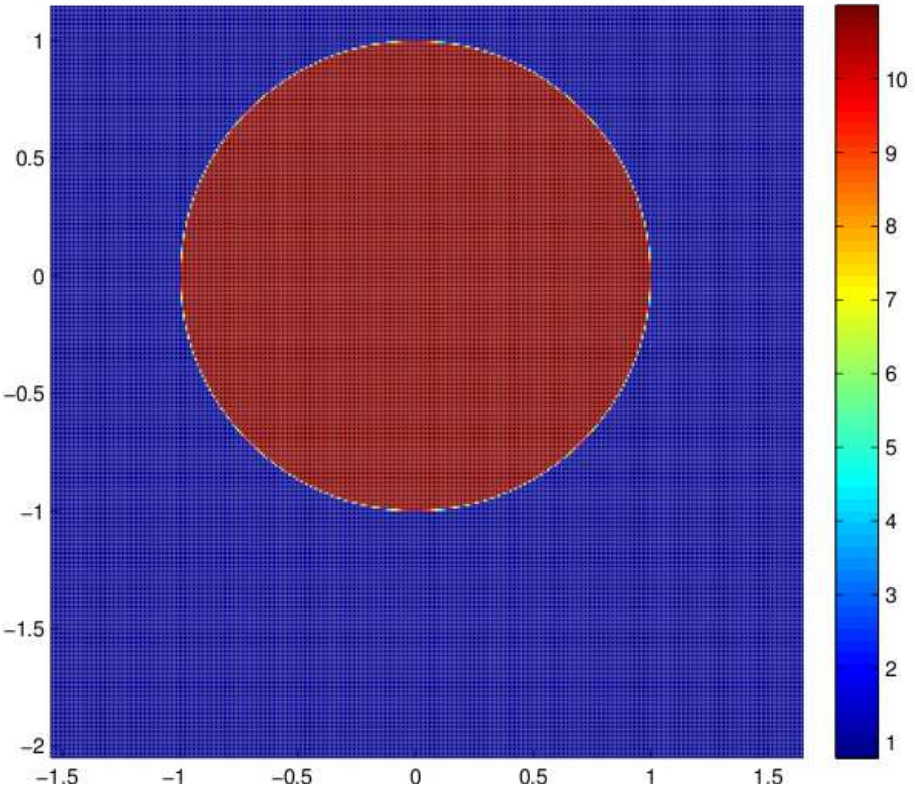


(a) Initial time.

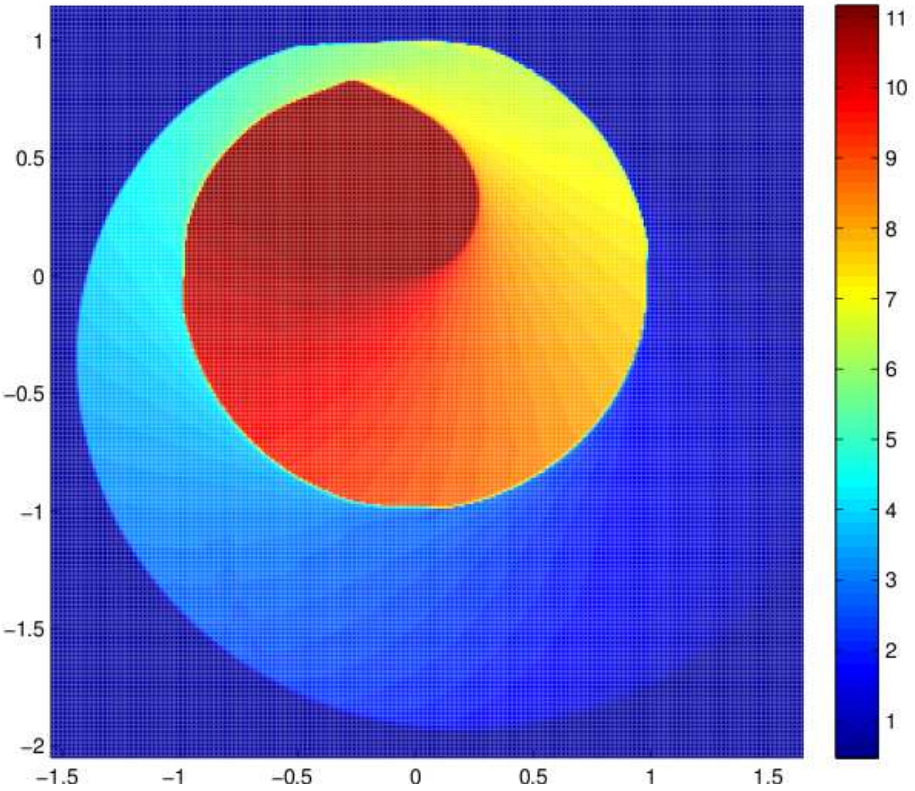


(b) Final time.

Figure 1.31: Solution of the two-dimensional inviscid Burgers problem using the third-order DG scheme with the vertex-based slope limiters on an unstructured grid made of 10000 polygonal cells.

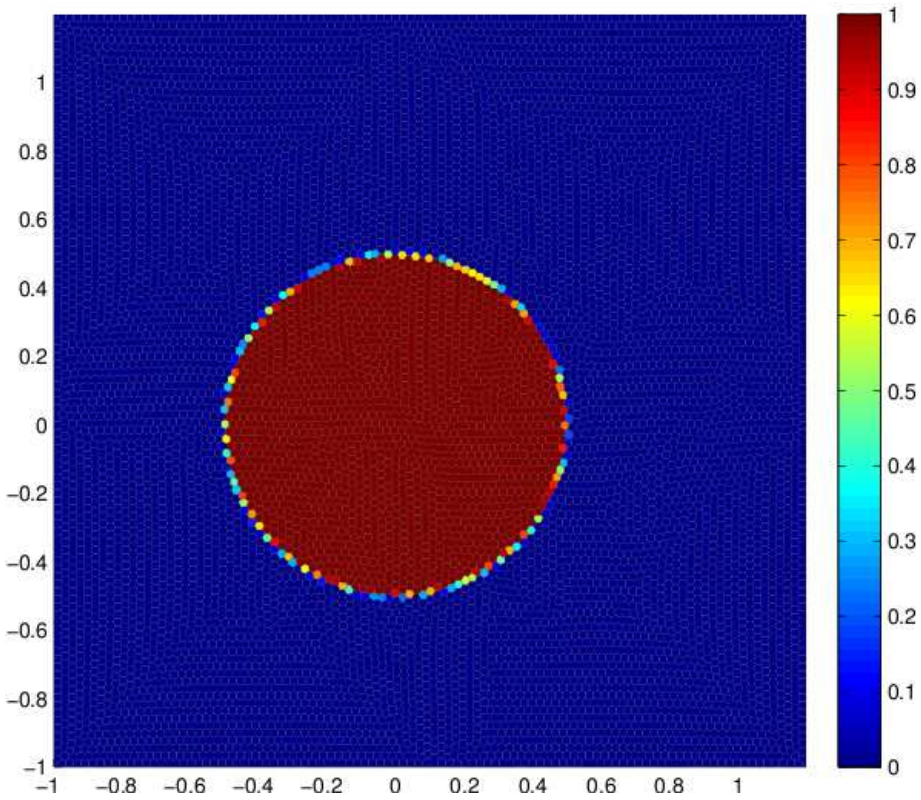


(a) Initial time.

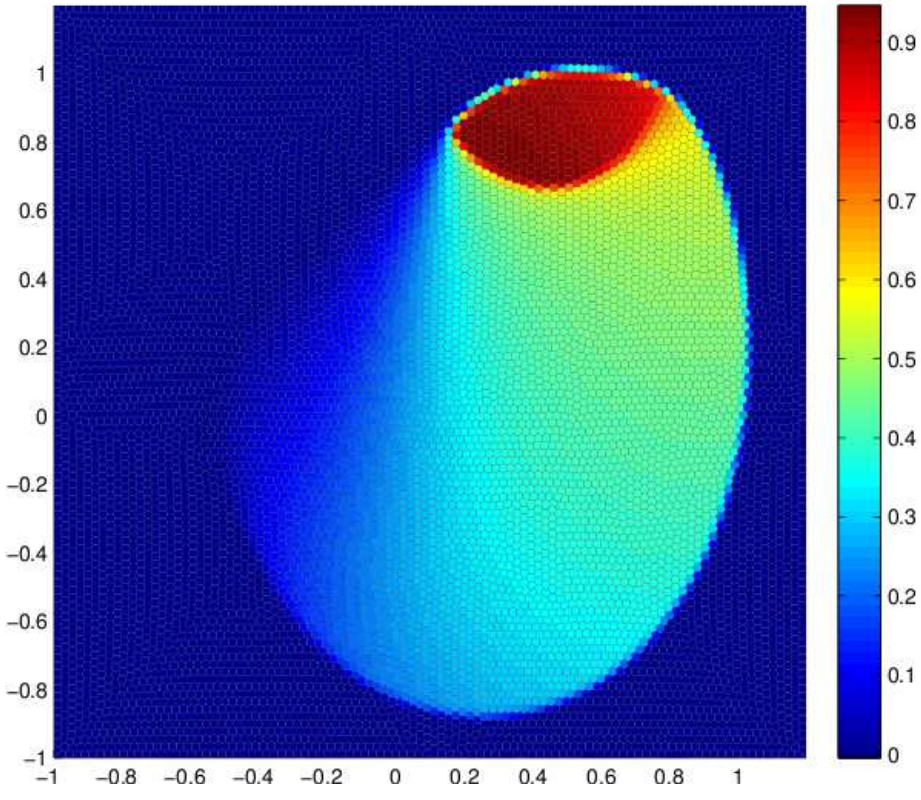


(b) Final time.

Figure 1.32: Solution of the KPP problem using the third-order DG scheme with the hierarchical vertex-based slope limiter on a 320×320 Cartesian grid.



(a) Initial time.



(b) Final time.

Figure 1.33: Solution of the Buckley-Leverett problem using the third-order DG scheme with the hierarchical slope limiter on an unstructured grid made of 10000 polygonal cells.

Chapter 2

High-order DG discretization of one-dimensional systems of conservation laws

This chapter aims at describing the extension of the high-order discontinuous Galerkin discretization presented in Chapter 1 to one-dimensional systems of scalar conservation laws. To this end, we develop a step by step approach, starting from the study of the simple case of linear acoustics to end up with the nonlinear system of gas dynamics equations. In Section 2.1, we introduce the DG discretization of the linear acoustics. We present the construction of the numerical fluxes based on the solution of the Riemann problem. We also describe a limitation procedure which relies on the characteristic variables following the approach originally introduced by Cockburn, Lin and Shu in [26]. In this paper, the authors develop a high-order DG method for one-dimensional conservation laws. Their method applied to linear systems is characterized by a total variation bounded estimate (TVB). Moreover, they clearly show the superiority of a limiting procedure based on the characteristic fields compared to a component-wise-based limiting procedure. Bearing this in mind, we consider in Section 2.2, the extension of our DG scheme to the nonlinear shallow-water system with no source term. After having briefly introduce the concept of Eulerian and Lagrangian description, we write the shallow water equations under the Lagrangian formalism, and discretized it by means of a discontinuous Galerkin procedure. We describe the limitation procedure based on the characteristic variables. Gathering these knowledges, we conclude this chapter by discretizing in Section 2.3 the one-dimensional system of gas dynamics equations. The basic notions related to the kinematics of fluid motion such as the Lagrangian and Eulerian descriptions, are employed to transform the gas dynamics equations and write them under Lagrangian form. The numerical fluxes at the cell interface are defined by means of the classical Godunov acoustic solver [44, 31]. We prove that the semi-discrete scheme with these numerical fluxes satisfies an entropy inequality which ensures a consistent thermodynamic behavior regarding the second law of thermodynamic. Namely, this entropy inequality will ensure that the dissipation of kinetic energy into internal energy throughout irreversible process such as shock waves. Last, the limitation procedure based on the characteristic fields is applied to the Lagrangian gas dynamics system. In the three sections of this chapter, the DG method is implemented up to third-order. Its robustness and its accuracy are assessed by computing the numerical solutions corresponding to various analytical test cases.

2.1 Linear acoustics

In general, one can always obtain a linear system from a nonlinear problem by linearizing it around some given state. It gives a mathematically simpler problem that is useful in some situation, particularly when the interest is in studying small perturbations about some constant state. In this section, we focus on the discretization of the equations of the linear acoustics system. This system results from the linearization of the gas dynamics equations around a background flow. The method to derive the linear acoustics equations is detailed for instance in [67]. Considering the motionless background state characterized by the velocity $U_0 = 0$, the pressure, P, and the velocity, U, perturbations satisfy the linear acoustics system

$$\frac{\partial P}{\partial t} + \rho_0 a_0^2 \frac{\partial U}{\partial x} = 0, \tag{2.1a}$$

$$\frac{\partial U}{\partial t} + \frac{1}{\rho_0} \frac{\partial P}{\partial x} = 0, \tag{2.1b}$$

where ρ_0 and a_0 are the density and the sound speed characterizing the background flow. In what follows, we assume that these quantities are constant non-negative values.

2.1.1 DG discretization of the linear acoustics

The discontinuous Galerkin discretization of the one-dimensional acoustics system is obtained following the same procedure than the one used in the one-dimensional scalar conservation laws case, refer to Section 1.1.6. Let us introduce $C_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ a generic cell of size Δx_i and $\mathbb{P}^K(C_i)$ the set of polynomials of degree up to K. Let P_h^i and U_h^i be the restrictions of P_h and U_h to C_i. These functions are the polynomial approximations of the functions P and U over the cell C_i and they are given by

$$P_h^i(x,t) = \sum_{k=0}^{K} P_k^i(t)\, \sigma_k^i(x), \tag{2.2}$$

$$U_h^i(x,t) = \sum_{k=0}^{K} U_k^i(t)\, \sigma_k^i(x), \tag{2.3}$$

where P_k^i and U_k^i are the successive moments of P_h^i and U_h^i, and $\{\sigma_k^i\}_{k=0,\dots,K}$ is a basis of $\mathbb{P}^K(C_i)$. The coefficients $P_k^i(t)$ and $U_k^i(t)$ present in (2.2) and (2.3) are determined by writing the local variational formulation of the equations (2.1a) and (2.1b). We firstly consider the equation corresponding to the pressure

$$\begin{aligned}\int_{C_i} \frac{\partial P}{\partial t}\, \sigma_q^i \,\mathrm{d}x &= -\int_{C_i} \rho_0 a_0^2 \sigma_q^i \frac{\partial U}{\partial x}\,\mathrm{d}x, \\ &= \rho_0 a_0^2 \left(\int_{C_i} U \frac{\mathrm{d}\,\sigma_q^i}{\mathrm{d}x}\,\mathrm{d}x - \left[\overline{U}\,\sigma_q^i\right]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \right),\end{aligned}$$

where σ_q^c is a function picked into the chosen basis. The velocity field being discontinuous at the cell interfaces, the scheme conservation is ensured by defining a continuous numerical flux, $\overline{U}$, at each cell interface. Interchanging in the previous equations the functions P and U by their polynomial approximations over the cell C_i, respectively P_h^i and U_h^i, provides the $K+1$ semi-discrete equations of the successive moments of the pressure

$$\sum_{k=0}^{K} \left(\int_{C_i} \sigma_k^i\, \sigma_q^i \,\mathrm{d}x \right) \frac{\mathrm{d}\,P_k^i}{\mathrm{d}t} = \rho_0 a_0^2 \left(\sum_{k=0}^{K} U_k^i \int_{C_i} \sigma_k^i \frac{\mathrm{d}\,\sigma_q^i}{\mathrm{d}x}\,\mathrm{d}x - \left[\overline{U}\,\sigma_q^i\right]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \right). \tag{2.4}$$

Recalling the notation introduced for the scalar conservation laws in Section 1.1.6, we set $\boldsymbol{P}^i = \left(P_0^i, \dots, P_l^i, \dots, P_K^i\right)^{\mathrm{t}}$ and $\boldsymbol{U}^i = \left(U_0^i, \dots, U_l^i, \dots, U_K^i\right)^{\mathrm{t}}$ the unknowns vectors. Thus, the $K+1$ semi-discrete equations (2.4) rewrite in the compact form

$$\mathsf{M}^i \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{P}^i = \rho_0 a_0^2 \left(\mathsf{D}^i \boldsymbol{U}^i - \left[\overline{U}\,\boldsymbol{\sigma}^i(x)\right]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \right), \tag{2.5}$$

where the definitions of M^i, D^i and $\boldsymbol{\sigma}^i(x)$ are the same than those given in Section 1.1.6. Applying the same procedure to the velocity equation, one gets

$$\mathsf{M}^i \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}^i = \frac{1}{\rho_0} \left(\mathsf{D}^i \boldsymbol{P}^i - \left[\overline{P}\,\boldsymbol{\sigma}^i(x)\right]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \right), \tag{2.6}$$

where $\overline{P}$ is the numerical flux corresponding to the pressure value at the interfaces between cells. To complete the scheme, it remains to determine the numerical fluxes $\overline{P}$ and $\overline{U}$ in terms of the left and right states on both sides of the cell interface.

2.1.2 Linear Riemann problem and numerical fluxes for linear acoustics

A solution to define the numerical fluxes consists in computing $\overline{P}$ and $\overline{U}$ as the solutions of the Riemann problem characterized by the initial condition

$$P^0(x) = \begin{cases} P_L & \text{for } x < 0 \\ P_R & \text{for } x > 0 \end{cases} \quad \text{and} \quad U^0(x) = \begin{cases} U_L & \text{for } x < 0 \\ U_R & \text{for } x > 0 \end{cases}. \tag{2.7}$$

To solve exactly this Riemann problem, we need to diagonalize the linear acoustics system (2.1) to recover two standard decoupled advection equations. Let $\mathrm{V} = (P, U)^{\mathrm{t}}$ be the vector whose components are the pressure and the velocity perturbations. The system (2.1) can be rewritten as the linear system

$$\frac{\partial \mathrm{V}}{\partial t} + \mathsf{A}\frac{\partial \mathrm{V}}{\partial x} = 0, \tag{2.8}$$

where the matrix A reads

$$A = \begin{pmatrix} 0 & \rho_0 a_0^2 \\ \dfrac{1}{\rho_0} & 0 \end{pmatrix}.$$

This matrix admits two real and distinct eigenvalues: $-a_0$ and a_0. Thus, system (2.8) is hyperbolic and its diagonalization yields

$$\frac{\partial \alpha_+}{\partial t} + a_0 \frac{\partial \alpha_+}{\partial x} = 0, \tag{2.9a}$$

$$\frac{\partial \alpha_-}{\partial t} - a_0 \frac{\partial \alpha_-}{\partial x} = 0, \tag{2.9b}$$

where $\alpha_\pm = P \pm \rho_0 a_0 U$ are the characteristic variables conserved along the characteristic curves $(\mathcal{C}^\pm)$ defined by $y(x,t) = x \pm a_0\, t$. The variables α_+ and α_- are the Riemann invariants. Knowing that the Riemann invariants are constant along the characteristic curves, refer to Figure 2.1, leads to the following relationships

$$\overline{P} + \rho_0 a_0 \overline{U} = P_L + \rho_0 a_0 U_L,$$
$$\overline{P} - \rho_0 a_0 \overline{U} = P_R - \rho_0 a_0 U_R.$$

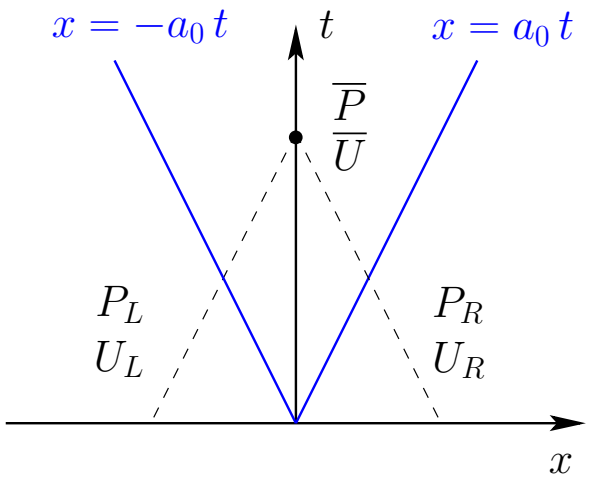


Figure 2.1: Riemann problem in the (x,t)-plan: apparition of two waves of velocity $\pm a_0$.

The solution of the previous system gives the expressions of the numerical fluxes

$$\overline{P} = \frac{P_L + P_R}{2} - \frac{\rho_0 a_0}{2}(U_R - U_L), \tag{2.10}$$

$$\overline{U} = \frac{U_L + U_R}{2} - \frac{1}{2\rho_0 a_0}(P_R - P_L). \tag{2.11}$$

We use this Riemann problem solution to define the numerical fluxes required by our DG scheme setting $P_{L\backslash R}$ and $U_{L\backslash R}$ as the extrapolated values of the functions P_h and U_h on both sides of the interface.

To complete the implementation of our DG scheme we make use of the Taylor basis and of a TVD Runge-Kutta time discretization. The last point which remains to explain is the limitation procedure to enforce the monotonicity of the solutions and avoid the apparition of spurious oscillations, refer to Figure 2.2.

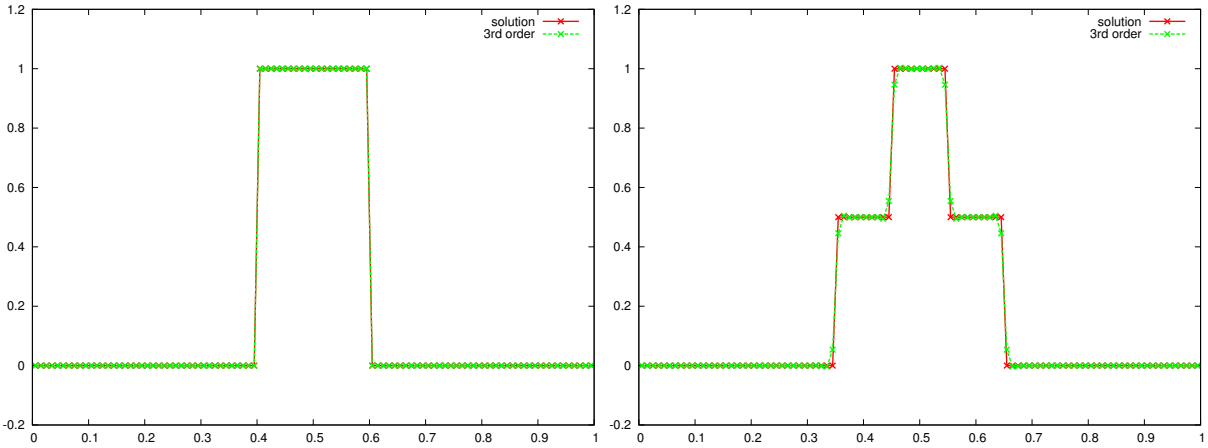
2.1.3 Limiting procedure based on the characteristic variables

In the first chapter, we have presented an high-order limitation procedure preserving smooth extrema. Thus, we can easily apply this procedure to the pressure and velocity polynomial approximations, respectively P_h and U_h. Doing so, the equations not being independent, some oscillations are still visible, refer to Figure 2.3. The results obtained for the scalar conservation laws have motivated our choice of working with the diagonalized form of the linear acoustics system. Hence, the slope limiters are applied on the characteristic variables following the approach given by B. Cockburn and C.-W. Shu [29]. Let us introduce $\alpha^i_{\pm,h}$ the polynomial approximation on the cell C_i of the Riemann invariants $\alpha_\pm$

$$\alpha^i_{\pm,h} = \sum_{k=0}^{K} \alpha^i_{\pm,k}\,\sigma^i_k, \tag{2.12}$$

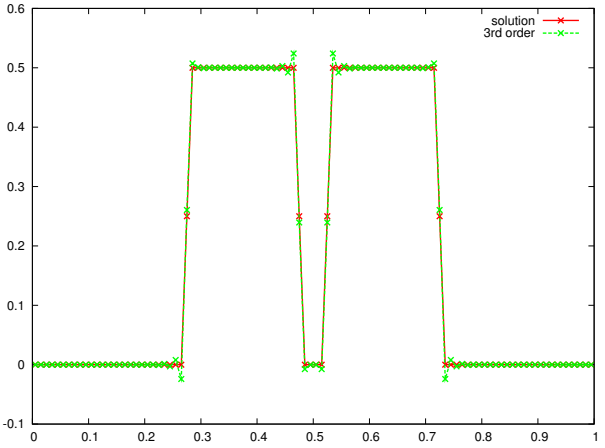
where $\alpha^i_{\pm,k}$ are the successive moments of $\alpha^i_{\pm,h}$. Using the definitions of the Riemann invariants, we get

$$\alpha^i_{\pm,k} = P^i_k \pm \rho_0 a_0\, U^i_k. \tag{2.13}$$

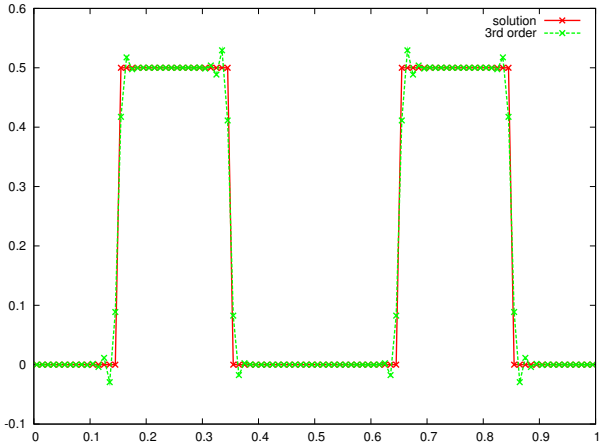


(a) $t = 0$.

(b) $t = 0.05$.



(c) $t = 0.125$.



(d) $t = 0.25$.

Figure 2.2: Solution obtained with the third-order DG scheme for a rectangular initial pressure and with a null initial velocity over 100 cells. Apparition of the two acoustic waves: pressure field.

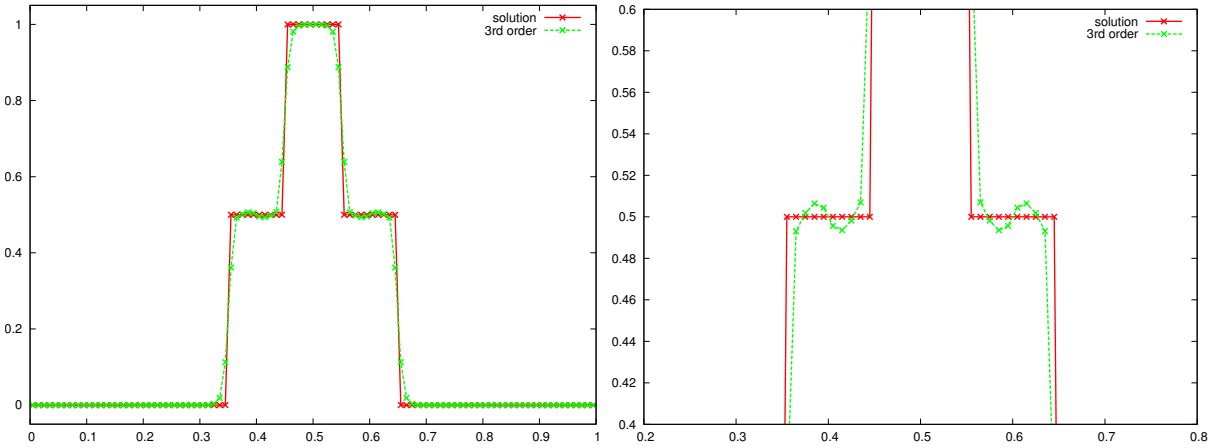
The chosen procedure consists in limiting the polynomial approximations of the Riemann invariants. Then, solving the 2×2 linear system of equations (2.13), we recover the limited coefficients corresponding to P_h^i and U_h^i

$$P_k^i = \frac{\alpha_{+,k}^i + \alpha_{-,k}^i}{2} \quad \text{and} \quad U_k^i = \frac{\alpha_{+,k}^i - \alpha_{-,k}^i}{2\rho_0 a_0}.$$

This limitation procedure avoids the apparition of spurious oscillations and enforces correctly the monotonicity of the pressure and the velocity functions, refer to Figure 2.4.
Now, we briefly presents some results for the one-dimensional linear acoustics to assess the numerical accuracy of the DG scheme presented based on a Runge-Kutta time discretization and the vertex-based slope limiter.

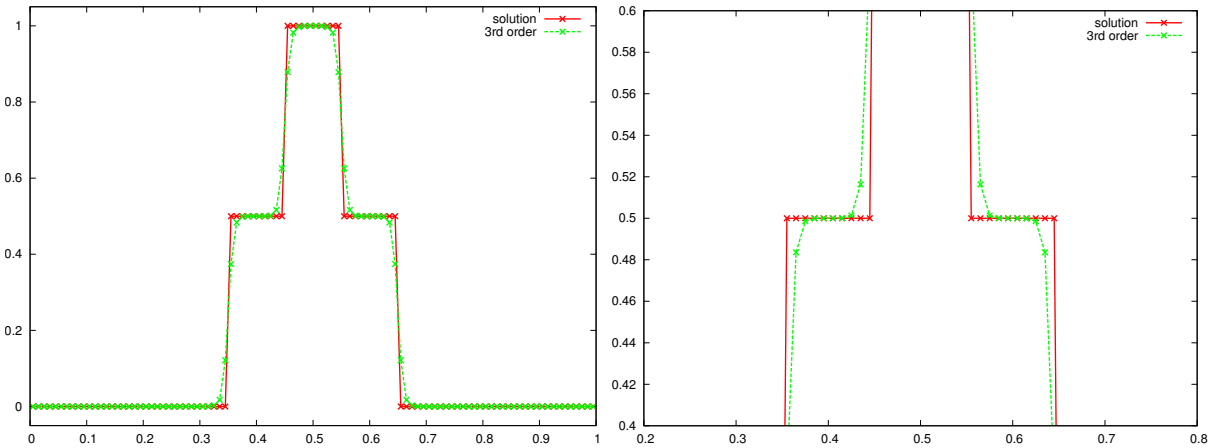
2.1.4 Numerical results for linear acoustics

First, to depict the good behavior of our DG scheme, we compute a similar discontinuous test case than the one used in Section 2.1.3. We initialize the problem on the domain $[-1, 1]$. At



(a) Global view. (b) Zoom on $[0.2, 0.8] \times [0.4, 0.6]$

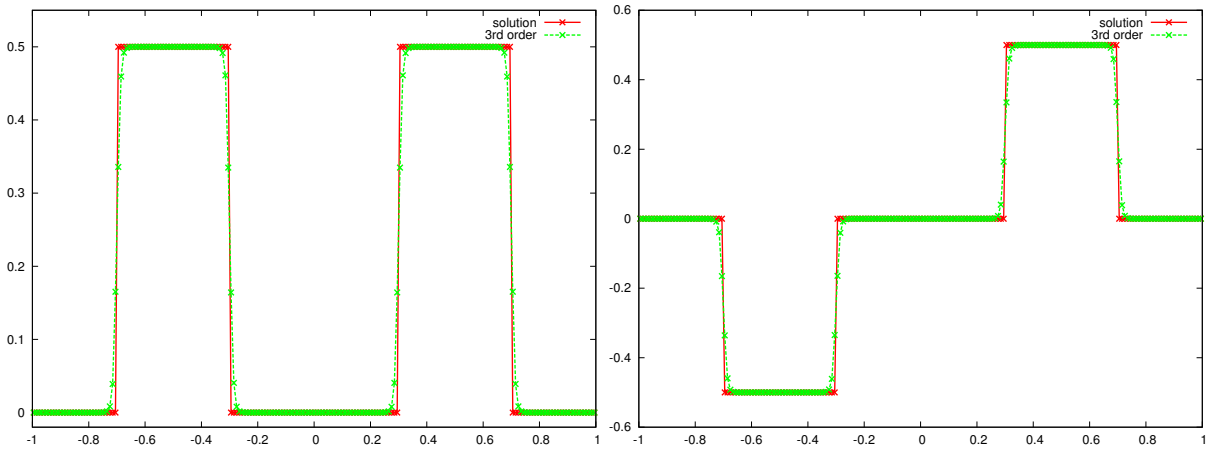
Figure 2.3: Solution obtained with the third-order DG scheme with the limitation procedure on the pressure and the velocity, for a rectangular initial pressure and with a null initial velocity over 100 cells at time $t = 0.05$: pressure field.



(a) Global view. (b) Zoom on $[0.2, 0.8] \times [0.4, 0.6]$

Figure 2.4: Solution obtained with the third-order DG scheme with the limitation procedure on the Riemann invariants, for a rectangular initial pressure and with a null initial velocity over 100 cells at time $t = 0.05$: pressure field.

the initial time, the velocity perturbation is null, and the pressure perturbation consists in a rectangular function between $[-0.2, 0.2]$. The density and sound speed, ρ_0 and a_0, characterizing the background are equal to 1. We display in Figure 2.5 the results obtained by the third-order DG scheme using the vertex-based limiter, at the final time $t = 0.5$ on 200 cells. The limitation of the polynomial approximation based on the characteristic variables suppresses spurious oscillations. Now, to assess the accuracy of the third-order DG scheme, we perform a convergence analysis. The numerical errors are computed for the smooth case wherein the initial pressure and velocity values write $P^0(x) = (\sin(\pi x))^2$ and $U^0(x) = (\sin(\pi x))^2$. The results obtained with the third-order scheme are displayed in Table 2.1. We note that the results totally confirm the high-order of accuracy of the DG scheme applied to the linear acoustics. We also notice considering the results gathered in Table 2.2 that even if the limitation spoils a little



(a) Pressure field. (b) Velocity field.

Figure 2.5: Solution computed with the third-order DG scheme with the limitation on the Riemann invariants, for a rectangular initial pressure and with a null initial velocity over 200 cells.

bit the accuracy of the scheme, it remains third-order accurate.

	L_1		L_2		L_∞	
Δx	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{100}$	3.19E-7	3.00	4.28E-7	3.00	9.47E-7	3.00
$\frac{1}{200}$	3.98E-8	3.00	5.36E-8	3.00	1.18E-7	3.00
$\frac{1}{400}$	4.98E-9	3.00	6.70E-9	3.00	1.48E-8	3.00
$\frac{1}{800}$	6.22E-10	3.00	8.37E-10	3.00	1.85E-9	3.00
$\frac{1}{1600}$	7.79E-11	-	1.05E-10	-	2.32E-10	-

Table 2.1: Rate of convergence on the pressure for the third-order DG scheme without slope limiters computed in the smooth case where $P^0(x) = U^0(x) = (\sin(\pi x))^2$ on the $[0, 1]$ domain, at time $t = 1$ with a CFL$= 0.1$.

	L_1		L_2		L_∞	
Δx	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{100}$	3.43E-6	3.33	9.11E-6	2.95	4.03E-5	2.49
$\frac{1}{200}$	3.41E-7	3.32	1.18E-6	3.23	7.15E-6	2.50
$\frac{1}{400}$	3.42E-8	3.33	1.26E-7	2.70	1.26E-6	2.54
$\frac{1}{800}$	3.40E-9	3.33	1.94E-8	3.00	2.16E-7	2.53
$\frac{1}{1600}$	3.37E-10	-	2.42E-9	-	3.73E-8	-

Table 2.2: Rate of convergence on the pressure for the third-order DG scheme with slope limiters on the Riemann invariants, computed in the smooth case where $P^0(x) = U^0(x) = (\sin(\pi x))^2$ on the $[0, 1]$ domain, at time $t = 1$ with a CFL$= 0.1$.

2.2 Shallow water equations

2.2.1 Governing equations

According to [67] the one-dimensional shallow water equations are derived by considering a fluid in a channel of unit width and assuming that the vertical velocity of the fluid is negligible and the horizontal velocity U is roughly constant throughout any cross section of the channel. Moreover, supposing that the fluid is incompressible, its height, H, is allowed to vary. Writing down the conservation of mass and momentum leads to the following system of equations

$$\frac{\partial H}{\partial t} + \frac{\partial}{\partial x}(HU) = 0, \tag{2.14a}$$

$$\frac{\partial HU}{\partial t} + \frac{\partial}{\partial x}(HU^2 + \frac{1}{2}gH^2) = 0, \tag{2.14b}$$

where $H(x) > 0$ is the water height, $U(x)$ the velocity averaged on the water height and g the strength of the gravitational field which is considered as constant. Introducing the pressure, $P = \frac{1}{2}gH^2$, we observe that the above system is equivalent to the isentropic equations of gas dynamics with the value of polytropic index $\gamma = 2$.
Let us recall that the ultimate goal of this chapter is the implementation of a high-order DG scheme for the one-dimensional Lagrangian gas dynamics equations. The study of the shallow water equations corresponds to a first step in achieving this goal, knowing that the shallow water system is a particular case of the gas dynamics equations. Before describing the DG discretization, we first rewrite the shallow water equations under the Lagrangian formalism. To this end, we shall recall briefly the relationships between the Lagrangian and the Eulerian descriptions of a fluid flow.

2.2.2 Lagrangian form of the shallow water equations

Considering a moving fluid, its flow is described mathematically by the function Φ

$$\Phi : X \longrightarrow x = \Phi(X, t). \tag{2.15}$$

This function, is a one-to-one transformation that maps the initial configuration of the fluid to its actual configuration at time $t > 0$. Here, X is called the Lagrangian coordinate and denotes the initial position of a fluid particle. Its image by the flow map, $x = \Phi(X, t)$, is the position of the same fluid particle at time $t > 0$, it is called the Eulerian coordinate. The flow map, Φ, is characterized by its Jacobian $J = J(X, t)$

$$J = \frac{\partial \Phi}{\partial X} = \frac{\partial x}{\partial X}. \tag{2.16}$$

Knowing that $J(X, 0) = 1$, and assuming by continuity that $J(X, t) > 0$ for all $t > 0$ ensures the existence of the inverse of the flow map for all times. If f denotes a fluid variable, it can be expressed either in terms of the Lagrangian coordinate or in terms of the Eulerian one, that is $f(X, t) = f(x, t)$, knowing that $x = \Phi(X, t)$. We introduce the material derivative of $f = f(X, t)$, which measures the rate of change of f following a fluid particle along its motion as

$$\frac{\mathrm{d} f}{\mathrm{d} t} \equiv \frac{\partial}{\partial t} f(X, t). \tag{2.17}$$

Let us note that it corresponds to the partial time derivative in the Lagrangian description. Now, considering the fluid variable, $f = f(x, t)$, expressed in terms of the Eulerian coordinate, $x = \Phi(X, t)$, and using the chain rule leads to write its material derivative as

$$\frac{\mathrm{d} f}{\mathrm{d} t} = \frac{\partial}{\partial t} f(x, t) + \frac{\mathrm{d} x}{\mathrm{d} t} \frac{\partial}{\partial x} f(x, t). \tag{2.18}$$

Here, the time derivative of the Eulerian coordinate, which corresponds to the velocity of the fluid particle is defined by the trajectory equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\,x(X,t) = U(x,t), \qquad x(X,0) = X, \tag{2.19}$$

where U denotes the velocity of the fluid particle. Bearing this in mind, the material derivative of the fluid variable, f, expressed in terms of the Eulerian coordinate writes

$$\frac{\mathrm{d}\,f}{\mathrm{d}t} = \frac{\partial\, f}{\partial t} + U\,\frac{\partial\, f}{\partial x}. \tag{2.20}$$

Now, utilizing the trajectory equation and the Jacobian definition, one can easily show that the time rate of change of the Jacobian writes

$$\frac{\partial\, J}{\partial t} - \frac{\partial\, U}{\partial X} = 0. \tag{2.21}$$

In terms of the Eulerian coordinate and using the Jacobian definition, the above equation rewrites

$$\frac{\mathrm{d}\,J}{\mathrm{d}t} - J\frac{\partial\, U}{\partial x} = 0. \tag{2.22}$$

Finally, combining (2.22) and (2.20) leads to the following important relationship

$$\frac{\mathrm{d}\,fJ}{\mathrm{d}t} = J\left[\frac{\partial\, f}{\partial t} + \frac{\partial\, fU}{\partial x}\right]. \tag{2.23}$$

This identity allows to express the shallow water equations under the Lagrangian formalism. Applying (2.23) to H and using (2.14a) yields

$$\frac{\mathrm{d}\,HJ}{\mathrm{d}t} = 0.$$

The time integration of this equation, knowing that $J(X,0) = 1$ leads to the mass conservation equation

$$H(X,t)J(X,t) = H^0(X), \tag{2.24}$$

where H^0 denotes the initial water height. Finally, applying (2.23) respectively to the Jacobian equation (2.22) and to the momentum equation (2.14b) leads to the Lagrangian form of the shallow water equations

$$H\frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{H}) - \frac{\partial\, U}{\partial x} = 0, \tag{2.25a}$$

$$H\frac{\mathrm{d}\,U}{\mathrm{d}t} + \frac{\partial\, P}{\partial x} = 0. \tag{2.25b}$$

This Lagrangian form is called Lagrangian updated since the spatial derivative are still expressed in terms of the Eulerian coordinate which is updated by means of the trajectory equation. Finally, substituting (2.24) into the above system and employing the definition of the Jacobian leads to the Lagrangian form of the shallow water equations

$$H^0\frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{H}) - \frac{\partial\, U}{\partial X} = 0, \tag{2.26a}$$

$$H^0\frac{\mathrm{d}\,U}{\mathrm{d}t} + \frac{\partial\, P}{\partial X} = 0. \tag{2.26b}$$

Let us point out that this time the spatial derivative is expressed in terms of the Lagrangian coordinate.

2.2.3 DG discretization of the shallow water equations

The DG discretization of the one-dimensional shallow-water equations is obtained following the same procedure than for the linear acoustic system. Let $(\frac{1}{H})_h^i$ and U_h^i be the restrictions of $(\frac{1}{H})_h$ and U_h, the polynomial approximations of the functions $\frac{1}{H}$ and U, over cell C_i at time t

$$\left(\frac{1}{H}\right)_h^i(X,t) = \sum_{k=0}^{K} \left(\frac{1}{H}\right)_k^i(t)\,\sigma_k^i(X), \tag{2.27}$$

$$U_h^i(X,t) = \sum_{k=0}^{K} U_k^i(t)\,\sigma_k^i(X), \tag{2.28}$$

where $(\frac{1}{H})_k^i$ and U_k^i are the successive moments of $(\frac{1}{H})_h^i$ and U_h^i, and $\{\sigma_k^i\}_{k=0,\dots,K}$ is a basis of $\mathbb{P}^K(C_i)$. As before, these successive moments are determined by writing the local variational formulation of the equations (2.26a) and (2.26b). Here, we make use of similar definitions than the ones used in Section 2.1.1

$$\left(\frac{\mathbf{1}}{\boldsymbol{H}}\right)^i = \left(\left(\frac{1}{H}\right)_0^i,\dots,\left(\frac{1}{H}\right)_l^i,\dots,\left(\frac{1}{H}\right)_K^i\right)^{\mathrm{t}} \quad \text{and} \quad \boldsymbol{U}^i = \left(U_0^i,\dots,U_l^i,\dots,U_K^i\right)^{\mathrm{t}}. \tag{2.29}$$

With these definitions, applying the local variational formulation to the equation (2.26a) leads to the semi-discrete equation

$$\mathsf{M}^i \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathbf{1}}{\boldsymbol{H}}\right)^i = -\mathsf{D}^i\,\boldsymbol{U}^i + \left[\overline{U}\,\boldsymbol{\sigma}^i(X)\right]_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}}, \tag{2.30}$$

where the D^i and $\boldsymbol{\sigma}^i(x)$ correspond to the definitions introduced in Section 1.1.6. However, a slightly difference lies in the definition of M^i. This time, regarding the system of equations (2.26), the mass matrix is weighted by the initial water height H^0

$$\mathsf{M}_{qk}^i = \int_{C_i} H^0(X)\,\sigma_k^i(X)\,\sigma_q^i(X)\,\mathrm{d}X.$$

Thus, we make use of an alternative version of the Taylor basis (1.43). This time, the polynomial basis is constructed by means of the Taylor basis expansion at the center of mass Ξ_i defined as follows

$$\Xi_i = \frac{1}{m_i}\int_{C_i} H^0(X)\,X\,\mathrm{d}X,$$

where $m_i = \int_{C_i} H^0 \mathrm{d}X$. This Taylor basis modification will be detailed in the gas dynamics Section 2.3.2.
Employing exactly the same approach for the momentum equation (2.26b) leads to

$$\mathsf{M}^i \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}^i = \mathsf{D}^i\,\boldsymbol{P}^i - \left[\overline{P}\,\boldsymbol{\sigma}^i(X)\right]_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}}, \tag{2.31}$$

where $\boldsymbol{P}^i = \left(P_0^i,\dots,P_l^i,\dots,P_K^i\right)^{\mathrm{t}}$, denotes the vector of the successive moments of the polynomial approximation of the pressure $P = \frac{1}{2}gH^2$. This approximation is obtained by projecting the function P over the polynomial approximation space defined on the cell C_i.

In the semi-discrete equations (2.30) and (2.31), $\overline{U}$ and $\overline{P}$ are the numerical fluxes corresponding to the velocity and the pressure values at the cell interfaces.

2.2.4 Approximate Riemann solver and numerical fluxes for the shallow water equations

We define the numerical fluxes, $\overline{U}$ and $\overline{P}$, by means of an approximate Riemann solver. To this end, let us recall some classical results regarding the mathematical structure of the shallow water system [67]. First, by definition of the pressure we get

$$\mathrm{d}P = gH\,\mathrm{d}H = -gH^3\,\mathrm{d}\left(\frac{1}{H}\right). \tag{2.32}$$

Assuming the fluid variables are sufficiently smooth and interchanging $\frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{H})$ by $\frac{\mathrm{d}P}{\mathrm{d}t}$ in equation (2.25a) leads to the non-conservative form of the shallow-water system

$$\frac{\mathrm{d}\,P}{\mathrm{d}t} + gH^2\frac{\partial U}{\partial x} = 0, \tag{2.33a}$$

$$\frac{\mathrm{d}\,U}{\mathrm{d}t} + \frac{1}{H}\frac{\partial P}{\partial x} = 0. \tag{2.33b}$$

Let us point out that the above system is very close to the linear acoustics system. Now, introducing the vector $\mathrm{V} = (P, U)^{\mathrm{t}}$, we rewrite the above system under the compact form

$$\frac{\mathrm{d}\,\mathrm{V}}{\mathrm{d}t} + \mathsf{A}\frac{\partial \mathrm{V}}{\partial X} = 0, \tag{2.34}$$

where the matrix A is defined by

$$A = \begin{pmatrix} 0 & gH^2 \\ \dfrac{1}{H} & 0 \end{pmatrix}.$$

This matrix admits the two real eigenvalues $\lambda_\pm = \pm\sqrt{gH}$. Thus, the shallow water system is hyperbolic. Its diagonalization (2.33) leads to the following system

$$\frac{\mathrm{d}\,\alpha_+}{\mathrm{d}t} + \lambda_+\frac{\partial\,\alpha_+}{\partial x} = 0, \tag{2.35a}$$

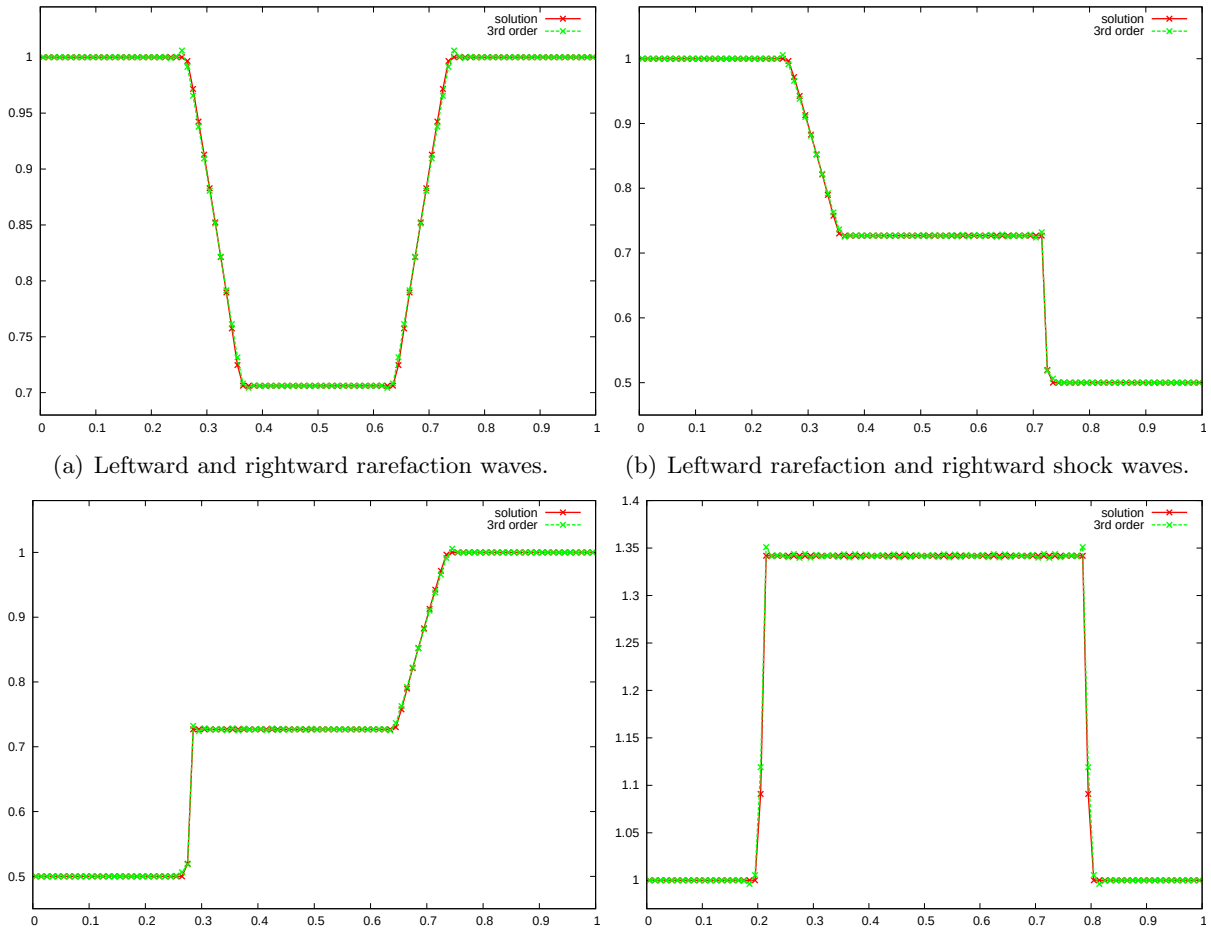
$$\frac{\mathrm{d}\,\alpha_-}{\mathrm{d}t} + \lambda_-\frac{\partial\,\alpha_-}{\partial x} = 0, \tag{2.35b}$$

where $\alpha_\pm$ are the characteristic variables or Riemann invariants conserved along the two characteristic curves ($\mathcal{C}^\pm$) defined by $\frac{\mathrm{d}\,y}{\mathrm{d}t}(x,t) = \lambda_\pm$, $y(x,0) = x$. The Riemann invariants are characterized by the differential relationships

$$\mathrm{d}\alpha_\pm = \mathrm{d}P \pm H\sqrt{gH}\mathrm{d}U. \tag{2.36}$$

The solution of the Riemann problem for the shallow water consists of the combination of two waves: either shock or rarefaction waves, originating from the interface, refer to Figure 2.6.

The direct solution to determine the numerical fluxes consists in studying all the waves combinations and solve exactly the Riemann problem, refer for instance to [67]. Here, we shall adopt a simpler point of view by employing an approximate Riemann solver which is easier to implement. This approximate solver is obtained by linearizing the differential Riemann invariants. Practically, for a Riemann problem defined by the left and right initial states respectively (P_L, U_L) and (P_R, U_R), we linearize $\mathrm{d}\alpha_\pm$ around the initial data, meaning that at the left side



(a) Leftward and rightward rarefaction waves.

(b) Leftward rarefaction and rightward shock waves.

(c) Leftward shock and rightward rarefaction waves.

(d) Leftward and rightward shock waves.

Figure 2.6: Solution obtained with the third-order DG scheme without limitation on 100 cells for the four different Riemann problems: water height.

of the interface $\alpha_{\pm} = P \pm H_L\sqrt{gH_L}\,U$ whereas at the right side $\alpha_{\pm} = P \pm H_R\sqrt{gH_R}\,U$. This leads to a linear system whose solution writes

$$\overline{P} = \frac{Z_R\,P_L + Z_L\,P_R}{Z_L + Z_R} - \frac{Z_L\,Z_R}{Z_L + Z_R}(U_R - U_L), \tag{2.37}$$

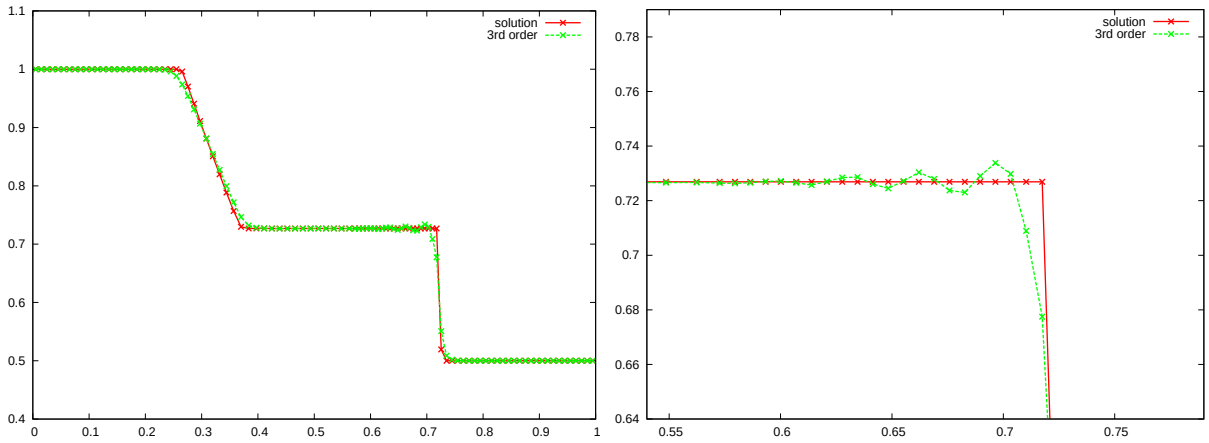
$$\overline{U} = \frac{Z_L\,U_L + Z_R\,U_R}{Z_L + Z_R} - \frac{1}{Z_L + Z_R}(P_R - P_L). \tag{2.38}$$

where $Z = H\sqrt{gH}$. This is the classical Godunov acoustic solver.

This approximate Riemann solver provides the numerical fluxes required by our DG discretization. The linearization procedure of the differential Riemann invariants around the initial condition would correspond to linearize these quantities around the mean values in the cells, evaluated at the previous time, for the calculation of the numerical fluxes. Practically, $P_{L\backslash R}$ and $U_{L\backslash R}$ would be computed as the extrapolated values of the functions P_h and U_h on both sides of the interface, and the acoustic impedances, Z_L and Z_R by the mean values $(\frac{1}{H})_0^{L\backslash R}$ in the cells C_L and C_R surrounding the interface.

2.2.5 Limiting procedure based on the characteristic variables

The results obtained for the system of linear acoustics in Section 2.1.3 show that the slope limiters have to be applied to the polynomial approximation of the Riemann invariants. This important fact is also confirmed in the case of the shallow water equations. Indeed, if we apply the limitation procedure componentwise to the physical variables $\frac{1}{H}$ and U some spurious oscillations occur as it is displayed in Figure 2.7.



(a) Global view.

(b) Zoom on $[0.54, 0.79] \times [0.64, 0.79]$

Figure 2.7: Solution obtained with the third-order DG scheme with the limitation procedure on the water height inverse and the velocity, for a dam break problem (defined in Section 2.2.6) over 100 cells: water height.

Here, we shall develop a vertex-based limiting procedure which applies to the characteristic variables. Considering the Riemann invariants definition (2.36), the first step is to combine these expressions to recover easily the water height, $(\frac{1}{H})_h$, and the velocity U_h. Thanks to the relation (2.32), we redefine the two differential Riemann invariants as follows

$$\mathrm{d}\alpha_{\pm} = \mathrm{d}(\frac{1}{H}) \pm \frac{(\frac{1}{H})^{\frac{3}{2}}}{\sqrt{g}} \mathrm{d}U. \tag{2.39}$$

To find the link with $(\frac{1}{H})_h$ and U_h we linearize $\mathrm{d}\alpha_{\pm}$ on each cell around the mean cell value. To this end, let us introduce $\alpha^i_{\pm,h}$ the polynomial approximation on the cell C_i of the Riemann invariants $\alpha_{\pm}$

$$\alpha^i_{\pm,h} = \sum_{k=0}^{K} \alpha^i_{\pm,k}\, \sigma^i_k, \tag{2.40}$$

where $\alpha^i_{\pm,k}$ are the successive moments of $\alpha^i_{\pm,h}$ which write

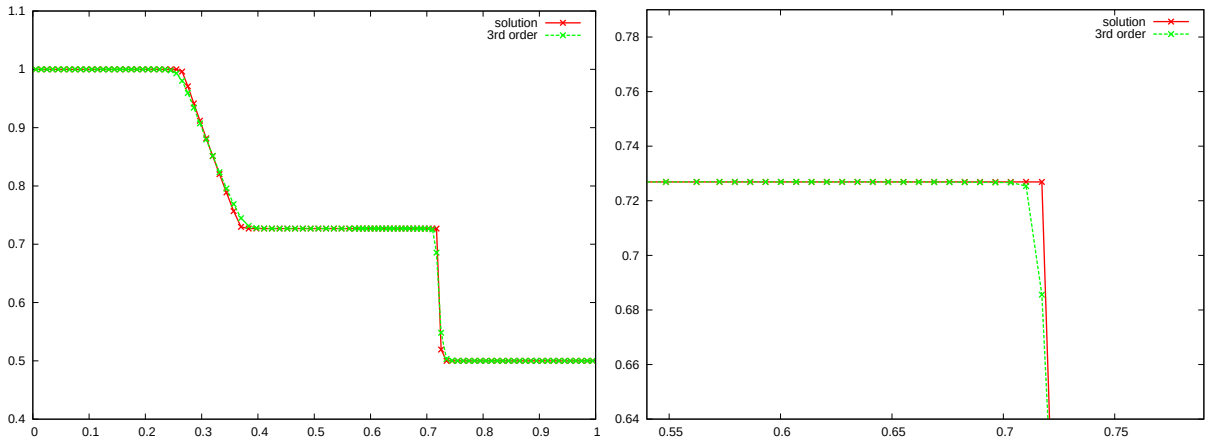
$$\alpha^i_{\pm,k} = (\frac{1}{H})^i_k \pm \frac{\left((\frac{1}{H})^i_0\right)^{\frac{3}{2}}}{\sqrt{g}}\, U^i_k. \tag{2.41}$$

Thus, the chosen procedure consists in limiting the polynomial approximations of the Riemann invariants. Solving the 2×2 linear system of equations (2.41) allows to recover the limited

coefficients corresponding to the polynomial approximations of the original physical variables $(\frac{1}{H})_h^i$ and U_h^i

$$(\frac{1}{H})_k^i = \frac{\alpha_{+,k}^i + \alpha_{-,k}^i}{2},$$
$$U_k^i = \frac{\alpha_{+,k}^i - \alpha_{-,k}^i}{2\left((\frac{1}{H})_0^i\right)^{\frac{3}{2}}}\sqrt{g}.$$

This linearized Riemann invariants limitation avoids the apparition of oscillations and enforces perfectly the monotonicity of the water height and the velocity functions, as we can observe in Figure 2.8.



(a) Global view. (b) Zoom on $[0.54, 0.79] \times [0.64, 0.79]$

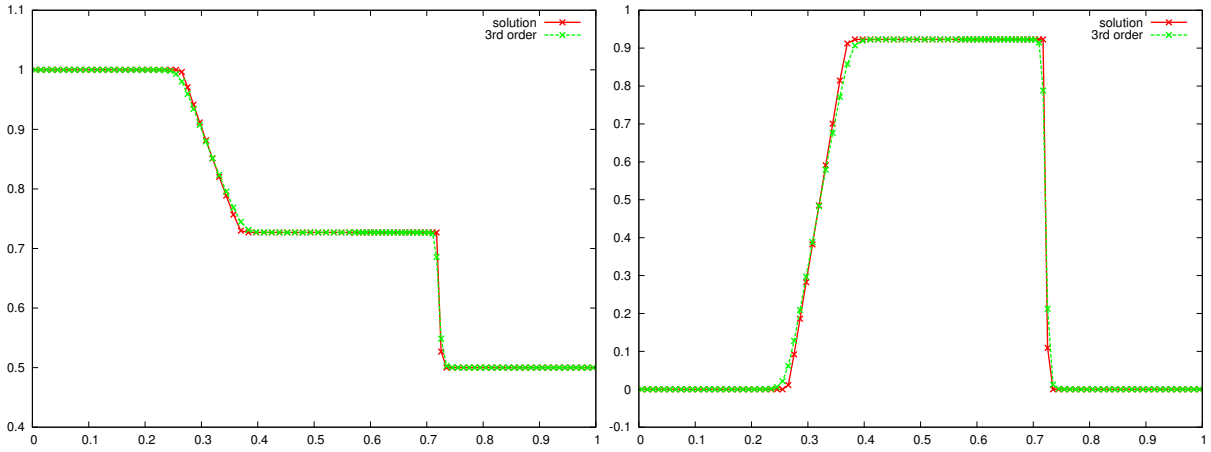
Figure 2.8: Solution obtained with the third-order DG scheme with the limitation procedure on the Riemann invariants, for a dam break problem (defined in Section 2.2.6) over 100 cells: water height .

Now, we briefly present some numerical results to assess the accuracy of our third-order DG discretization.

2.2.6 Numerical results

Dam break problem

We illustrate the high accuracy of the DG scheme presented on the dam break test case [67]. We initialize the problem on the domain $[0, 1]$ with an interface located at $X = 0.5$. At the initial time, the states on the left and the right sides of the interface are constant. The left state is characterized by a high water height $(H_L, U_L) = (1, 0)$, the right by a low water height $(H_R, U_R) = (0.5, 0)$. We display in Figure 2.9 the results obtained with the third-order scheme with the Riemann invariant limitation procedure, on 100 cells at the stopping time $t = 0.075$. We clearly see on this problem the ability of our DG scheme to track accurately the rarefaction wave and the shock wave, refer to Figure 2.9. We also note that the limitation procedure perfectly enforces the monotonicity of the solutions. To quantify the numerical accuracy observed in the numerical results we perform a convergence analysis. For that purpose, we develop a smooth analytical solution.



(a) Water height field. (b) Velocity field.

Figure 2.9: Solution obtained for a dam break problem over 100 cells: comparison between the analytical solution and the numerical solutions obtained with the third-order DG scheme.

Smooth analytical solution for the shallow water equations and convergence analysis

This solution is constructed through the use of the Riemann invariants of the shallow water equations written under the Eulerian form. Integrating the differential expressions (2.36) leads to the expression of the characteristic variables

$$\alpha_{\pm} = U \pm 2\sqrt{gH}. \tag{2.42}$$

These quantities are conserved along the characteristic curves ($\mathcal{C}^{\pm}$) which write in the Eulerian framework

$$(\mathcal{C}^{\pm}) \begin{cases} \dfrac{\mathrm{d}\, y}{\mathrm{d}t}(x,t) = U \pm \sqrt{gH}, \\ y(x,0) = x. \end{cases} \tag{2.43}$$

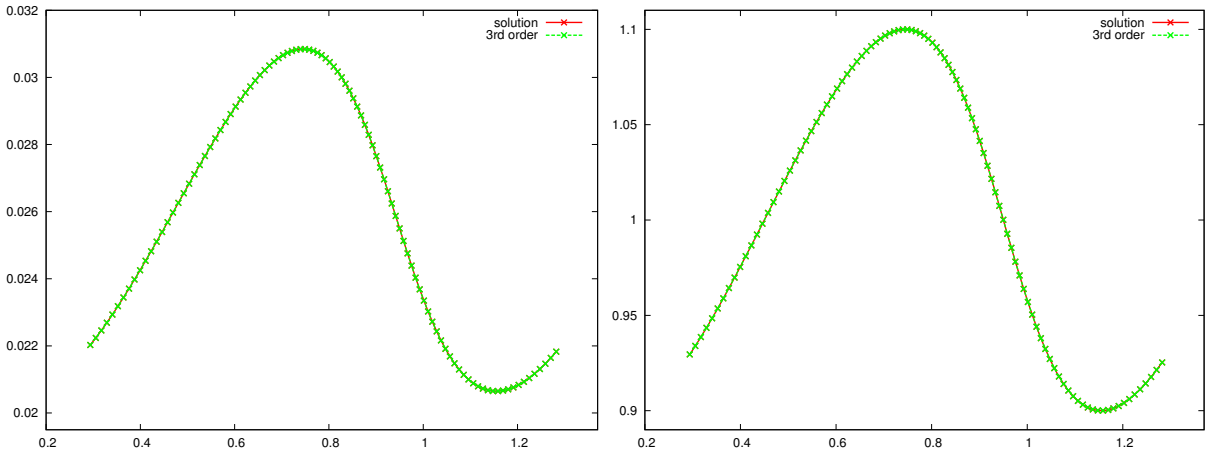
Using the Riemann invariants expression (2.42), the characteristic curves equations rewrite

$$(\mathcal{C}^{\pm}) \begin{cases} \dfrac{\mathrm{d}\, y}{\mathrm{d}t}(x,t) = \dfrac{(2 \pm 1)\alpha_+ + (2 \mp 1)\alpha_-}{4}, \\ y(x,0) = x. \end{cases} \tag{2.44}$$

Here, we have expressed the slopes of the characteristic curves in terms of the Riemann invariants. In the special case where $\alpha_- = 0$, we get $U = 2\sqrt{gH}$ and the shallow water equations boil down to the following non-linear conservation law equation

$$\frac{\partial U}{\partial t} + \frac{3}{2} U \frac{\partial U}{\partial x} = 0.$$

This equation is quite close to the Burgers equation and it is easy to construct an analytical solution. For the numerical applications, we initialize the problem with a constant velocity perturbed by a sinusoidal signal $U^0(X) = 1 + 0.1\sin(2\pi X)$. In this case, the initial water height reads $H^0(X) = \frac{(U^0)^2}{4g}$. We run this test case with the third-order scheme on 100 cells on the domain $[0, 1]$, up to the stopping time $t = 0.3$ with a CFL$= 0.1$. The results are plotted in Figure 2.10. Using this analytical solution we compute the global truncation error corresponding to the third-order DG scheme scheme and display it in Table 2.3.



(a) Water height field.

(b) Velocity field.

Figure 2.10: Solution obtained for the smooth problem over 100 cells: comparison between the analytical solution and the numerical solutions obtained with the third-order DG scheme.

	L_1		L_2		L_∞	
ΔX	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{100}$	3.23E-9	3.00	5.40E-9	3.00	3.02E-8	2.98
$\frac{1}{200}$	4.03E-10	3.00	6.76E-10	3.00	3.83E-9	2.99
$\frac{1}{400}$	5.04E-11	2.99	8.46E-11	3.00	4.82E-10	2.99
$\frac{1}{800}$	6.33E-12	3.00	1.06E-11	3.00	6.05E-11	3.00
$\frac{1}{1600}$	7.90E-13	-	1.33E-12	-	7.58E-12	-

Table 2.3: Rate of convergence on the water height for the third-order DG scheme without slope limiter computed using the smooth initial condition, on the domain $[0, 1]$, at time $t = 0.3$ with a CFL$= 0.1$.

Considering the values of numerical errors, the results gathered in Table 2.3 confirm the high performance of our DG scheme.

In the previous section dealing with the limitation, we have shown that applying the slope limiters to the polynomial approximations of the Riemann invariants perfectly enforces the monotonicity. We still have to verify if this procedure does not decrease the order of accuracy of the scheme. To do so, we perform a convergence rate study with the third-order DG scheme using the limitation based on the characteristic variables, with similar conditions to the ones used before. Considering Table 2.4, we note that the scheme remains third-order accurate.

	L_1		L_2		L_∞	
ΔX	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{100}$	9.32E-9	3.41	2.79E-8	3.22	2.26E-7	2.81
$\frac{1}{200}$	8.78E-10	3.56	2.99E-9	3.66	3.23E-8	3.11
$\frac{1}{400}$	7.47E-11	3.56	2.37E-10	4.48	3.73E-9	2.62
$\frac{1}{800}$	6.33E-12	3.00	1.06E-11	3.00	6.05E-11	3.00
$\frac{1}{1600}$	7.90E-13	-	1.33E-12	-	7.58E-12	-

Table 2.4: Rate of convergence on the water height for the third-order DG scheme with slope limiter based on the characteristic variables, computed in the smooth case on the domain $[0, 1]$, at time $t = 3$ with a CFL= 0.1.

2.3 Lagrangian gas dynamics

The aim of this section is to describe the high-order DG discretization of the gas dynamics equations written under the Lagrangian form. Before we proceed any further, we start by recalling the Eulerian form of the gas dynamics equations

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho U) = 0, \tag{2.45a}$$

$$\frac{\partial}{\partial t}(\rho U) + \frac{\partial}{\partial x}(\rho U^2 + P) = 0, \tag{2.45b}$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x}(\rho U E + P U) = 0. \tag{2.45c}$$

The above equations express respectively the conservation of the mass, the momentum and the total energy of a fluid particle. Here, $\rho > 0$ is the density, U the velocity, P the pressure and E the specific total energy. The thermodynamical closure of this system is obtained through the use of an equation of state, which writes $P = P(\rho, \varepsilon)$, where ε is the specific internal energy, $\varepsilon = E - \frac{1}{2}U^2$. For numerical applications, we use a gamma gas law, *i.e.*, $P = \rho(\gamma - 1)\varepsilon$, where γ is the polytropic index of the gas. Following the developments made in Section 2.2 for the shallow-water equations, we transform system (2.45) to write it under the Lagrangian formalism.

To this end, let us recall that the fluid flow is described mathematically by the continuous transformation $\Phi(X, t) = x$, where t denotes the time, x the Eulerian coordinate, that is the position at time t of a fluid particle initially located at X, the Lagrangian coordinate. The Jacobian of the transformation is denoted by $J = J(X, t) = \frac{\partial x}{\partial X}$. Using the notions introduced in Section 2.2, and considering a fluid variable, $f = f(x, t)$ expressed in terms of the Eulerian coordinate, one can show that the following identity holds

$$\frac{\mathrm{d}\, fJ}{\mathrm{d}t} = J\left[\frac{\partial f}{\partial t} + \frac{\partial fU}{\partial x}\right],$$

where $\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial f}{\partial t} + U\frac{\partial f}{\partial x}$ denotes the material derivative of f, that to say the time derivative along the fluid particle trajectory. Applying the above identity to $f = \rho$ and using the mass conservation equation yields

$$\frac{\mathrm{d}\, \rho J}{\mathrm{d}t} = 0.$$

This is the Lagrangian form of mass conservation. Time integrating this equation leads to the other form of the mass conservation

$$\rho(X, t)J(X, t) = \rho^0(X),$$

where ρ^0 denotes the initial density. Now, applying the above identity to $f = (1, \rho U, \rho E)$ and using the equation of momentum and total energy conservation leads to the updated Lagrangian form of the gas dynamics equations

$$\rho \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho}) - \frac{\partial U}{\partial x} = 0, \tag{2.46a}$$

$$\rho \frac{\mathrm{d}\, U}{\mathrm{d}t} + \frac{\partial P}{\partial x} = 0, \tag{2.46b}$$

$$\rho \frac{\mathrm{d}\, E}{\mathrm{d}t} + \frac{\partial}{\partial x}(PU) = 0. \tag{2.46c}$$

Let us notice that in this Lagrangian system, the differential operators $\frac{\partial}{\partial x}$ are expressed with respect to the Eulerian coordinate. Thus, a discretization of this system would lead to a numerical scheme based on a moving mesh, the motion of the mesh points being governed by the trajectory equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\, x(X,t) = U(x,t), \qquad x(X,0) = X.$$

Finally, recalling that $\rho J = \rho^0$ and employing the Jacobian definition allows to transform the updated Lagrangian form into the total Lagrangian form

$$\rho^0 \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho}) - \frac{\partial U}{\partial X} = 0, \tag{2.47a}$$

$$\rho^0 \frac{\mathrm{d}\, U}{\mathrm{d}t} + \frac{\partial P}{\partial X} = 0, \tag{2.47b}$$

$$\rho^0 \frac{\mathrm{d}\, E}{\mathrm{d}t} + \frac{\partial}{\partial X}(PU) = 0. \tag{2.47c}$$

Here, the spatial derivatives are expressed in terms of the Lagrangian coordinate which does not depend on time. This is why, this Lagrangian form is named total Lagrangian form. We are now in position to construct a high-order Galerkin discretization to solve the above system.

2.3.1 Discontinuous Galerkin discretization of Lagrangian gas dynamics equations

The DG discretization of the Lagrangian gas dynamics equations is quite similar to the one described previously for the shallow water equations. Let us introduce $(\frac{1}{\rho})_h^i$, U_h^i, P_h^i and E_h^i the polynomial approximations in the cell C_i, at time t, of the functions $\frac{1}{\rho}$, U, P and E. To determine the successive moments composing these polynomial approximations, we perform local variational formulations on the equations of system (2.47). Here, to simplify the notation, we identify the functions $\frac{1}{\rho}$, U, P and E to their polynomial approximation over the considered cell.

The volume equation discretization over cell C_i writes

$$\begin{aligned}\int_{C_i} \rho^0 \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})\sigma_q^i \,\mathrm{d}X &= \sum_{k=0}^{K} \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})_k^i \int_{C_i} \rho^0 \sigma_q \sigma_k \,\mathrm{d}X, \\ &= -\int_{C_i} U \frac{\partial\, \sigma}{\partial X}_q^{\,i} \,\mathrm{d}X + \left[\overline{U}\, \sigma_q^i\right]_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}},\end{aligned} \tag{2.48}$$

where σ_q^i is a function picked into the chosen basis $\{\sigma_k^i\}_{k=0,\dots,K}$ of $\mathbb{P}^K(C_i)$. The numerical flux $\overline{U}$ corresponds to the function defining the velocity of the cell interfaces. Here, we identify $\int_{C_i} \rho^0 \sigma_q \sigma_k \,\mathrm{d}X$ as the generic coefficient of the mass matrix which is symmetric positive definite.

For the momentum equation, the procedure is similar. Local variational formulation of equation (2.47b) on C_i leads to

$$\begin{aligned}\int_{C_i} \rho^0 \frac{\mathrm{d}\, U}{\mathrm{d}t} \sigma_q^i \,\mathrm{d}X &= \sum_{k=0}^{K} \frac{\mathrm{d}\, U_k^i}{\mathrm{d}t} \int_{C_i} \rho^0 \sigma_q \sigma_k \,\mathrm{d}X, \\ &= \int_{C_i} P \frac{\partial\, \sigma}{\partial X}_q^i \,\mathrm{d}X - \left[\overline{P}\, \sigma_q^i\right]_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}}, \end{aligned} \tag{2.49}$$

where $\overline{P}$ is the numerical flux corresponding to the pressure function and P_h^i the polynomial approximation of the pressure obtained by projecting it onto the approximation space the cell C_i. Applying once more the local variational formulation to the equation (2.47c) on C_i, one gets

$$\begin{aligned}\int_{C_i} \rho^0 \frac{\mathrm{d}\, E}{\mathrm{d}t} \sigma_q^i \,\mathrm{d}X &= \sum_{k=0}^{K} \frac{\mathrm{d}\, E_k^i}{\mathrm{d}t} \int_{C_i} \rho^0 \sigma_q \sigma_k \,\mathrm{d}X, \\ &= \int_{C_i} PU \frac{\partial\, \sigma}{\partial X}_q^i \,\mathrm{d}X - \left[\overline{PU}\, \sigma_q^i\right]_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}}. \end{aligned} \tag{2.50}$$

where $\overline{PU}$ is the numerical flux corresponding to the pressure work.

In the end, the DG discretization of the gas dynamics equation is given by the $3 \times (K+1)$ semi-discrete equations (2.48), (2.49) and (2.50) where the three numerical fluxes $\overline{U}$, $\overline{P}$ and $\overline{PU}$ remain to be defined. We have also to recall the definition of the Taylor basis which spans the polynomial space approximation. The Taylor basis for the Lagrangian gas dynamics requires some modifications compared to the Taylor basis defined by (1.43) in Section 1.1.8. These modifications essentially comes from the fact that this time the mass matrix is weighted by the initial density.

2.3.2 Taylor basis modification for Lagrangian gas dynamics

Knowing that the mass matrix is weighted by the initial density, we introduce an alternative version of the Taylor basis. In the basis definition (1.43) in Section 1.1.8, we replace the centroid X_i by the center of mass $\mathcal{X}_i$ defined by

$$\mathcal{X}_i = \frac{1}{m_i} \int_{C_i} \rho^0(X)\, X \,\mathrm{d}X, \tag{2.51}$$

where $m_i = \int_{C_i} \rho^0 \mathrm{d}X$ is the constant mass of the cell C_i. Let us point out that in case of a uniform initial density, the center of mass and the centroid coincide. Let $\langle \phi \rangle_i$ be the mass averaged value of ϕ over the C_i

$$\langle \phi \rangle_i = \frac{1}{m_i} \int_{C_i} \rho^0(X)\, \phi(X) \,\mathrm{d}X. \tag{2.52}$$

Knowing that we want our polynomial representation to preserve the mass averaged values leads to define the Taylor basis as follows

$$\sigma_k^i = \frac{1}{k} \left[\left(\frac{X - \mathcal{X}_i}{\Delta X_i} \right)^k - \left\langle \left(\frac{X - \mathcal{X}_i}{\Delta X_i} \right)^k \right\rangle_i \right]. \tag{2.53}$$

This definition of the Taylor basis leads to a new relation on ϕ_0^i, the first component of the polynomial approximation function ϕ_h^i

$$\begin{aligned}\left\langle \phi_h^i \right\rangle_i &= \frac{1}{m_i} \int_{C_i} \rho^0 \, \phi_h^i \, \mathrm{d}X, \\ &= \frac{1}{m_i} \sum_{k=0}^{K} \phi_k^i \int_{C_i} \rho^0 \, \sigma_k^i \, \mathrm{d}X, \\ &= \phi_0^i + \sum_{\boldsymbol{k=1}} \phi_k^i \underbrace{\int_{C_i} \rho^0 \, \sigma_k^i \, \mathrm{d}X}_{=0}, \\ &= \phi_0^i.\end{aligned}$$

Consequently, we identify the first component of ϕ_h^i to the mass averaged value $\langle \phi \rangle_i$. Concerning the definition of the mass matrix, we point out that it is characterized by the following property

$$\mathsf{M}_{0k}^i = \mathsf{M}_{k0}^i = \int_{C_i} \rho^0 \sigma_0^i \, \sigma_k^i \, \mathrm{d}X = \int_{C_i} \rho^0 \sigma_k^i \, \mathrm{d}X = m_i \, \delta_{0k}.$$

Thus, in (2.48), (2.49) and (2.50) the equations corresponding to the first moment are independent of the equations corresponding to the moments of high-order.

We also introduce the scalar product weighted by the initial density

$$\langle u, v \rangle = \int_{C_i} \rho^0(X) \, u(X) \, v(X) \, \mathrm{d}X. \tag{2.54}$$

This new definition of the scalar product has to be included in the projection procedure, applied to the initial conditions and the pressure function, to lead to the polynomial approximations expected.

2.3.3 Entropy inequality and numerical fluxes

The aim of this section is to design numerical fluxes so that our semi-discrete DG scheme satisfies a global entropy inequality. First, we recall the main features of the mathematical structure of the one-dimensional gas dynamics system. Let us assume that the fluid variables are sufficiently smooth to compute their partial derivatives. Dot-multiplying the momentum equation (2.46b) by the velocity and subtracting it from the total energy equation (2.46c) leads to

$$\rho \frac{\mathrm{d}\,\varepsilon}{\mathrm{d}t} + P \frac{\partial U}{\partial x} = 0.$$

Now, substituting the volume equation (2.46b) in the above equation leads to

$$\rho \frac{\mathrm{d}\,\varepsilon}{\mathrm{d}t} + P \, \rho \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho}) = 0.$$

Recalling the definition of the specific entropy, η, thanks to the Gibbs formula

$$\theta \mathrm{d}\eta = \mathrm{d}\varepsilon + P \, \mathrm{d}(\frac{1}{\rho}),$$

where θ denotes the temperature, allows to write the time rate of change of the specific entropy as

$$\frac{\mathrm{d}\,\eta}{\mathrm{d}t} = 0. \tag{2.55}$$

This equation shows that the specific entropy is conserved along the trajectory equation. Thanks to this result and expressing the pressure in terms of the density and the entropy, $P = P(\rho, \eta)$, yields

$$\mathrm{d}P = (\frac{\partial\, P}{\partial \rho})_\eta\, \mathrm{d}\rho + (\frac{\partial\, P}{\partial \eta})_\rho\, \mathrm{d}\eta. \tag{2.56}$$

Introducing the thermodynamic sound speed $a^2 = (\frac{\partial P}{\partial \rho})_\eta > 0$ and knowing that $\frac{d}{dt}\eta = 0$ leads to the following relation between the pressure and the density material derivatives

$$\frac{\mathrm{d}\,P}{\mathrm{d}t} = -\rho^2 a^2\, \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho}). \tag{2.57}$$

Gathering the previous results, we obtain the non-conservative form of the gas dynamics equations

$$\frac{\mathrm{d}\,P}{\mathrm{d}t} + \rho a^2 \frac{\partial\, U}{\partial x} = 0, \tag{2.58a}$$

$$\frac{\mathrm{d}\,U}{\mathrm{d}t} + \frac{1}{\rho}\frac{\partial\, P}{\partial x} = 0, \tag{2.58b}$$

$$\frac{\mathrm{d}\,\eta}{\mathrm{d}t} = 0. \tag{2.58c}$$

Setting $\mathrm{V} = (P, U, \eta)^{\mathrm{t}}$, we can rewrite system (2.58) as

$$\frac{\mathrm{d}\,\mathrm{V}}{\mathrm{d}t} + \mathsf{A}\frac{\partial\, \mathrm{V}}{\partial x} = 0, \tag{2.59}$$

where the matrix A is defined by

$$\mathsf{A} = \begin{pmatrix} 0 & \rho\, a^2 & 0 \\ \frac{1}{\rho} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

This matrix admits three real eigenvalues: $\lambda_1 = 0$, $\lambda_2 = a$, $\lambda_3 = -a$ and thus the above system is hyperbolic [43]. Its diagonalization leads to three scalar conservation equations of the Riemann invariants α_l, where $l = 1, 2, 3$, writing

$$\frac{\mathrm{d}\,\alpha_l}{\mathrm{d}t} + \lambda_l \frac{\partial\, \alpha_l}{\partial x} = 0, \tag{2.60}$$

Let us recall that the λ_l correspond to the slopes of the characteristic curves along which the Riemann invariants are conserved. From an Eulerian point of view, equations (2.60) rewrite

$$\frac{\partial\, \alpha_l}{\partial t} + (U + \lambda_l)\frac{\partial\, \alpha_l}{\partial x} = 0, \tag{2.61}$$

and in this case, the slopes of the characteristic curves are given by U, $U + a$ and $U - a$.

System (2.46) being non-linear, its diagonalization leads to the definition of $\mathrm{d}\alpha_l$, the differential of the Riemann invariants

$$\mathrm{d}\alpha_1 = \mathrm{d}\eta, \tag{2.62}$$

$$\mathrm{d}\alpha_2 = \mathrm{d}P + \rho a \mathrm{d}U, \tag{2.63}$$

$$\mathrm{d}\alpha_3 = \mathrm{d}P - \rho a \mathrm{d}U. \tag{2.64}$$

These quantities will be used in the limitation procedure presented in the next section to enforce the monotonicity of the solutions. They can also be used to construct an approximate Riemann solver similar to the one described in the context of the shallow water equations.

Next, we investigate if it is possible to define the numerical fluxes in such a way that our semi-discrete DG scheme satisfies an entropy inequality. This inequality is very important to ensure that the semi-discrete scheme has a correct behavior from a thermodynamical point of view. Namely, we want our semi-discrete formulation to correctly dissipate kinetic energy into specific energy through irreversible processes such as shock waves. To this end, we construct a semi-discrete equation satisfied by the specific entropy, we multiply respectively (2.47a), (2.47b) and (2.47c) by the test functions P_h^i, U_h^i and 1 and integrate over C_i. As in the previous section, for the sake of conciseness we use the same notations for the functions $\frac{1}{\rho}$, U, P and E than for their polynomial approximation $(\frac{1}{\rho})_h^i$, U_h^i, P_h^i and E_h^i over the considered cell C_i

$$\int_{C_i} \rho^0 P \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})\mathrm{d}X = [P\overline{U}]_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}} - \int_{C_i} U \frac{\partial P}{\partial X}\mathrm{d}X, \tag{2.65a}$$

$$\int_{C_i} \rho^0 U \frac{\mathrm{d}U}{\mathrm{d}t}\mathrm{d}X = -[\overline{P}U]_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}} + \int_{C_i} P \frac{\partial U}{\partial X}\mathrm{d}X, \tag{2.65b}$$

$$\int_{C_i} \rho^0 \frac{\mathrm{d}E}{\mathrm{d}t}\mathrm{d}X = -[\overline{PU}]_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}}. \tag{2.65c}$$

The combination (2.65c)−(2.65b)+(2.65a) leads to

$$\begin{aligned}\int_{C_i} \rho^0 \left[\frac{\mathrm{d}}{\mathrm{d}t}(E - \frac{1}{2}U^2) + P\frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})\right]\mathrm{d}X &= [P\overline{U} + \overline{P}U - \overline{PU}]_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}} - \int_{C_i}\left(P\frac{\partial U}{\partial X} + U\frac{\partial P}{\partial X}\right)\mathrm{d}X, \\ &= [P\overline{U} + \overline{P}U - \overline{PU} - PU]_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}}.\end{aligned}$$

Recalling that specific internal energy writes as $\varepsilon = E - \frac{1}{2}U^2$, and specific entropy is expressed according to the Gibbs formula as $\theta \mathrm{d}\eta = \mathrm{d}\varepsilon + P\,\mathrm{d}(\frac{1}{\rho})$, we deduce

$$\int_C \rho^0 \theta \frac{\mathrm{d}\eta}{\mathrm{d}t}\mathrm{d}X = \sum_{i,cells} R_i,$$

where $R_i = [P\overline{U} + \overline{P}U - \overline{PU} - PU]_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}}$.
At this point, it remains to express the numerical fluxes in such a way that an entropic inequality is satisfied. To this end, we first make the following fundamental assumption

$$\overline{PU} = \overline{P}\,\overline{U}. \tag{2.66}$$

This assumption allows to factorize R_i and to write it as $R_i = [(P - \overline{P})(\overline{U} - U)]_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}}$. Thus, the entropy production related to the semi-discrete scheme writes

$$\int_C \rho^0 \theta \frac{\mathrm{d}\eta}{\mathrm{d}t}\mathrm{d}X = \sum_{i,cells} [(P - \overline{P})(\overline{U} - U)]_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}}. \tag{2.67}$$

We want our DG formulation to satisfy the second law of thermodynamics, that is we want it to convert kinetic energy into internal energy through shock waves. This amounts to design numerical fluxes so that $\sum_i R_i \geq 0$. Interchanging the sum from cells to nodes and setting $\phi_h(X^-_{i+\frac{1}{2}}) = \phi_L$ and $\phi_h(X^+_{i+\frac{1}{2}}) = \phi_R$ yields

$$\sum_{i,cells} R_i = \sum_{i,nodes} \left[(P_L - \overline{P}_{i+\frac{1}{2}})(\overline{U}_{i+\frac{1}{2}} - U_L) - (P_R - \overline{P}_{i+\frac{1}{2}})(\overline{U}_{i+\frac{1}{2}} - U_R) \right].$$

Here, we note that the previous equation has been obtained using periodic boundary conditions. We claim that a sufficient condition to satisfy $\sum_i R_i \geq 0$ consists in setting

$$\overline{P}_{i+\frac{1}{2}} = P_R - Z_R(U_R - \overline{U}_{i+\frac{1}{2}}), \tag{2.68}$$

$$\overline{P}_{i+\frac{1}{2}} = P_L + Z_L(U_L - \overline{U}_{i+\frac{1}{2}}), \tag{2.69}$$

where $Z_{L/R}$ are positive scalars which have the physical dimension of a density times a velocity. The numerical fluxes at node $X_{i+\frac{1}{2}}$ are obtained by solving the previous linear system

$$\overline{P}_{i+\frac{1}{2}} = \frac{Z_L P_R + Z_R P_L}{Z_L + Z_R} - \frac{Z_L Z_R}{Z_L + Z_R}(U_R - U_L), \tag{2.70}$$

$$\overline{U}_{i+\frac{1}{2}} = \frac{Z_L U_L + Z_R U_R}{Z_L + Z_R} - \frac{1}{Z_L + Z_R}(P_R - P_L). \tag{2.71}$$

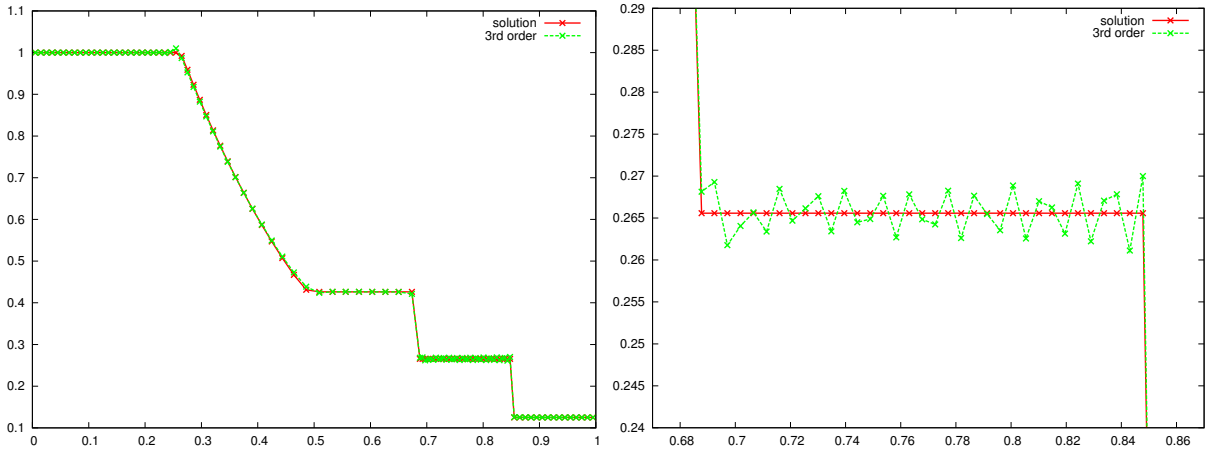
In the non-periodic boundary conditions case, we would have relation (2.68) on the first cell at the first node, and relation (2.69) on the last cell located at the last node. In these two relations we would impose either the velocity or the pressure on the boundary in order to recover both numerical fluxes. In (2.70) and (2.71), by taking $Z = \rho\, a$, we recover the classical acoustic Godunov solver [44, 31] and Z corresponds to the acoustic impedance.

In this section, we have presented a procedure to design numerical fluxes ensuring an entropy inequality. For more details, the interested reader may refer to [31]. To achieve the DG discretization of the Lagrangian gas dynamics equations, we apply the TVD Runge-Kutta time discretization described in Section 1.1.6 to the semi-discrete equations (2.48), (2.49) and (2.50). In Figure 2.11, we have displayed the results obtained for the Sod shock tube problem. We clearly observe the accuracy of the third-order scheme. More precisely, the shock as well as the rarefaction wave are well captured. However, spurious oscillations are visible on the shock plateau. This non monotonic behavior will be cured in the next section by developing a limiting procedure based on the characteristic variables.

2.3.4 Limiting procedure based on the characteristic variables for the Lagrangian gas dynamics equations

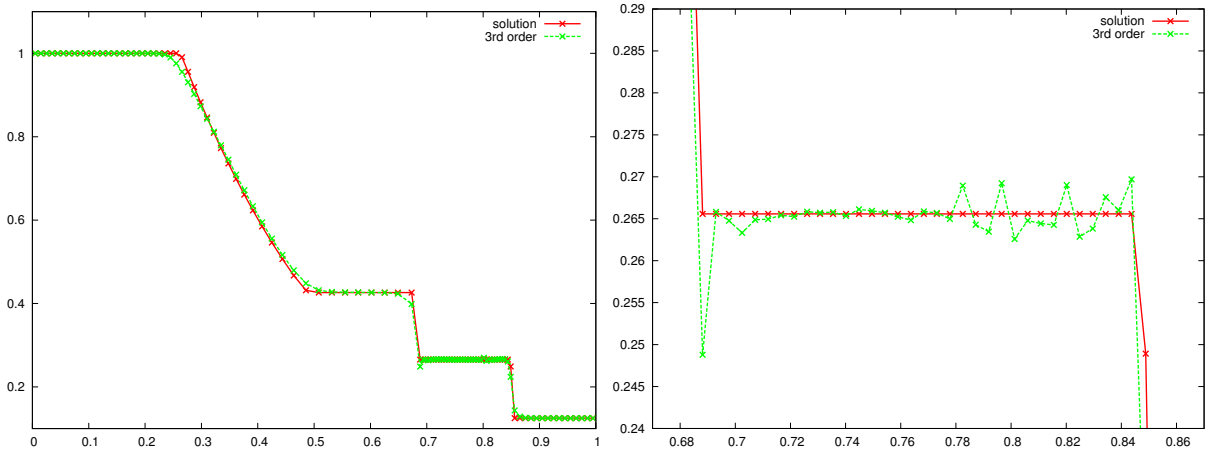
In the case of discontinuous problem, without a specific treatment, high-order numerical schemes produce solutions containing spurious oscillations, refer to Figure 2.11. On the other hand, if we apply a limiting procedure directly to the polynomial approximation of the physical variables $(\frac{1}{\rho})_h$, U_h and E_h, we cannot enforce totally the monotonicity of the solutions, refer to Figure 2.12. To correct this flaw, as we did for the shallow water equations, we construct a limiting procedure based on the Riemann invariants. To this end, we need to derive relationships between the physical variables and the Riemann invariants. First, let us recall that for an isentropic flow the differential of the pressure expresses as

$$\mathrm{d}P = -\rho^2 a^2\, \mathrm{d}(\frac{1}{\rho}).$$



(a) Global view. (b) Zoom on $[0.67, 0.87] \times [0.24, 0.29]$

Figure 2.11: Solution obtained with the third-order DG scheme for the shock Sod tube problem over 100 cells without limitation: density field.



(a) Global view. (b) Zoom on $[0.67, 0.87] \times [0.24, 0.29]$

Figure 2.12: Solution obtained with the third-order DG scheme with the limiting based on the physical variables, for the Sod shock tube problem over 100 cells: density field.

Thus, interchanging $\mathrm{d}P$ and $\mathrm{d}(\frac{1}{\rho})$ in the Riemann invariant differentials (2.63) and (2.64), we are able to redefine $\mathrm{d}\alpha_2$ and $\mathrm{d}\alpha_3$ as follows

$$\mathrm{d}\alpha_2 = \mathrm{d}(\frac{1}{\rho}) + \frac{(\frac{1}{\rho})}{a}\,\mathrm{d}U, \tag{2.72}$$

$$\mathrm{d}\alpha_3 = \mathrm{d}(\frac{1}{\rho}) - \frac{(\frac{1}{\rho})}{a}\,\mathrm{d}U. \tag{2.73}$$

In order to limit the total energy polynomial representation we need to have access to $\mathrm{d}E$ in the Riemann invariant differentials. To do so, using the Gibbs formula, the definition of the remained equation (2.62) yields

$$\mathrm{d}\alpha_1 = \mathrm{d}E - U\mathrm{d}U + P\mathrm{d}(\frac{1}{\rho}). \tag{2.74}$$

We get new expressions of the three Riemann invariants. We linearize these $\mathrm{d}\alpha_l$ quantities on each cell around the mean values of the cell. Let us introduce $\alpha^i_{l,h}$ the polynomial approximation on cell C_i of the Riemann invariants α_l as

$$\alpha^i_{l,h} = \sum_{k=0}^{K} \alpha^i_{l,k}\, \sigma^i_k, \tag{2.75}$$

where $\alpha^i_{l,k}$ are the successive moments of $\alpha^i_{l,h}$ and write

$$\alpha^i_{1,k} = E^i_k - U^i_0\, U^i_k + P^i_0\, (\frac{1}{\rho})^i_k, \tag{2.76}$$

$$\alpha^i_{2,k} = (\frac{1}{\rho})^i_k + \frac{(\frac{1}{\rho})^i_0}{a^i_0}\, U^i_k. \tag{2.77}$$

$$\alpha^i_{3,k} = (\frac{1}{\rho})^i_k - \frac{(\frac{1}{\rho})^i_0}{a^i_0}\, U^i_k. \tag{2.78}$$

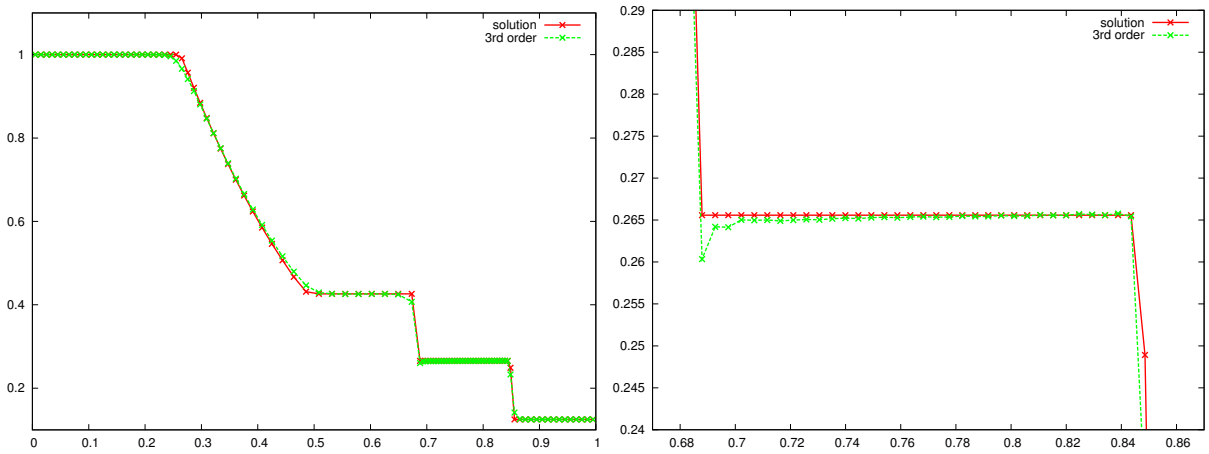
Hence, the chosen procedure is again to limit the polynomial approximations of the Riemann invariants and solving the 3×3 linear system of equations (2.76), (2.77) and (2.78) we recover the limited coefficients corresponding to the physical variables polynomial approximations $(\frac{1}{\rho})^i_h$, U^i_h and E^i_h

$$\begin{aligned}
(\frac{1}{\rho})^i_k &= \frac{\alpha^i_{2,k} + \alpha^i_{3,k}}{2}, \\
U^i_k &= \frac{\alpha^i_{2,k} - \alpha^i_{3,k}}{2(\frac{1}{\rho})^i_0}\, a^i_0. \\
E^i_k &= \alpha^i_{1,k} + \frac{1}{2}\alpha^i_{2,k}\left(P^i_0 + U^i_0 a^i_0\right) + \frac{1}{2}\alpha^i_{3,k}\left(P^i_0 - U^i_0 a^i_0\right).
\end{aligned}$$

As we can see in the results displayed in Figure 2.13, the limiting procedure based on the characteristic variables avoids the apparition of spurious oscillations and enforce the monotonicity of the solutions. The undershoot visible in Figure 2.13 is independent of the limitation. This phenomenon is inherent to the Lagrangian formalism, and is due to the strong difference in the cell aspect ratio before and after the contact discontinuity. Next, we present some numerical results to assess the accuracy of our high-order DG discretization.

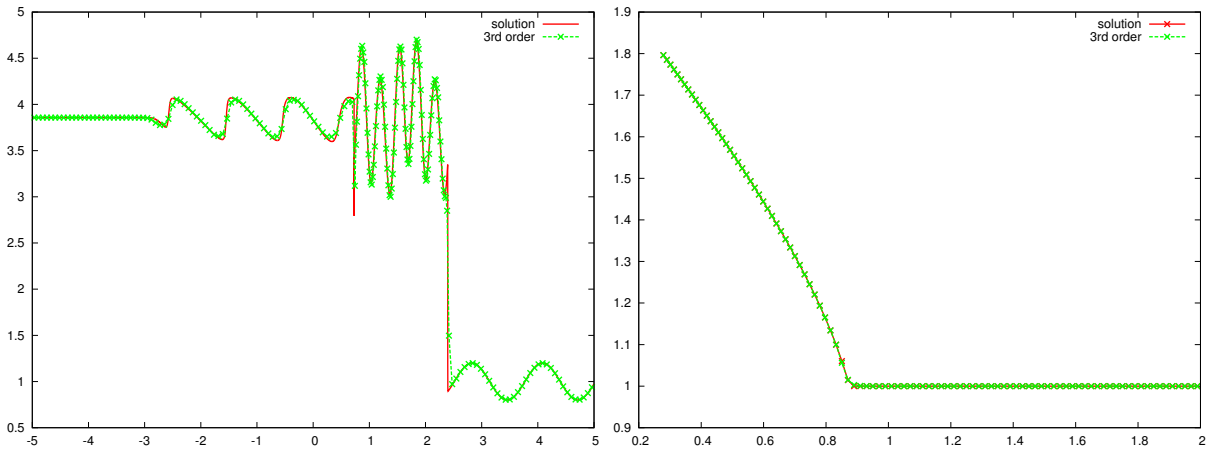
2.3.5 Numerical results

To demonstrate the accuracy and the robustness of our scheme solving we have run test cases taken from the literature. These results, displayed in Figure 2.14, have been obtained with our third-order scheme with slope limiters. In Figure 2.14(a), our scheme performs on the Shu oscillating shock tube problem, presented by Chi-Wang Shu and Stanley Osher in [99]. Despite the strong perturbations, we note that the numerical solution is very close to the reference solution. In Figure 2.14(b), the run test case is the uniformly accelerated piston problem, defined by Pierre-Henri Maire in [74]. For this smooth solution, we notice once more, how accurate the numerical solution is. Next, we perform a convergence analysis of our DG scheme using a smooth solution of the gas dynamics equations.



(a) Global view. (b) Zoom on $[0.67, 0.87] \times [0.24, 0.29]$

Figure 2.13: Solution obtained with the third-order DG scheme with the limiting procedure based on the linearized Riemann invariants, for the Sod shock tube problem over 100 cells: density field.



(a) Shu oscillation shock tube with 200 cells. (b) Uniformly accelerated piston with 100 cells.

Figure 2.14: Numerical solutions for gas dynamics with limitation for the third-order DG method: density field.

Smooth solution and rates of convergence

Here, we construct a smooth solution of the gas dynamics equations by means of the Riemann invariants. Among all the possible formulas expressing the Riemann invariant differentials presented in the previous sections, we introduce $\mathrm{d}\alpha_\pm$ corresponding to the former $\mathrm{d}\alpha_2$ and $\mathrm{d}\alpha_3$ and defined by

$$\mathrm{d}\alpha_\pm = \mathrm{d}U \pm \frac{a}{\rho}\mathrm{d}\rho. \tag{2.79}$$

In case of an isentropic fluid flow and for a gamma gas law, the pressure writes

$$\frac{P}{P^0} = \left(\frac{\rho}{\rho^0}\right)^\gamma. \tag{2.80}$$

Recalling that the sound speed writes $a^2 = \frac{\gamma P}{\rho}$ and using (2.80) yields

$$\frac{a}{a^0} = \left(\frac{\rho}{\rho^0}\right)^{\frac{\gamma-1}{2}}. \tag{2.81}$$

We are now able to integrate analytically the Riemann invariants of equation (2.79)

$$\alpha_\pm = U \pm \frac{2}{\gamma-1}\, a. \tag{2.82}$$

The Riemann invariants are constant along the characteristic curves, these curves $(\mathcal{C}^\pm)$ being defined by the differential equations

$$(\mathcal{C}^\pm) \begin{cases} \dfrac{\mathrm{d}\, y}{\mathrm{d}t}(x,t) = U \pm a, \\ y(x,0) = x. \end{cases} \tag{2.83}$$

Using the Riemann invariants definitions (2.82), the characteristic curves equations rewrite

$$(\mathcal{C}^\pm) \begin{cases} \dfrac{\mathrm{d}\, y}{\mathrm{d}t}(x,t) = \dfrac{(2 \pm (\gamma-1))\alpha_+ + (2 \mp (\gamma-1))\alpha_-}{4}, \\ y(x,0) = x. \end{cases} \tag{2.84}$$

Here, we have expressed the slopes of the characteristic curves in terms of the Riemann invariants. In the general case, these curves are not straight lines because even if $\alpha_\pm$ is constant along $(\mathcal{C}^\pm)$, it is not the case for $\alpha_\mp$. Nevertheless, in the particular case where $\gamma = 3$ the characteristic curves are straight lines defined by

$$(\mathcal{C}^\pm) \begin{cases} \dfrac{\mathrm{d}\, y}{\mathrm{d}t}(x,t) = \alpha_\pm, \\ y(x,0) = x. \end{cases} \tag{2.85}$$

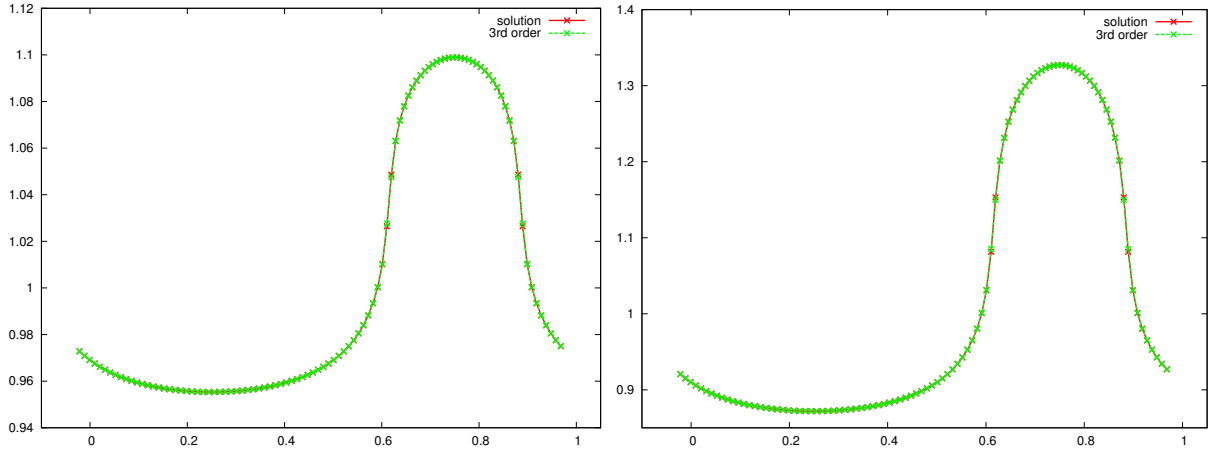
Hence, the gas dynamics equations are equivalent to the two following Burgers equations

$$\frac{\partial\, \alpha_\pm}{\partial t} + \alpha_\pm \frac{\partial\, \alpha_\pm}{\partial x} = 0,$$

for which an analytical solution is easy to construct.

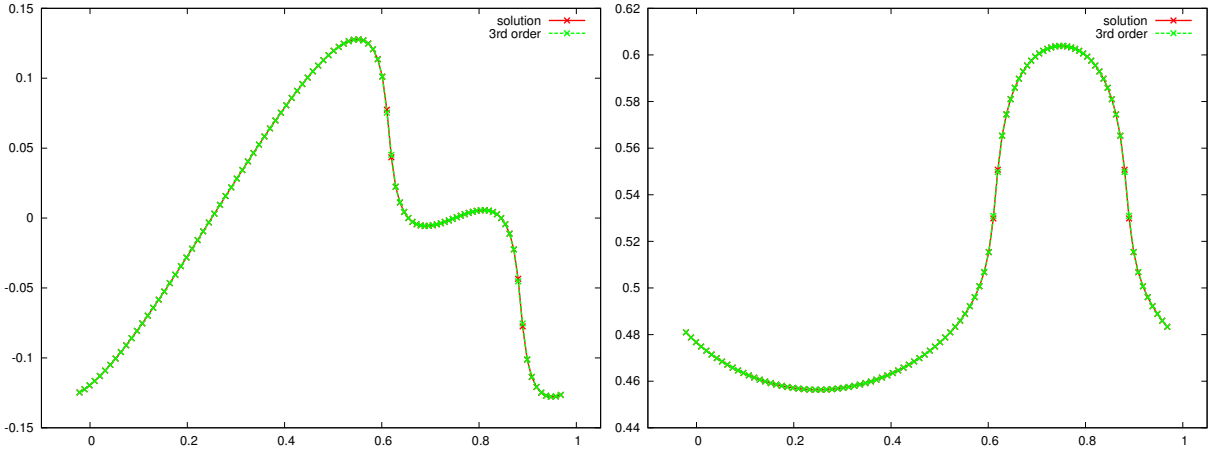
For the numerical simulations, we initialize the problem with a constant density perturbed by a sinusoidal signal $\rho^0(X) = 1 + 0.1\sin(2\pi X)$, an initial pressure given by the isentropic relation (2.80) $P^0(X) = \left(\rho^0\right)^\gamma$ and a null initial velocity $U^0(X) = 0$. We run our third-order scheme on this test case with 100 cells on the $[0,1]$ domain, at time $t = 0.8$ with a CFL$= 0.1$. The results are displayed in Figure 2.15. Using this analytical solution we compute the global truncation error corresponding to the third-order DG scheme and display it in Table 2.5. These results confirm the high accuracy of this scheme.

In the previous section dealing with the limitation, we have shown that applying the slope limiters to the polynomial approximations of the Riemann invariants enforces perfectly the monotonicity. We still have to verify if this procedure does not decrease the order of accuracy of the scheme. To do so, we perform a convergence analysis with the third-order limited DG scheme using once more the previous smooth test case. Observing Table 2.6, we note that even if the slope limiter spoils the accuracy of the scheme, it remains third-order accurate, the rate of convergence being close to three.



(a) Density field.

(b) Pressure field.



(c) Velocity field.

(d) Total energy field.

Figure 2.15: Solution obtained for the smooth test case over 100 cells: comparison between the analytical solution and the solution obtained with the third-order DG scheme

	L_1		L_2		L_∞	
ΔX	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{50}$	9.09E-5	3.01	3.40E-4	2.87	2.20E-3	2.79
$\frac{1}{100}$	1.13E-5	3.58	4.64E-5	3.28	3.17E-4	2.70
$\frac{1}{200}$	9.40E-7	3.30	4.79E-6	3.34	4.89E-5	2.64
$\frac{1}{400}$	9.57E-8	3.03	4.74E-7	3.07	7.85E-6	2.91
$\frac{1}{800}$	1.17E-8	-	5.63E-8	-	1.04E-6	-

Table 2.5: Rate of convergence on the velocity for the third order DG scheme without slope limiters computed in the smooth case on the $[0, 1]$ domain, at time $t = 0.8$ with a CFL$= 0.1$.

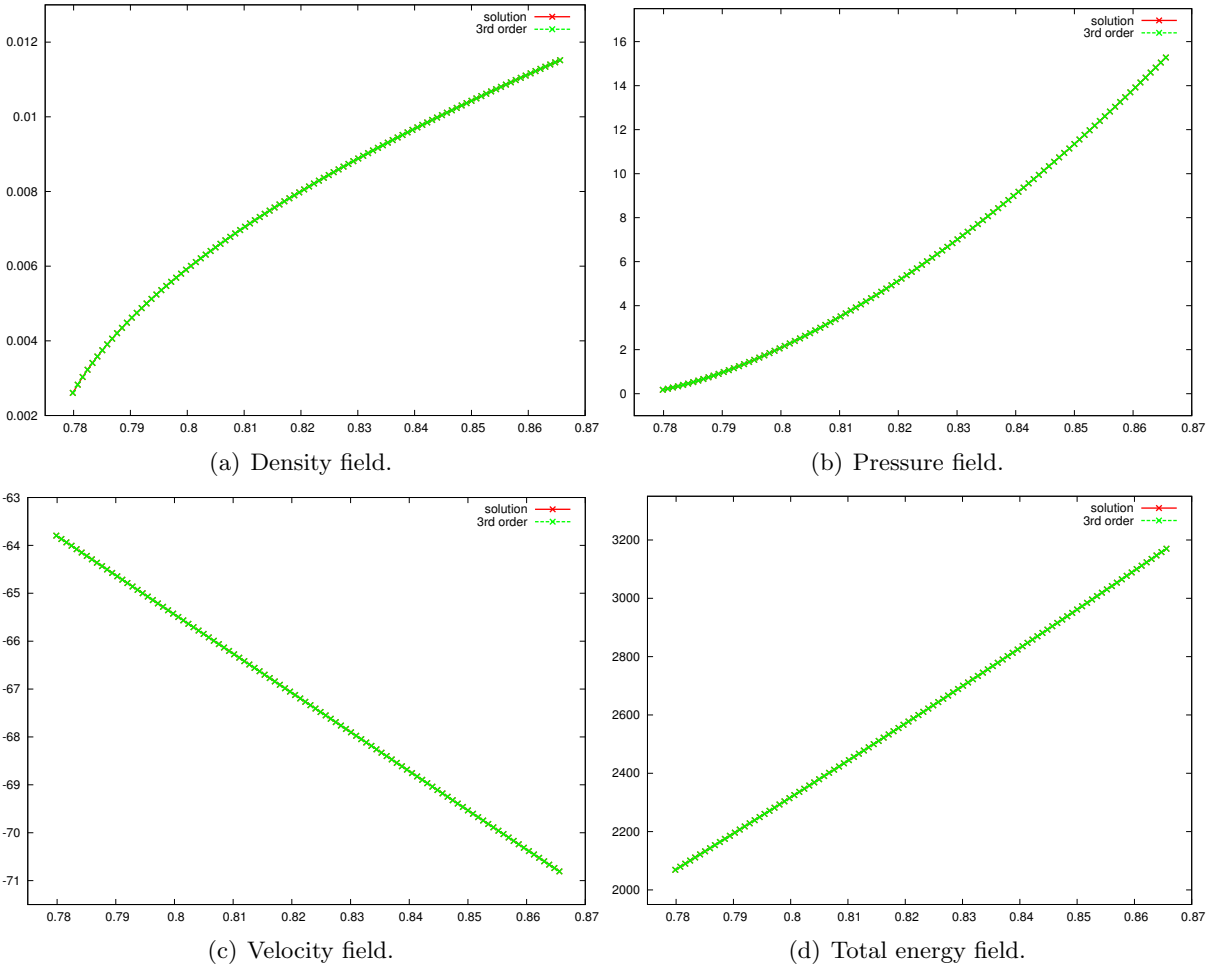
Kidder isentropic compression

Now, we make use of the Kidder isentropic compression introduced in [54], to ensure once again that the high accuracy order of our scheme is reached. This test case will be more detailed in the next Chapter 4 Section 4.2.4. It corresponds to an isentropic compression of a shell filled with

	L_1		L_2		L_∞	
ΔX	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{50}$	3.44E-4	2.19	1.07E-3	1.80	5.63E-3	1.55
$\frac{1}{100}$	7.53E-5	2.68	3.09E-4	2.31	1.92E-3	1.97
$\frac{1}{200}$	1.18E-5	2.99	6.22E-5	2.65	4.96E-4	2.40
$\frac{1}{400}$	1.48E-6	3.22	9.94E-6	2.82	9.23E-5	2.25
$\frac{1}{800}$	1.59E-7	-	1.40E-6	-	1.95E-5	-

Table 2.6: Rate of convergence on the velocity for the third-order DG scheme with slope limiter computed in the smooth case on the $[0, 1]$ domain, at time $t = 0.8$ with a CFL$= 0.1$.

perfect gas, initially located at $[0.9, 1]$ and which collapses at time $t = \tau$. This problem admits an analytical self-similar solution and thus we can compare the numerical solution computed by our DG scheme and the analytical solution, refer to Figure 2.16.



(a) Density field.

(b) Pressure field.

(c) Velocity field.

(d) Total energy field.

Figure 2.16: Solution obtained for the Kidder isentropic compression over 100 cells on the $[0.9, 1]$ domain at time $t = \frac{\tau}{2}$: comparison between the analytical solution and the solution obtained with the third-order DG scheme.

Using this analytical solution we compute the global truncation error corresponding to the

	L_1		L_2		L_∞	
ΔX	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{50}$	1.04E-5	3.62	1.94E-4	3.73	6.71E-3	3.52
$\frac{1}{100}$	8.46E-7	3.33	1.46E-5	3.44	5.87E-4	3.38
$\frac{1}{200}$	8.38E-8	3.17	1.34E-6	3.23	5.64E-5	3.23
$\frac{1}{400}$	9.33E-9	3.08	1.43E-7	3.12	6.00E-6	3.13
$\frac{1}{800}$	1.10E-9	-	1.64E-8	-	6.85E-7	-

Table 2.7: Rate of convergence on the velocity for the third-order DG scheme, computed in the case of the Kidder isentropic compression, on the $[0.9, 1]$ domain, at time $t = \frac{\tau}{2}$ with a CFL$= 0.1$.

third DG scheme scheme and display in Table 2.7. These results confirm the high accuracy of the considered scheme. Concerning the limitation, in this particularly smooth isentropic case, we note that the numerical errors are almost identical, see Table 2.8. The scheme remains third-order accurate.

	L_1		L_2		L_∞	
ΔX	E_{L_1}	q_{L_1}	E_{L_2}	q_{L_2}	E_{L_∞}	q_{L_∞}
$\frac{1}{50}$	9.81E-6	3.54	1.89E-4	3.69	6.72E-3	3.52
$\frac{1}{100}$	8.46E-7	3.33	1.46E-5	3.44	5.87E-4	3.38
$\frac{1}{200}$	8.38E-8	3.17	1.34E-6	3.23	5.64E-5	3.23
$\frac{1}{400}$	9.33E-9	3.08	1.43E-7	3.12	6.00E-6	3.13
$\frac{1}{800}$	1.10E-9	-	1.64E-8	-	6.85E-7	-

Table 2.8: Rate of convergence on the velocity for the third-order DG scheme with slope limiter based on the linearized Riemann invariants, computed in the case of the Kidder isentropic compression, on the $[0.9, 1]$ domain, at time $t = \frac{\tau}{2}$ with a CFL$= 0.1$.

In the two first chapters, we have presented a DG discretization using Taylor basis for solving one and two-dimensional scalar conservations laws on general unstructured grids and also one-dimensional gas dynamics equations written under Lagrangian form. Numerical flux has been designed to enforce L_2 stability and an entropy inequality in the case of gas dynamics equations. A robust and accurate limitation procedure has been constructed, based on the polynomial representation of the linearized Riemann invariants in the non-linear case. All these results have been summarized in our article entitled *Cell-centered discontinuous Galerkin discretizations for two-dimensional scalar conservation laws on unstructured grids and for one-dimensional Lagrangian hydrodynamics*, published in Computers & Fluids, refer to [104].

Chapter 3

Cell-centered Finite Volume Lagrangian scheme on moving mesh

This Chapter is dedicated to the study of multidimensional Lagrangian hydrodynamics. In the first Section 3.1, we briefly recall the evolution of the researches based on the implementation of numerical schemes for the Lagrangian hydrodynamics. In Section 3.2, we describe the important concepts related to kinematics of fluid motion such as the Lagrangian and Eulerian formalisms briefly introduced in the second chapter. Then in Section 3.3, we recall the first order cell-centered EUCCLHYD scheme (Explicit Unstructured Lagrangian HYDrodynamics) developed by Pierre-Henri Maire and Jérome Breil, which is a Lagrangian Finite Volume scheme on moving mesh. This particular scheme in its first-order version is the corner stone of our DG formulation. Let us emphasize that in its first-order version our DG method naturally recovers the first-order EUCCLHYD scheme.

3.1 Cell-Centered Lagrangian scheme overview

Fluid dynamics can be described by two different formalisms: the Eulerian description and the Lagrangian description. In the former, the conservation laws are written using a fixed reference frame whereas in the latter they are written through the use of a moving reference frame that follows the fluid motion. Lagrangian representation is particularly well adapted to describe the time evolution of fluid flows contained in regions undergoing large shape changes due to strong compressions or expansions. Moreover, in this representation, there is no mass flux across the boundary surface of a control volume moving with the fluid velocity. Therefore, Lagrangian formalism provides a natural framework to track accurately material interfaces present in multi-material compressible fluid flows. All these reasons lead to the use of the Lagrangian description to model the compressible flows encountered in the context of Inertial Confinement Fusion. Indeed, the hydrodynamic flow resulting from the implosion of an ICF target is a source of very intense shock and rarefaction waves. The size of the domain occupied by the fluid varies strongly over time. Furthermore, the target being composed of several distinct materials, this type of flow generally contains several interfaces that need to be followed with accuracy during implosion. Bearing this context in mind, our main motivation consists in deriving a robust and accurate numerical method that solves the compressible gas dynamics equations written under Lagrangian form. Before we proceed any further in the description of this method, we start by presenting a brief historical overview about Lagrangian numerical methods. For a thoroughly presentation of the classical computational methods in Lagrangian and Eulerian hydrocodes, the interested reader may refer to the review paper of Benson [7].

In contrast to Eulerian methods, classical Lagrangian methods are characterized by a moving computational grid. Thus, in the Lagrangian framework, one has to discretize not only the physical conservation laws but also the mesh motion. Moreover, the numerical fluxes of the physical conservation laws must be determined in a compatible manner with the cell deformation. More precisely, the zone volume that is computed directly from its coordinates must be equal to the zone volume that is deduced from solving the discrete volume equation, otherwise called the Geometric Conservation Law (GCL). This critical requirement related to the GCL is the cornerstone on which any proper multi-dimensional Lagrangian scheme should rely.

The most natural method to fulfill this requirement consists in employing a staggered discretization wherein the velocity is located at the nodes whereas the thermodynamic variables (density, pressure and specific internal energy) are defined at the cell center. This choice leads to solve a partial differential equation for the specific internal energy. Although it is written under a non-conservative form, the adding of an artificial viscosity term renders this equation compatible with the Second Law of thermodynamics. This term, that mimics the physical viscosity, ensures a proper transformation of kinetic energy into internal energy through shock waves. The first staggered scheme has been introduced for one-dimensional flows by von Neumann and Richtmyer during the 40's in Los Alamos and published later in their seminal paper [105]. The bi-dimensional extension to elastic-plastic flows has been derived by Wilkins in [107]. In its initial version, this scheme did not conserve the total energy, in addition it was sensitive to numerical spurious modes. Since two decades, many improvements have been done in order to solve theses flaws. In their paper [19], Caramana and Shashkov show that with an appropriate discretization of the subzonal forces resulting from subzonal pressures, hourglass motion and spurious vorticity can be eliminated. The introduction of a compatible discretization of the divergence and gradient operators leads to a compatible staggered scheme which conserves the total energy [14, 18]. Regarding the artificial viscosity, its discretization has also been considerably improved. First, by introducing formulations for multidimensional shock wave computations in [17] and then by using a discretization based on mimetic finite difference method in [15, 68]. With all these improvements, staggered Lagrangian schemes are accurate and robust numerical methods, which can produce impressive results, even on unstructured polygonal grids [16]. We want also to mention the recent papers [13, 81] wherein two-dimensional staggered discretizations are developed by constructing artificial viscosities which relies on the use of an approximate Riemann solver.

Up to our knowledge, the interpretation of the staggered schemes of Wilkins and Goad [107, 41] by means of the finite element method has been initially introduced by Lascaux at the beginning of the 70's in [61, 62]. This finite element interpretation has been further developed producing various interesting staggered schemes. In [3], Barlow presents a compatible finite element Lagrangian hydrodynamics algorithm used in a multi-material Arbitrary Lagrangian Eulerian (ALE) strategy. We also note the development of a variational multi-scale stabilized approach in finite element computation of Lagrangian hydrodynamics, where a piecewise linear approximation was adopted for the variables [93, 92]. The case of Q1/P0 finite element is studied in [95], where the kinematic variables are represented using a piecewise linear continuous approximation, while the thermodynamic variables utilize a piecewise constant representation. More recently, a high-order extension of the staggered discretization based on a curvilinear finite element method has been derived in [34, 35]. The corresponding artificial viscosity being described in [57]. This particular scheme has proved to be highly capable producing impressive results.

As motioned by Barlow in [4], the staggered grid schemes employed in most hydro-codes have been remarkably successful over the past decades in solving complex multi-dimensional compressible fluid flows. However, they clearly have some theoretical and practical deficiencies. Mesh imprinting and symmetry breaking are important examples. The need to use artificial viscosities, hourglass filters and sub-zonal pressure schemes is also undesirable. A staggered scheme is also inelegant as all variables are not conserved over the same space. High resolution cell-centered Lagrangian Godunov schemes can overcome some of these problems. In contrast to staggered approach, these conservative schemes are based on a cell-centered placement of all the fluid variables and solve the total energy equation. The numerical face fluxes are determined by means of one-dimensional approximate Riemann solvers in the direction normal to the cell interface. The main difficulty lies in the definition of consistent Lagrangian nodal velocities with which to move the computational mesh. This probably explains why progress has been slow in extending these ideas to Lagrangian schemes, while Eulerian Godunov methods have been well established for a long time.

A first attempt has been done by Dukowicz and co-workers while developing CAVEAT code in Los Alamos during the 80's. In [2] they proposed to use a weighted least squares algorithm to compute the vertex velocity by requiring that the vertex velocity projected in the direction of a face normal should equal the normal velocity resulting from the approximate Riemann solver. It turns out that this algorithm is capable of generating additional spurious components in the vertex velocity field. Thus, it leads to an artificial grid motion which requires a very expensive treatment [36]. This flaw follows from the fact that the volume flux is not compatible with the node motion in the sense that the GCL is not satisfied.

A solution to circumvent this problem consists in deriving cell-centered discretizations based on a total Lagrangian formulation of the gas dynamics equations. That it is, a formulation wherein the divergence and gradient operators are expressed in terms of the Lagrangian coordinates. In this framework, the computational grid is fixed, however one has to follow the time evolution of the Jacobian matrix, also called deformation gradient tensor, associated to the Lagrange-Euler flow map. In addition, a new difficulty arises due to the fact that the gas dynamics system written under total Lagrangian form is only weakly hyperbolic [50, 70, 83, 33, 31]. This loss of strict hyperbolicity is related to the supplementary conservation law that results from the time evolution of the deformation gradient tensor. Following this approach, a Lagrangian Discontinuous Galerkin-type method has been successfully developed in [1, 70]. The research presented in the present manuscript was strongly inspired by the work done by Loubère during his PhD thesis, supervised by Abgrall and Ovadia, and can be viewed as its natural extension. However, the use of this method is limited to a representation in the initial configuration since it cannot be rigorously interpreted as a moving mesh method. This was due to the loss of compatibility between the two configurations based on Eulerian and Lagrangian coordinates. This issue has been addressed by Després and Mazeran in their seminal paper [33] where they propose a first-order cell-centered discretization of the gas dynamics equations that can be written in both total and updated Lagrangian forms thus leading to a moving mesh finite volume scheme which satisfies exactly the GCL. The numerical fluxes are defined at the node in the following manner: there is a unique nodal velocity and one nodal pressure per each cell surrounding the node. These fluxes are computed through the use of a node-centered approximate Riemann solver. This scheme conserves the momentum and the total energy and also satisfies an entropy inequality. It has been called GLACE in [20] where a high-order multi-dimensional extension is provided, the acronym GLACE being shorthand for Godunov-type LAgrangian scheme Conservative for total Energy. A thorough study of the properties of the GLACE node-centered

solver reveals a strong sensitivity to the cell aspect ratio which can leads to severe numerical instabilities [89]. This problem is critical for ICF Lagrangian simulations characterized by computational grids that frequently contain very high aspect ratio cells. To overcome this difficulty, Maire, Abgrall and Breil have proposed an alternative scheme [78] that successfully solves the cell aspect ratio problem and keeps the compatibility between fluxes discretization and vertices velocity computation. This first-order scheme is also characterized by node-based fluxes, it conserves the momentum and the total energy, and fulfills a local entropy inequality. Its main feature lies in the discretization of the pressure gradient, which is designed using two pressures at each node of a cell, each nodal pressure being associated with the direction of the unit outward normals related to the edges originating from the node. These nodal pressures are linked to the nodal velocity thanks to approximate Riemann invariants. This scheme, which is called EUCCLHYD (Explicit Unstructured Cell-Centered Lagrangian HYDrodynamics), has been extended to higher order using first a classical MUSCL-type approach [79], and then a prolongation of works initiated by Ben-Artzi and Falcovitz [6] concerning the Generalized Riemann Problem [74]. The application of this scheme to the tri-dimensional geometry is presented in [82]. The extension to the bi-dimensional axisymmetric geometry is described in [73]; this latter version being absolutely necessary for simulations of a spherical ICF target implosion. More recently, the concept of sub-cell forces which has been initially introduced in the context of staggered discretization [18] has been extended to cell-centered schemes. This formulation covers different centered schemes proposed previously and is described in [75, 76]. This framework, adapted here to cell-centered discretization, should provide deeper understanding of relations between cell-centered and staggered Lagrangian schemes. Note that this formulation has quite a lot in common with the approach taken very recently in [4]. In this work, the author also draws an analogy between forces and the momentum fluxes in the momentum equation and uses this to produce a total energy conservation equation expressed in work form as it is has been done in [75], which allows the total energy to be evolved for forces which include terms that appear as a pressure average and a linear vector artificial viscosity.

In [55, 56], Kluth and Després have developed a cell-centered Lagrangian scheme on unstructured mesh for hyperelasticity. This method is based on the GLACE scheme [20] for compressible gas dynamics. They make use of the multiplicative decomposition of the gradient of deformation and the entropy property to derive their scheme. They also prove the compatibility of this discretization with the discretization of the specific volume. The present work on the discretization of the flow map and of the deformation gradient tensor has also been motivated by the research made by Kluth and Després. Thus, the development of our high-order cell-centered DG schemes presents a lot of similarities with the work presented in [55, 56].

Following the approach described in [1, 70], a high-order discontinuous Galerkin spectral finite element method has been developed for solving the gas dynamics equations under total Lagrangian formalism [51]. The nodal solver uses the HLLC approximate Riemann solver to compute the velocities of the vertex, and a HWENO (Hermite WENO) reconstruction algorithm is developed and used as limiters for the RKDG time discretization. However, the Piola compatibility condition is not ensured properly. This is due to the loss of consistency in the definition of the deformation gradient tensor in the cells and on the boundaries of the cells. Moreover, the deformation tensor being discretized by means of a Finite Volume method over the cells, this scheme is limited to the representation of computational meshes with straight-line edges. Last, it is worth pointing out the recent work presented in [87, 10] which consists in a high-order curvilinear finite volume method that solves the Lagrangian gas dynamics. In [10], a first-order Finite Volume scheme has been developed on total unstructured meshes wherein

the cell interfaces are parametrized by conicals. This original GCL satisfying approach relies on a weak formulation to compute the nodal velocity using an acoustic Riemann solver approximation. Let us emphasize that in its first-order version this method naturally recovers the first-order EUCCLHYD scheme.

In parallel, other type of Lagrangian schemes have been developed by Cheng and Shu during the past years. In [21], they present Lagrangian type schemes for solving the Euler equations of compressible gas dynamics both in the Cartesian and in the cylindrical coordinates. These schemes are based on high-order ENO reconstruction. In these particular schemes, the physical variables are computed through the use of a high-order ENO conservative reconstruction wherein the fluxes are evaluated by means of three different typical numerical fluxes (Godunov flux, LF flux and HLLC flux). The determination of the vertex velocity is obtained from the conserved variables. The time discretization is performed through a TVD Runge-Kutta type method. In [22], they demonstrate that the accuracy degeneracy phenomenon observed for the high-order Lagrangian type scheme is due to the error from the quadrilateral mesh with straight-line edges, which restricts the accuracy of the resulting scheme to at most second-order. Thus, in [22] they extend the discretization presented in [21] to the third-order on curvilinear meshes. The accuracy test given in this paper shows that the third-order Lagrangian type scheme can actually obtain uniformly third-order accuracy even on distorted meshes by using curvilinear meshes. These particular Lagrangian type schemes have proved their high capability over a large number of test cases. However, the GCL inherent to the Lagrangian formalism is not taken into account in the discretization of physical variables. In [69], they present the Lax-Wendroff (LW) type time discretization applied on their high-order ENO Lagrangian type scheme. Comparing with the Runge-Kutta time discretization procedure, an advantage of the LW time discretization is the apparent saving in computational cost and memory requirement. They also described in [23] an extension of their scheme to cylindrical geometry through a discretization on the true volume.

We conclude this review by saying that cell-centered finite volume schemes [20, 74] that fulfill the GCL compatibility requirement seem to be a promising alternative to the usual staggered finite difference discretizations [18] for compressible Lagrangian hydrodynamics. These cell-centered Godunov-type discretizations exhibit the good property of being naturally conservative. This point is of great importance since convergent conservative schemes are known to converge to weak solutions in the presence of shocks by the Lax-Wendroff theorem [63]. Furthermore, they do not require the use of an artificial viscosity since the numerical dissipation built in the approximate Riemann solver automatically converts kinetic energy into internal energy through shock waves. Last, these cell-centered schemes allow a straightforward implementation of conservative remapping methods when they are used in the context of the Arbitrary Lagrangian Eulerian (ALE) strategy.

3.2 Lagrangian and Eulerian descriptions

The aim of this section is to recall briefly the main notions of kinematics required to introduced the Lagrangian and Eulerian descriptions, see [48].

3.2.1 Notations

Let us introduce the d-dimensional Euclidean space $\mathbb{R}^d$, where d is an integer ranging from 1 to 3. Let $\mathcal{D}$ be a region of $\mathbb{R}^d$ filled by a moving fluid. The fluid flow is described mathematically by the continuous transformation, $\boldsymbol{\Phi}$, of $\mathcal{D}$ into itself as

$$\boldsymbol{\Phi} : \boldsymbol{X} \longrightarrow \boldsymbol{x} = \boldsymbol{\Phi}(\boldsymbol{X}, t). \tag{3.1}$$

Here, t, which is a non-negative real number, denotes the time and $\boldsymbol{X} = (X, Y, Z)$ is the position a time $t = 0$ of a particle moving with the fluid which occupies the position $\boldsymbol{x}$ at time $t > 0$. It turns out that $\boldsymbol{x}$ is determined as a function of $\boldsymbol{X}$ and t, and the fluid flow is represented by the transformation $\boldsymbol{x} = \boldsymbol{\Phi}(\boldsymbol{X}, t)$. By definition $\boldsymbol{\Phi}$ satisfies $\boldsymbol{\Phi}(\boldsymbol{X}, 0) = \boldsymbol{X}$. For a fixed $\boldsymbol{X}$, the time evolution of (3.1) describes the trajectory of a fluid particle initially located at $\boldsymbol{X}$. Let us consider $\omega = \omega(t)$ a moving sub-region of $\mathcal{D}$ at time t. ω corresponds to the image of a fixed sub-region Ω in the flow map with $\omega = \{\boldsymbol{x} = \boldsymbol{\Phi}(\boldsymbol{X}, t) | \boldsymbol{X} \in \Omega\}$. The boundaries of ω and Ω are respectively $\partial\omega$ and $\partial\Omega$, and their unit outward normals are $\boldsymbol{n}$ and $\boldsymbol{N}$, refer to Figure 3.1.

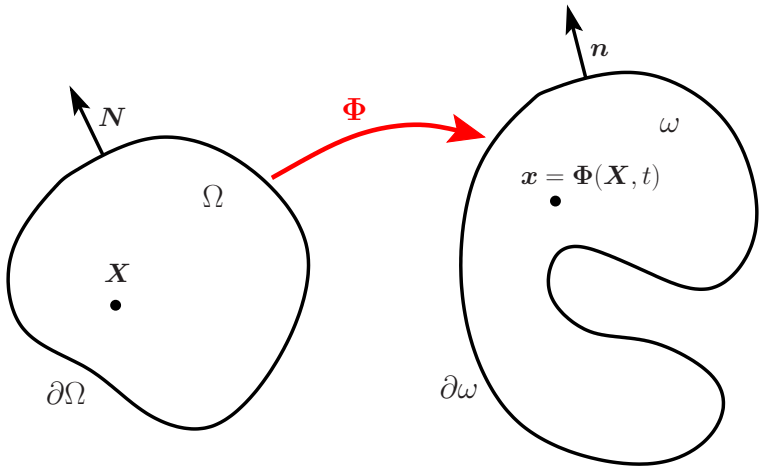


Figure 3.1: Notation for the flow map.

At this point, we can introduce the two usual descriptions of the flows, namely the Lagrangian description and the Eulerian description. The Lagrangian description, otherwise called material description, consists in observing the fluid by following the motion of fluid particles from their initial location. The independent variables used for this description are $(\boldsymbol{X}, t)$. On the other hand the Eulerian description, otherwise called spatial description, consists in observing the fluid at fixed locations in the space. The independent variables used for this description are $(\boldsymbol{x}, t)$. We notice that through the use of the transformation (3.1) any fluid quantity f which is expressed in terms of Eulerian variables $(\boldsymbol{x}, t)$ can also be expressed in terms of Lagrangian variables $(\boldsymbol{X}, t)$ and conversely. To emphasize the used variables and for the sake of conciseness, the same notation is used to denote the value of the physical quantity regardless the employed description

$$f = f(\boldsymbol{x}, t) = f(\boldsymbol{\Phi}(\boldsymbol{X}, t), t) = f(\boldsymbol{X}, t). \tag{3.2}$$

To be more precise, $f(\boldsymbol{X}, t)$ is the value of the physical quantity experienced at time t by the fluid article initially located at $\boldsymbol{X}$ whereas $f(\boldsymbol{x}, t)$ is the value of f experienced by the fluid particle which is located at position $\boldsymbol{x}$ at time t.

3.2.2 Material time derivative

Let f be a fluid variable with a sufficient smoothness to allow the computation of its first partial derivatives with respect to both Lagrangian and Eulerian variables. First, we introduce the material derivative of f which measures the rate of change of f following a fluid particle along its motion as

$$\frac{\mathrm{d}\, f}{\mathrm{d}t} \equiv \frac{\partial}{\partial t} f(\boldsymbol{X}, t). \tag{3.3}$$

Note that it corresponds to the partial time derivative in the Lagrangian description. The velocity of a fluid particle is denoted $\boldsymbol{U}$ and is defined as the material derivative of the position vector $\boldsymbol{x}$

$$\boldsymbol{U}(\boldsymbol{X}, t) = \frac{\mathrm{d}\, \boldsymbol{x}}{\mathrm{d}t}. \tag{3.4}$$

As defined, $\boldsymbol{U}$ is a function of the Lagrangian variables, however it is possible to also express it in terms of the Eulerian variables writing $\boldsymbol{U} = \boldsymbol{U}(\boldsymbol{x}, t)$. Knowing the velocity field expressed in terms of the Eulerian coordinates, it is thus possible to determine the fluid flow by solving the system of ordinary differential equations

$$\frac{\mathrm{d}\, \boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{U}(\boldsymbol{x}, t), \quad \boldsymbol{x}(\boldsymbol{X}, 0) = \boldsymbol{X}, \tag{3.5}$$

which corresponds to the trajectory equations of the fluid flow. Considering the physical quantity f expressed in terms of the Eulerian variables, that is $f = f(\boldsymbol{x}, t)$, we compute its material derivative employing (3.2) and the chain rule of composite derivative to get

$$\frac{\mathrm{d}\, f}{\mathrm{d}t} = \frac{\partial\, f}{\partial t} + \left(\frac{\mathrm{d}\, \boldsymbol{x}}{\mathrm{d}t}\right) \boldsymbol{.}\, \boldsymbol{\nabla}_x f, \tag{3.6}$$

where $\boldsymbol{\nabla}_x$ denotes the gradient operator with respect to Eulerian coordinates, which writes in the chosen coordinate system as $\boldsymbol{\nabla}_x f = (\frac{\partial}{\partial x} f, \frac{\partial}{\partial y} f, \frac{\partial}{\partial z} f)^{\mathrm{t}}$. Substituting the velocity definition into (3.6) leads to

$$\frac{\mathrm{d}\, f}{\mathrm{d}t} = \frac{\partial\, f}{\partial t} + \boldsymbol{U} \boldsymbol{.}\, \boldsymbol{\nabla}_x f. \tag{3.7}$$

This last equation may be interpreted as expressing the time rate of change of an arbitrary physical quantity $f = f(\boldsymbol{x}, t)$ apparent to a observer located on the moving particle instantaneously at the position $\boldsymbol{x}$. Let $\boldsymbol{f}$ denotes a vector valued function, we compute its material derivative applying (3.7) componentwise to finally get

$$\frac{\mathrm{d}\, \boldsymbol{f}}{\mathrm{d}t} = \frac{\partial\, \boldsymbol{f}}{\partial t} + (\nabla_x \boldsymbol{f})\, \boldsymbol{U}, \tag{3.8}$$

where $\nabla_x \boldsymbol{f}$ represents the gradient tensor defined by (A.25).

3.2.3 Deformation gradient tensor

Let $\boldsymbol{X}_0$ be the initial position vector of a specific fluid particle. Its image in the fluid flow transformation writes $\boldsymbol{x}_0 = \boldsymbol{\Phi}(\boldsymbol{X}_0, t)$. Let $\boldsymbol{x} = \boldsymbol{\Phi}(\boldsymbol{X}, t)$ denotes the vector position of a fluid particle at time t which was initially located in the vicinity of $\boldsymbol{X}_0$. A first-order Taylor expansion of the transformation $\boldsymbol{\Phi}$ in Lagrangian variables yields

$$\begin{aligned}\boldsymbol{x} &= \boldsymbol{\Phi}(\boldsymbol{X}_0 + \boldsymbol{X} - \boldsymbol{X}_0, t),\\ &= \boldsymbol{\Phi}(\boldsymbol{X}_0) + \nabla_X \boldsymbol{\Phi}(\boldsymbol{X}_0)(\boldsymbol{X} - \boldsymbol{X}_0) + O(|\boldsymbol{X} - \boldsymbol{X}_0|^2).\end{aligned}$$

Here, $\nabla_X \boldsymbol{\Phi}$ denotes the gradient tensor of $\boldsymbol{\Phi}$ with respect to the Lagrangian coordinates, it represents the Jacobian matrix of the transformation characterizing the fluid flow. This Jacobian matrix is also named the deformation gradient tensor and is defined as

$$\mathsf{F} = \nabla_X \boldsymbol{\Phi} = \nabla_X \boldsymbol{x}. \tag{3.9}$$

With the use of the trajectory equation (3.5), the deformation gradient tensor definition (3.9) yields an equation on time rate of change of F

$$\frac{\mathrm{d}\,\mathsf{F}}{\mathrm{d}t} = \nabla_X \boldsymbol{U}. \tag{3.10}$$

This equation characterizes the time evolution of the Jacobian matrix. Now, we compute the material time derivative of the determinant of this Jacobian matrix. Knowing that $J = \det F$, we make use of the chain rule of composed derivatives (A.27) in the following manner

$$\begin{aligned}\frac{\mathrm{d}\,J}{\mathrm{d}t} &= \frac{\partial}{\partial \mathsf{F}}(\det \mathsf{F}) : \frac{\mathrm{d}\,\mathsf{F}}{\mathrm{d}t},\\ &= (\det \mathsf{F})\mathsf{F}^{-\mathrm{t}} : \frac{\mathrm{d}\,\mathsf{F}}{\mathrm{d}t}, \quad \text{thanks to (A.28)}\\ &= J\mathsf{F}^{-\mathrm{t}} : \frac{\mathrm{d}\,\mathsf{F}}{\mathrm{d}t}.\end{aligned} \tag{3.11}$$

Let us point out that if S and T are two arbitrary second-order tensors, their inner product is defined by $\mathsf{S} : \mathsf{T} = \mathrm{tr}(\mathsf{S}^{\mathrm{t}}\mathsf{T})$, refer to Appendix A.1.3. This relation expresses the time evolution of the Jacobian J by means of the material derivative of the deformations gradient tensor. Using (3.10), this relations yields

$$\begin{aligned}\frac{\mathrm{d}\,J}{\mathrm{d}t} &= J\mathsf{F}^{-\mathrm{t}} : \nabla_X \boldsymbol{U},\\ &= J\mathsf{F}^{-\mathrm{t}} : (\nabla_x \boldsymbol{U})\ (\nabla_X \boldsymbol{x}),\\ &= J\mathsf{F}^{-\mathrm{t}}\mathsf{F}^{\mathrm{t}} : \nabla_x \boldsymbol{U}, \quad \text{thanks to (3.9) and (A.18a)}\\ &= J\,\mathrm{tr}(\nabla_x \boldsymbol{U}).\end{aligned}$$

Noticing that the right-hand side of the last line in the previous equation is proportional to the trace of the velocity gradient tensor which is nothing but the divergence of the velocity field (refer to (A.30)), material derivative of the determinant of the Jacobian matrix finally reads

$$\frac{\mathrm{d}\,J}{\mathrm{d}t} = J\nabla_x \cdot \boldsymbol{U}. \tag{3.12}$$

Here $\nabla_x \cdot \boldsymbol{U}$ denotes the divergence of the velocity with respect to Eulerian coordinates, which writes in the chosen coordinate system $\nabla_x \cdot \boldsymbol{U} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}$, where (u, v, w) denote the components of the velocity field.

3.2.4 Transformation formulas

Let $\mathrm{d}V$ and $\mathrm{d}\boldsymbol{X}$ denote a Lagrangian volume element and an infinitesimal displacement, and $\mathrm{d}v$ and $\mathrm{d}\boldsymbol{x}$ their corresponding quantities in the Eulerian space through the transformation of the flow. These volumes and displacements can be related through the following formulas

$$\mathrm{d}v = J\mathrm{d}V, \tag{3.13}$$

$$\mathrm{d}\boldsymbol{x} = \mathsf{F}\mathrm{d}\boldsymbol{X}. \tag{3.14}$$

These formulas show that the Jacobian is a measure of the volume change, and the deformation gradient tensor quantifies the change of shape of infinitesimal vectors through the fluid motion. Knowing that the Eulerian and Lagrangian volume elements are related by (3.13), we introduce the Nanson formula which relates Eulerian and Lagrangian area elements.

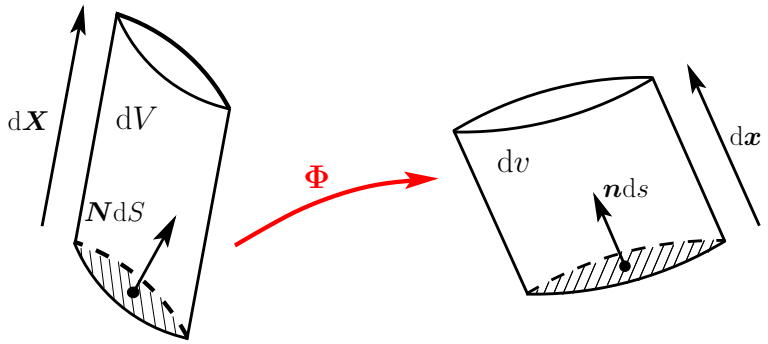


Figure 3.2: Area elements transformation.

Let us consider the volume element $\mathrm{d}V$, refer to Figure 3.2, computed as the infinitesimal volume corresponding to the cylinder characterized by the base $\mathrm{d}S$ and the displacement vector $\mathrm{d}\boldsymbol{X}$

$$\mathrm{d}V = \mathrm{d}\boldsymbol{X} \cdot \boldsymbol{N}\mathrm{d}S. \tag{3.15}$$

Using the same reasoning, the corresponding volume element in the Eulerian space writes as

$$\mathrm{d}v = \mathrm{d}\boldsymbol{x} \cdot \boldsymbol{n}\mathrm{d}s. \tag{3.16}$$

Substituting $\mathrm{d}v = J\mathrm{d}V$ and $\mathrm{d}\boldsymbol{x} = \mathsf{F}\mathrm{d}\boldsymbol{X}$ in equations (3.15) and (3.16), we obtain

$$\begin{aligned} J\boldsymbol{N}\mathrm{d}S \cdot \mathrm{d}\boldsymbol{X} &= \mathsf{F}\mathrm{d}\boldsymbol{X} \cdot \boldsymbol{n}\mathrm{d}s, \\ &= \mathsf{F}^{\mathrm{t}}\boldsymbol{n}\mathrm{d}s \cdot \mathrm{d}\boldsymbol{X}. \end{aligned}$$

Since this equality holds for an arbitrary displacement vector $\mathrm{d}\boldsymbol{X}$, we obtain

$$\boldsymbol{n}\mathrm{d}s = J\mathsf{F}^{-\mathrm{t}}\boldsymbol{N}\mathrm{d}S. \tag{3.17}$$

This formula expresses the transport of an oriented area element through the flow deformation. It is known as the Nanson formula and is one of the main ingredient to derive the Lagrangian forms of the conservation laws of fluid mechanics. The integration of the left-hand side of equation (3.17) over the boundary of a Lagrangian fluid volume gives the well known geometric interpretation following

$$\int_{\partial\omega} \boldsymbol{n}\mathrm{d}s = \boldsymbol{0},$$

which means that the integral of the unit outward normal over a close surface is equal to zero. Substituting (3.17) in this last formula yields

$$\int_{\partial\Omega} J\mathsf{F}^{-\mathrm{t}}\boldsymbol{N}\mathrm{d}S = \int_{\Omega} \nabla_X \cdot (J\mathsf{F}^{-\mathrm{t}})\,\mathrm{d}V = \mathbf{0}. \tag{3.18}$$

Since this relation holds for any Lagrangian fluid volume Ω, this geometric interpretation rewrites

$$\nabla_X \cdot (J\mathsf{F}^{-\mathrm{t}}) = \mathbf{0}. \tag{3.19}$$

This is the Piola identity which is well known in continuum mechanics [84, 94]. It ensures the compatibility between the two configurations based on Eulerian and Lagrangian coordinates.

3.2.5 Reynolds transport formula

Let $\omega = \omega(t)$ denotes an arbitrary moving volume and let $f(\boldsymbol{x}, t)$ be a scalar valued function representing some physical quantity. Transport formula consists in expressing the time rate of change of the integral of the physical quantity f over the moving volume ω, *i.e*,

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\omega(t)} f(\boldsymbol{x}, t)\,\mathrm{d}v. \tag{3.20}$$

Here, we derive the Reynolds transport formula which corresponds to a volume moving with the fluid velocity. We consider $f = f(\boldsymbol{x}, t)$ as a continuously differentiable function over the region ω moving with the fluid velocity $\boldsymbol{U}$. Since ω is time-dependent, we cannot compute the time derivative (3.20) directly. To do so, we shall use the following change of variables

$$\int_{\omega} f(\boldsymbol{x}, t)\,\mathrm{d}v = \int_{\Omega} f(\boldsymbol{X}, t) J\,\mathrm{d}V. \tag{3.21}$$

In writing this equality, we have introduced the fixed Lagrangian coordinate, $\boldsymbol{X}$, as new variable of integration and used the fact that for this change of variable the change of volume element is ruled by (3.13). The initial volume occupied by the fluid, Ω is mapped into ω at time t through the flow transformation, *i.e.*, $\omega = \boldsymbol{\Phi}(\Omega, t)$. Thanks to the above change of variables, we can compute the time derivative (3.20) as follows

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\int_{\omega} f(\boldsymbol{x}, t)\,\mathrm{d}v &= \int_{\Omega}\left(J\frac{\mathrm{d}\,f}{\mathrm{d}t} + f\frac{\mathrm{d}\,J}{\mathrm{d}t}\right)\mathrm{d}V, \\
&= \int_{\Omega} J\left(\frac{\mathrm{d}\,f}{\mathrm{d}t} + f\nabla_x \cdot \boldsymbol{U}\right)\mathrm{d}V, && \text{thanks to (3.12)} \\
&= \int_{\omega}\left(\frac{\mathrm{d}\,f}{\mathrm{d}t} + f\nabla_x \cdot \boldsymbol{U}\right)\mathrm{d}v. && \text{thanks to (3.13)}
\end{aligned}$$

Substituting the definition of the material derivative (3.7) in the right-hand side of the last line leads to the transport theorem otherwise named the Reynolds transport formula

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\omega} f(\boldsymbol{x}, t)\,\mathrm{d}v = \int_{\omega}\left(\frac{\partial\, f}{\partial t} + \nabla_x \cdot (f\boldsymbol{U})\right)\mathrm{d}v. \tag{3.22}$$

Through the use of the divergence theorem (A.34b), Reynolds transport formula can also be written under the following form

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\omega} f(\boldsymbol{x}, t)\,\mathrm{d}v = \int_{\omega}\frac{\partial\, f}{\partial t}\,\mathrm{d}v + \int_{\partial\omega} f\boldsymbol{U} \cdot \boldsymbol{n}\,\mathrm{d}s, \tag{3.23}$$

where $\boldsymbol{n}$ denotes the outward unit normal to the boundary $\partial\omega$. This equation states that the rate of increase of the total amount of f attached to the moving volume ω is balanced by the rate of increase of the total amount of f inside ω plus the net rate of outward flux of f transferred by the fluid through the boundary $\partial\omega$.

To extend the Reynolds transport formula to the vector valued function $\boldsymbol{f}$, we proceed with the computation of the time derivative of its volume integral as before

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_\omega \boldsymbol{f}(\boldsymbol{x},t)\,\mathrm{d}v &= \int_\Omega J\left(\frac{\mathrm{d}\,\boldsymbol{f}}{\mathrm{d}t} + \boldsymbol{f}\nabla_x \cdot \boldsymbol{U}\right)\mathrm{d}V,\\ &= \int_\omega \left(\frac{\mathrm{d}\,\boldsymbol{f}}{\mathrm{d}t} + \boldsymbol{f}\nabla_x \cdot \boldsymbol{U}\right)\mathrm{d}v,\\ &= \int_\omega \left(\frac{\partial\,\boldsymbol{f}}{\partial t} + (\nabla_x \boldsymbol{f})\boldsymbol{U} + \boldsymbol{f}\nabla_x \cdot \boldsymbol{U}\right)\mathrm{d}v. \qquad \text{thanks to (3.8)}\end{aligned}$$

The use of tensor identity (A.32b) in the right-hand side of the last line yields the vector form of the transport formula

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\omega \boldsymbol{f}(\boldsymbol{x},t)\,\mathrm{d}v = \int_\omega \left(\frac{\partial\,\boldsymbol{f}}{\partial t} + \nabla_x \cdot (\boldsymbol{f}\otimes\boldsymbol{U})\right)\mathrm{d}v, \tag{3.24}$$

where symbol $\otimes$ denotes the tensor product of two vectors defined by (A.1) and (A.2). Proceeding with the divergence theorem (A.34a) as before yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\omega \boldsymbol{f}(\boldsymbol{x},t)\,\mathrm{d}v = \int_\omega \frac{\partial\,\boldsymbol{f}}{\partial t}\,\mathrm{d}v + \int_{\partial\omega} \boldsymbol{f}(\boldsymbol{U}\cdot\boldsymbol{n})\,\mathrm{d}s. \tag{3.25}$$

A straightforward application of the Reynolds transport formula consists in deriving the continuity equation from the principle of mass conservation. This fundamental principle corresponds to the postulate that the mass of fluid in a material volume ω does not change as ω moves with the fluid. Let $\rho = \rho(\boldsymbol{x},t)$ denotes the mass density of the fluid, *i.e.*, the mass per unit volume. Mass conservation writes as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\omega \rho(\boldsymbol{x},t)\,\mathrm{d}v = 0. \tag{3.26}$$

Expressing the integrand in terms of the Lagrangian coordinate and integrating between initial and final time yields

$$\begin{aligned}\int_\omega \rho(\boldsymbol{x},t)\,\mathrm{d}v &= \int_\Omega \rho(\boldsymbol{X},t)J\,\mathrm{d}V,\\ &= \int_\Omega \rho^0(\boldsymbol{X})\,\mathrm{d}V,\end{aligned}$$

where ρ^0 is the initial density field. From this equation, it follows easily that

$$\frac{\mathrm{d}}{\mathrm{d}t}(\rho J) = 0, \tag{3.27a}$$

$$\rho J = \rho^0. \tag{3.27b}$$

Here, we have derived two forms of the Lagrangian equation of continuity. Combining (3.22) for $f = \rho$ and (3.26), we derive the Eulerian equation of continuity in its integral form

$$\int_\omega \left(\frac{\partial\,\rho}{\partial t} + \nabla_x \cdot (\rho\boldsymbol{U})\right)\mathrm{d}v = 0.$$

Since ω is an arbitrary volume, this last equation implies

$$\frac{\partial \rho}{\partial t} + \nabla_x \cdot (\rho \boldsymbol{U}) = 0, \tag{3.28}$$

which is the local form of the Eulerian equation of continuity. It is a necessary and sufficient condition for a motion to conserve the mass of each moving fluid volume. A by-product of the transport theorem and mass conservation is obtained by computing the time rate of change of the integral over a moving volume of the density times an arbitrary function

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_\omega \rho f \,\mathrm{d}v =& \frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \rho J f \,\mathrm{d}V \\ =& \int_\Omega \rho J \frac{\mathrm{d}\, f}{\mathrm{d}t}\,\mathrm{d}V. \qquad \text{thanks to (3.27a)}\end{aligned}$$

Coming back to the Eulerian coordinates in the right-hand side of the last line, we finally get

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\omega \rho f \,\mathrm{d}v = \int_\omega \rho \frac{\mathrm{d}\, f}{\mathrm{d}t}\,\mathrm{d}v. \tag{3.29}$$

This formula, which is also valid for vector valued function, will be of great interest in the development of the gas dynamics equations in Lagrangian integral form.

3.2.6 Eulerian and Lagrangian form of the gas dynamics equations

Here, we summarize the conservation laws which govern the time evolution of mass, volume, momentum and total energy of a moving fluid domain. The gas dynamics equations, otherwise named Euler equations, characterize fluids in which viscous stress and heat conduction are negligible. Here we recall the integral Eulerian form of the gas dynamics equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\omega \rho \,\mathrm{d}v = 0, \tag{3.30a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\omega \mathrm{d}v - \int_{\partial\omega} \boldsymbol{U} \cdot \boldsymbol{n} \,\mathrm{d}s = 0, \tag{3.30b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\omega \rho \boldsymbol{U} \,\mathrm{d}v + \int_{\partial\omega} P\boldsymbol{n} \,\mathrm{d}s = \boldsymbol{0}, \tag{3.30c}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\omega \rho E \,\mathrm{d}v + \int_{\partial\omega} P\boldsymbol{U} \cdot \boldsymbol{n} \,\mathrm{d}s = 0. \tag{3.30d}$$

We point out that this Eulerian integral form of the gas dynamics conservation laws is also referred as to the integral updated Lagrangian formulation. Knowing that the previous equations hold for any arbitrary moving fluid volume, we deduce that the local form of the gas dynamics equations with respect to the spatial (Eulerian) configuration writes

$$\frac{\partial \rho}{\partial t} + \nabla_x \cdot (\rho \boldsymbol{U}) = 0, \tag{3.31a}$$

$$\frac{\partial}{\partial t}(\rho \boldsymbol{U}) + \nabla_x \cdot (\rho \boldsymbol{U} \otimes \boldsymbol{U}) + \boldsymbol{\nabla}_x P = \boldsymbol{0}, \tag{3.31b}$$

$$\frac{\partial}{\partial t}(\rho E) + \nabla_x \cdot (\rho E \boldsymbol{U}) + \nabla_x \cdot (P\boldsymbol{U}) = 0, \tag{3.31c}$$

where ρ is the density of the fluid, $\boldsymbol{U}$ its velocity and E its total energy.

The thermodynamic closure of this system is obtained through the use of an equation of state, which writes $P = P(\rho, \varepsilon)$ where ε is the specific internal energy, $\varepsilon = E - \frac{\boldsymbol{U}^2}{2}$. For numerical application, again we use a gamma gas law, *i.e.*, $P = \rho(\gamma - 1)\varepsilon$ where γ is the polytropic index of the gas.

Now, if we want to write the gas dynamics equations in the Lagrangian form, we make use of the material derivative definition (3.7) to rewrite the equations of (3.31). For smooth fluid variables, gas dynamics equations can also be written under the non-conservative form

$$\rho \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho}) - \nabla_x \cdot \boldsymbol{U} = 0, \tag{3.32a}$$

$$\rho \frac{\mathrm{d}\,\boldsymbol{U}}{\mathrm{d}t} + \boldsymbol{\nabla}_x P = \boldsymbol{0}, \tag{3.32b}$$

$$\rho \frac{\mathrm{d}\,E}{\mathrm{d}t} + \nabla_x \cdot (P\boldsymbol{U}) = 0, \tag{3.32c}$$

This Lagrangian gas dynamics system of equations is written using the Eulerian coordinates $\boldsymbol{x}$. Thus, this Lagrangian form of the conservation laws are based on the moving configuration of the flow. Recalling the mass conservation (3.27b), we remark that the equation on the specific volume (3.32a) reduces to the equation of the material derivative of the Jacobian (3.12).

Another Lagrangian form of the gas dynamics equations comes from the development of the differential operators on the Lagrangian coordinates $\boldsymbol{X}$. In this frame, the system variables are expressed in terms of Lagrangian coordinates, and thus are based on the initial configuration of the flow. To obtain this local Lagrangian form of the conservation laws, we have to transform the divergence and gradient operators between Eulerian and Lagrangian configurations. Let us consider a vector $\boldsymbol{U} = \boldsymbol{U}(\boldsymbol{x}, t)$ and a scalar function $P = P(\boldsymbol{x}, t)$ which are expressed in terms of the spatial coordinates. Let us recall the definition of the divergence and gradient operators with respect to Eulerian coordinates

$$\nabla_x \cdot \boldsymbol{U} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}, \quad \boldsymbol{\nabla}_x P = (\frac{\partial P}{\partial x}, \frac{\partial P}{\partial y}, \frac{\partial P}{\partial z})^{\mathrm{t}}.$$

where (u, v, w) denotes the components of the velocity field in the orthogonal coordinate system (x, y, z). Here, subscript x emphasizes the fact that the divergence and gradient operators are expressed with respect to Eulerian coordinates. Now, if we consider $\boldsymbol{U}$ and P as functions expressed in terms of Lagrangian coordinates, then we define the divergence and gradient operators with respect to Lagrangian coordinates as follows

$$\nabla_X \cdot \boldsymbol{U} = \frac{\partial u}{\partial X} + \frac{\partial v}{\partial Y} + \frac{\partial w}{\partial Z}, \quad \boldsymbol{\nabla}_X P = (\frac{\partial P}{\partial X}, \frac{\partial P}{\partial Y}, \frac{\partial P}{\partial Z})^{\mathrm{t}}.$$

Divergence operator transformation. We investigate how the divergence operator transforms when pulling back to the Lagrangian configuration. To this end, we apply the divergence theorem twice as follows

$$\begin{aligned}
\int_\omega \nabla_x \cdot \boldsymbol{U}\, \mathrm{d}v &= \int_{\partial\omega} \boldsymbol{U} \cdot \boldsymbol{n}\, \mathrm{d}s \\
&= \int_{\partial\Omega} \boldsymbol{U} \cdot J\mathsf{F}^{-\mathrm{t}} \boldsymbol{N}\, \mathrm{d}S \quad \text{thanks to (3.17)} \\
&= \int_{\partial\Omega} J\mathsf{F}^{-1}\boldsymbol{U} \cdot \boldsymbol{N}\, \mathrm{d}S \\
&= \int_\Omega \nabla_X \cdot (J\mathsf{F}^{-1}\boldsymbol{U})\, \mathrm{d}V.
\end{aligned}$$

Coming back to the Lagrangian coordinates in the left-hand side of the previous equation, one gets

$$\int_{\omega} \nabla_x \cdot \boldsymbol{U} \, \mathrm{d}v = \int_{\Omega} J \nabla_x \cdot \boldsymbol{U} \, \mathrm{d}V.$$

The combination of the previous results leads to

$$\int_{\Omega} J \nabla_x \cdot \boldsymbol{U} \, \mathrm{d}V = \int_{\Omega} \nabla_X \cdot (J \mathsf{F}^{-1} \boldsymbol{U}) \, \mathrm{d}V.$$

Since this formula is valid for any arbitrary volume Ω, we deduce the final formula which expresses the transformation of the divergence operator into the Lagrangian configuration

$$\nabla_x \cdot \boldsymbol{U} = \frac{1}{J} \nabla_X \cdot (J \mathsf{F}^{-1} \boldsymbol{U}). \tag{3.33}$$

Now, developing the right-hand side of (3.33) through the use of identity (A.32a) yields

$$\nabla_x \cdot \boldsymbol{U} = \frac{1}{J} \nabla_X \cdot (J \mathsf{F}^{-\mathrm{t}}) \cdot \boldsymbol{U} + \mathsf{F}^{-1} : \nabla_X \boldsymbol{U}.$$

If $\boldsymbol{U}$ is an arbitrary constant vector, the previous identity yields

$$\nabla_X \cdot (J \mathsf{F}^{-\mathrm{t}}) = \boldsymbol{0},$$

which is nothing but the Piola identity (3.19).

Gradient operator transformation. We express the relation between the gradient operator expressed in terms of Eulerian coordinates and its counterpart expressed in terms of Lagrangian coordinates. To this end, we apply an identical procedure to the one used for the divergence

$$\begin{aligned}
\int_{\omega} \boldsymbol{\nabla}_x P \, \mathrm{d}v &= \int_{\partial\omega} P \boldsymbol{n} \, \mathrm{d}s, \\
&= \int_{\partial\Omega} P \, J \mathsf{F}^{-\mathrm{t}} \boldsymbol{N} \, \mathrm{d}S, \quad \text{thanks to (3.17)} \\
&= \int_{\Omega} \nabla_X \cdot (P \, J \mathsf{F}^{-\mathrm{t}}) \, \mathrm{d}V.
\end{aligned}$$

Recalling that $\mathrm{d}v = J \mathrm{d}V$, we finally obtain

$$\boldsymbol{\nabla}_x P = \frac{1}{J} \nabla_X \cdot (P \, J \mathsf{F}^{-\mathrm{t}}). \tag{3.34}$$

Developing the right-hand side, employing tensor identity (A.32d) and the Piola identity (3.19), we finally obtain the following variant of (3.34) which relates the gradient operator in both configuration as

$$\boldsymbol{\nabla}_x P = \mathsf{F}^{-\mathrm{t}} \boldsymbol{\nabla}_X P. \tag{3.35}$$

Note this last result could have been obtained employing the chain rule of composed derivatives.

Local Lagrangian gas dynamics system. At this time, using the definitions of the divergence (3.33) and gradient (3.34) operators expressed with respect to the Lagrangian coordinates, it is possible to write (3.32) in its total Lagrangian form

$$\frac{\mathrm{d}\,\mathsf{F}}{\mathrm{d}t} = \nabla_X \boldsymbol{U}, \tag{3.36a}$$

$$\rho^0 \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho}) - \nabla_X \cdot (J\mathsf{F}^{-1}\boldsymbol{U}) = 0, \tag{3.36b}$$

$$\rho^0 \frac{\mathrm{d}\,\boldsymbol{U}}{\mathrm{d}t} + \nabla_X \cdot (P\, J\mathsf{F}^{-\mathrm{t}}) = \mathbf{0}, \tag{3.36c}$$

$$\rho^0 \frac{\mathrm{d}\,E}{\mathrm{d}t} + \nabla_X \cdot (J\mathsf{F}^{-1}P\boldsymbol{U}) = 0, \tag{3.36d}$$

where $J = \det \mathsf{F}$ is the Jacobian of F.

Remark 1 *The gas dynamics system written under total Lagrangian form* (3.36) *is only weakly hyperbolic [50, 70, 83, 33, 31]. This loss of strict hyperbolicity is related to the supplementary conservation law that results from the time evolution of the deformation gradient tensor* (3.36a)*. We also note that* (3.36b) *derives from equation* (3.36a) *by means of equation* (3.11)*.*

Working with the initial position of fluid particles, the analysis is done on the fixed initial domain. However, the informations concerning the domain displacement and deformation are contained in the tensor F. The deformation gradient tensor is the geometry contribution to this gas dynamics system. It helps us to pass from the initial configuration of the flow to the actual one. To ensure the compatibility between these two configurations, the following relation must hold

$$\boldsymbol{\nabla}_X \times \mathsf{F} = \mathbf{0}, \tag{3.37}$$

where $\boldsymbol{\nabla}_X \times$ denotes the curl operator. This relation ensures that the deformation gradient tensor derived from a gradient.

In the next section, we present the two-dimensional Lagrangian hydrodynamics cell-centered scheme EUCCLHYD on unstructured grids which meets the compatibility GCL requirement. For a general polygonal grid, the discrete equations that govern the time rate of change of volume, momentum and total energy are obtained by means of a control volume formulation of the gas dynamics equations written using a cell-centered placement of the physical variables. The motion of the vertices of the grid is deduced from the trajectory equations, once the vertex velocity is defined. The general formalism presented relies on a sub-cell force-based discretization wherein the sub-cell force form is determined in such a manner that the resulting scheme satisfies an entropy inequality. Invoking the fundamental conservation principles of the momentum and the total energy leads to the construction of a node-centered solver that determines the nodal velocity. This particular scheme in its first-order version is the corner stone of our DG formulation presented in Chapter 4, our DG method naturally recovering the first-order EUCCLHYD scheme.

3.3 First-order cell-centered Lagrangian scheme: EUCCLHYD

This section aims at providing a finite volume discretization of the system of gas dynamics equations (3.32) using the concepts of sub-cell force and compatible discretization which have been initially introduced in [18] for Lagrangian hydrodynamics in the framework of staggered discretization.

3.3.1 Control volume form of the gas dynamics equations

Let $\mathcal{D}$ be an open set of $\mathbb{R}^2$, filled with an inviscid fluid and equipped with the orthonormal frame $(0, x, y)$ and the orthonormal basis $(\boldsymbol{e}_x, \boldsymbol{e}_y)$ which is naturally completed by the unit vector $\boldsymbol{e}_z = \boldsymbol{e}_x \times \boldsymbol{e}_y$. We want to develop a finite volume discretization of the gas dynamics equations (3.30). This formulation of the equations, which consists in following a control volume moving with the material velocity, writes

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\omega(t)} \mathrm{d}v - \int_{\partial\omega(t)} \boldsymbol{U} \cdot \boldsymbol{n}\, \mathrm{d}s = 0, \tag{3.38a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\omega(t)} \rho\boldsymbol{U}\, \mathrm{d}v + \int_{\partial\omega(t)} P\boldsymbol{n}\, \mathrm{d}s = \boldsymbol{0}, \tag{3.38b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\omega(t)} \rho E\, \mathrm{d}v + \int_{\partial\omega(t)} P\boldsymbol{U} \cdot \boldsymbol{n}\, \mathrm{d}s = 0. \tag{3.38c}$$

Here, $\omega(t)$ is the moving control volume and $\partial\omega(t)$ its boundary surface, whose unit outward normal is $\boldsymbol{n}$. Equations (3.38a)-(3.38c) express the conservation of volume, momentum and total energy. Obviously, the mass of the control volume $\omega(t)$ is conserved

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\omega(t)} \rho\, \mathrm{d}v = 0. \tag{3.39}$$

We note that first equation of system (3.38) expresses the time rate of change of the volume of the fluid and is often named Geometric Conservation Law (GCL). It is strongly linked to the trajectory equation

$$\frac{\mathrm{d}\,\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{U}(\boldsymbol{x}(t), t), \quad \boldsymbol{x}(0) = \boldsymbol{X}, \tag{3.40}$$

where $\boldsymbol{x} = \boldsymbol{x}(t)$ is the position vector of a material point at time $t > 0$, which was initially located at $\boldsymbol{X}$.

Let us remark that the integral form of the conservation laws (3.38) has been obtained utilizing the transport relation (3.29),*i.e.*, $\frac{\mathrm{d}}{\mathrm{d}t}\int_\omega \rho f\, \mathrm{d}v = \int_\omega \rho \frac{\mathrm{d}f}{\mathrm{d}t}\, \mathrm{d}v$, which is a consequence of the Reynolds transport relation [48] and mass conservation.

3.3.2 Compatible cell-centered discretization

We discretize the previous set of equations over a domain which is paved using a set of polygonal cells without gaps or overlaps. Let us introduce the notation and the assumptions that are necessary to develop our cell-centered discretization.

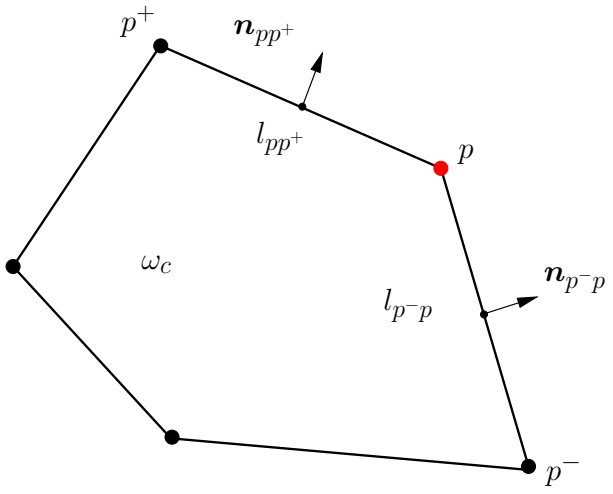


Figure 3.3: Notation for a polygonal cell.

Notation and assumptions

Each polygonal cell is assigned a unique index ω_c and is denoted by $\omega_c(t)$. A generic point (vertex) is labeled by the index p, its corresponding position vector is $\boldsymbol{x}_p$. For a cell ω_c, we introduce the set $\mathcal{P}(c)$ which is the counterclockwise ordered list of points of cell ω_c. Conversely, for a given point p, we introduce the set $\mathcal{C}(p)$ containing the cells that surround point p. Being given $p \in \mathcal{P}(c)$, p^- and p^+ are the previous and next points with respect to p in the ordered list of vertices of cell ω_c. The length and the unit outward normal related to the edge $[p, p^+]$ are l_{pp^+} and $\boldsymbol{n}_{pp^+}$, refer to Fig. 3.3. The control volume formulation (3.38) applied to the polygonal cell $\omega_c(t)$ leads to

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho_c}) - \int_{\partial\omega_c(t)} \boldsymbol{U} \cdot \boldsymbol{n} \, \mathrm{d}s = 0, \tag{3.41a}$$

$$m_c \frac{\mathrm{d}\,\boldsymbol{U}_c}{\mathrm{d}t} + \int_{\partial\omega_c(t)} P\boldsymbol{n} \, \mathrm{d}s = \boldsymbol{0}, \tag{3.41b}$$

$$m_c \frac{\mathrm{d}\,E_c}{\mathrm{d}t} + \int_{\partial\omega_c(t)} P\boldsymbol{U} \cdot \boldsymbol{n} \, \mathrm{d}s = 0. \tag{3.41c}$$

Here, $\partial\omega_c(t)$ is the boundary of the cell $\omega_c(t)$ and m_c denotes the mass of the polygonal cell, which is constant according to (3.39). For a flow variable ϕ, we have defined its mass averaged value over the cell $\omega_c(t)$ as

$$\phi_c = \frac{1}{m_c} \int_{\omega_c(t)} \rho\phi \, \mathrm{d}v.$$

We notice that the first equation corresponds to the GCL since $m_c = \rho_c v_c$ where v_c is the volume of cell ω_c. We have obtained a set of semi-discrete evolution equations for the primary variables $(\frac{1}{\rho_c}, \boldsymbol{U}_c, E_c)$ whose thermodynamic closure is given by the equation of state

$$P_c = P(\rho_c, \varepsilon_c), \quad \text{where} \quad \varepsilon_c = E_c - \frac{1}{2}\boldsymbol{U}_c^2. \tag{3.42}$$

The motion of the grid is ruled by the semi-discrete trajectory equation written at each point

$$\frac{\mathrm{d}\,\boldsymbol{x}_p}{\mathrm{d}t} = \boldsymbol{U}_p(\boldsymbol{x}_p(t), t), \quad \boldsymbol{x}_p(0) = \boldsymbol{X}_p. \tag{3.43}$$

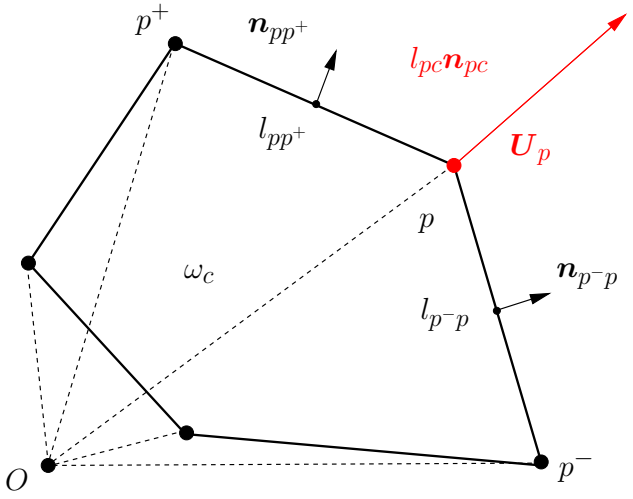


Figure 3.4: Triangulation of the polygonal cell ω_c.

To achieve the discretization, it remains not only to compute the numerical fluxes related to volume, momentum and total energy but also to determine the nodal velocity to compute the grid motion.

Geometric Conservation Law compatibility

Here, the expression GCL compatibility means that we are deriving a discrete divergence operator for the volume equation (3.38a) by requiring consistency of the divergence of the velocity field with the time rate of change of volume of a cell [32]. Noticing that $m_c = \rho_c v_c$, equation (3.41a) rewrites

$$\frac{\mathrm{d}\, v_c}{\mathrm{d}t} = \int_{\partial\omega_c(t)} \boldsymbol{U} \cdot \boldsymbol{n} \, \mathrm{d}s. \tag{3.44}$$

This equations states that the time rate of change of the cell volume v_c must be equal to the volume swept by the element boundary during its displacement with the flow velocity. Assuming that the volume $v_c(t)$ in the left-hand side can be computed exactly, this amounts to requiring the exact computation of the volume flux in the right-hand side also. In this manner, we obtain a compatible discretization of the volume flux. Using the triangulation displayed in Fig. 3.4, the polygonal cell volume writes

$$v_c(t) = \frac{1}{2} \sum_{p \in \mathcal{P}(c)} \left[\boldsymbol{x}_p(t) \times \boldsymbol{x}_{p^+}(t) \right] \cdot \boldsymbol{e}_z. \tag{3.45}$$

We remark that the volume is expressed as a function of the position vectors of its vertices. Using the chain rule derivative, time differentiation of equation (3.45) yields

$$\frac{\mathrm{d}\, v_c}{\mathrm{d}t} = \sum_{p \in \mathcal{P}(c)} \boldsymbol{\nabla}_{\boldsymbol{x}_p} v_c \cdot \frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{x}_p,$$

where $\boldsymbol{\nabla}_{\boldsymbol{x}_p} v_c$ is the gradient of the cell volume according to the position vector $\boldsymbol{x}_p$. This gradient is computed directly through the use of (3.45)

$$\begin{aligned} \boldsymbol{\nabla}_{\boldsymbol{x}_p} v_c =& \frac{1}{2} \boldsymbol{\nabla}_{\boldsymbol{x}_p} \left[(\boldsymbol{x}_{p^+} - \boldsymbol{x}_{p^-}) \times \boldsymbol{e}_z \cdot \boldsymbol{x}_p \right] \\ =& \frac{1}{2} (\boldsymbol{x}_{p^+} - \boldsymbol{x}_{p^-}) \times \boldsymbol{e}_z. \end{aligned}$$

We define the normal vector at corner pc by setting

$$l_{pc}\boldsymbol{n}_{pc} = \boldsymbol{\nabla}_{\boldsymbol{x}_p} v_c = \frac{1}{2}(\boldsymbol{x}_{p^+} - \boldsymbol{x}_{p^-}) \times \boldsymbol{e}_z, \tag{3.46}$$

where the length, l_{pc}, is determined knowing that $\boldsymbol{n}_{pc}^2 = 1$. We notice that this corner vector can be expressed in terms of the two outward normals impinging at node p as

$$l_{pc}\boldsymbol{n}_{pc} = \frac{1}{2}(l_{p^-p}\boldsymbol{n}_{p^-p} + l_{pp^+}\boldsymbol{n}_{pp^+}), \tag{3.47}$$

where l_{p^-p} and $\boldsymbol{n}_{p^-p}$ are the length and the unit outward normal related to the edge $[p^-, p]$ whereas l_{pp^+} and $\boldsymbol{n}_{pp^+}$ are the length and the unit outward normal related to the edge $[p, p^+]$, refer to Fig. 3.3. It turns out that the corner vector $l_{pc}\boldsymbol{n}_{pc}$ is the fundamental geometric object that allows to define uniquely the time rate of change of cell volume as

$$\frac{\mathrm{d}\, v_c}{\mathrm{d}t} = \sum_{p \in \mathcal{P}(c)} l_{pc}\boldsymbol{n}_{pc} \cdot \boldsymbol{U}_p. \tag{3.48}$$

Here, we have used the trajectory equation (3.43), i.e. $\dfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_p = \boldsymbol{U}_p$. The corner vector $l_{pc}\boldsymbol{n}_{pc}$ satisfies the fundamental geometrical identity

$$\sum_{p \in \mathcal{P}(c)} l_{pc}\boldsymbol{n}_{pc} = \boldsymbol{0}, \tag{3.49}$$

which is equivalent, using (3.47), to the well known result that the summation of the outward normals to a closed polygonal contour is equal to zero. This also involves that the volume of a Lagrangian cell in Cartesian geometry moving in a uniform flow does not change. We claim that with this purely geometric derivation we have completely defined the volume flux. Moreover, this definition is by construction compatible with mesh motion. We also remark that this result can be used to derive the discrete divergence operator, $\mathbb{DIV}_c$ over cell ω_c as

$$\mathbb{DIV}_c(\boldsymbol{U}) = \frac{1}{v_c}\int_{\partial\omega_c(t)} \boldsymbol{U} \cdot \boldsymbol{n}\, \mathrm{d}s.$$

Now, combining (3.44) and (3.48) we get

$$\mathbb{DIV}_c(\boldsymbol{U}) = \frac{1}{v_c}\frac{\mathrm{d}}{\mathrm{d}t}v_c = \frac{1}{v_c}\sum_{p \in \mathcal{P}(c)} l_{pc}\boldsymbol{n}_{pc} \cdot \boldsymbol{U}_p. \tag{3.50}$$

Let us remark that property (3.49) shows that the null space of the discrete divergence operator is spanned by the set of constant vectors. We recover the compatible discretization of the divergence operator currently used in the derivation of staggered grid Lagrangian schemes for compressible hydrodynamics [14, 106, 18].

Comment 3 *We want to mention that the above discrete divergence operator could have been derived utilizing an alternative approach which relies on the approximation of the surface integral,* $\int_{\partial\omega_c(t)} \boldsymbol{U} \cdot \boldsymbol{n}\, \mathrm{d}s$*, wherein a piecewise linear distribution of the velocity over each edge is assumed; this piecewise linear distribution being determined from the knowledge of the nodal velocity. Let us notice that this approach has been introduced a long time ago in [86] to compute the difference approximation for partial derivatives over an arbitrary region.*

Using the same approach, it is straightforward to define the discrete gradient operator, $\mathbb{GRAD}_c$, over cell ω_c. Let Q be a scalar valued function over cell ω_c, we define it as

$$\mathbb{GRAD}_c(Q) = \frac{1}{v_c} \int_{\partial\omega_c(t)} Q\boldsymbol{n} \, \mathrm{d}s.$$

The corresponding practical formula writes

$$\mathbb{GRAD}_c(Q) = \frac{1}{v_c} \sum_{p\in\mathcal{P}(c)} Q_p l_{pc} \boldsymbol{n}_{pc}. \tag{3.51}$$

where Q_p is the value taken by Q at the node p. We claim that the above formula is exact for linear functions. To demonstrate this result, let us introduce the useful geometrical identity

$$\sum_{p\in\mathcal{P}(c)} l_{pc}(\boldsymbol{n}_{pc} \otimes \boldsymbol{x}_p) = v_c \mathsf{I_d}. \tag{3.52}$$

This result can be demonstrated directly using simple algebra [77] or applying the divergence theorem as follows [48]

$$v_c \mathsf{I_d} = \int_{\omega_c} \nabla \boldsymbol{x} \, \mathrm{d}v = \int_{\partial\omega_c} \boldsymbol{x} \otimes \boldsymbol{n} \, \mathrm{d}s.$$

Bearing this in mind, we can demonstrate that (3.51) is exact for linear functions. To this end, let us consider the linear function, $Q = A + \boldsymbol{B} \cdot \boldsymbol{x}$, where A and $\boldsymbol{B}$ are a constant scalar and a constant vector. Obviously the nodal value of Q reads $Q_p = A + \boldsymbol{B} \cdot \boldsymbol{x}_p$. Substituting this expression into (3.51) leads to

$$\begin{aligned}
v_c \mathbb{GRAD}_c(Q) &= \sum_{p\in\mathcal{P}(c)} (A + \boldsymbol{B} \cdot \boldsymbol{x}_p) l_{pc} \boldsymbol{n}_{pc} \\
&= A \sum_{p\in\mathcal{P}(c)} l_{pc} \boldsymbol{n}_{pc} + \sum_{p\in\mathcal{P}(c)} l_{pc}(\boldsymbol{n}_{pc} \otimes \boldsymbol{x}_p)\boldsymbol{B}, \quad \text{recalling that } (\boldsymbol{u}\otimes\boldsymbol{v})\boldsymbol{w} = \boldsymbol{u}(\boldsymbol{v}\cdot\boldsymbol{w}) \\
&= v_c \mathsf{I_d} \boldsymbol{B}, \quad \text{thanks to (3.49) and (3.52)} \\
&= v_c \boldsymbol{B}.
\end{aligned}$$

Compatible discretization of a vector identity

In this paragraph, in view of the subsequent spatial discretization, we shall investigate the compatible discretization of the following vector identity

$$\nabla \cdot (Q\boldsymbol{U}) = \boldsymbol{U} \cdot \nabla Q + Q \nabla \cdot \boldsymbol{U}, \tag{3.53}$$

where Q is an arbitrary real valued function and $\boldsymbol{U}$ an arbitrary vector. Let us note that this identity, when applied to the real valued function, $Q = P$ and the velocity field $\boldsymbol{U}$, enables to derive the specific internal energy equation by subtracting the kinetic energy equation to the total energy equation. Knowing that the kinetic energy equation results from dot multiplying the momentum equation by the velocity vector.
We refer to Q_{pc} as the pointwise value of Q at the node p of cell ω_c. Let us consider the product $Q\boldsymbol{U}$, which is a vector, and let us define its pointwise value at node p by $(Q\boldsymbol{U})_p = Q_{pc}\boldsymbol{U}_p$ where $\boldsymbol{U}_p$ is the pointwise value of the vector $\boldsymbol{U}$ at node p. Employing the definition of the discrete divergence operator for a vector field (3.50) leads to

$$\mathbb{DIV}_c(Q\boldsymbol{U}) = \frac{1}{v_c} \sum_{p\in\mathcal{P}(c)} l_{pc}(Q_{pc}\boldsymbol{U}_p) \cdot \boldsymbol{n}_{pc}.$$

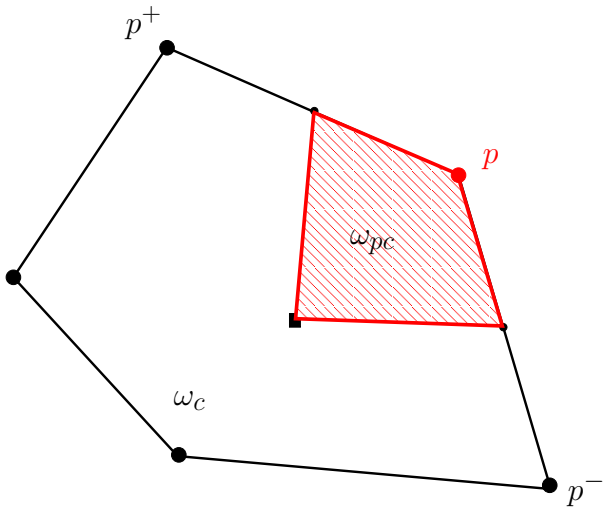


Figure 3.5: Sub-cell ω_{pc} related to polygonal cell ω_c and point p.

Now, combining the above formula, the discrete divergence operator for vector fields (3.50) and the discrete gradient operator for scalar fields (3.51) yields

$$v_c \left[\mathbb{DIV}_c(Q\boldsymbol{U}) - \boldsymbol{U}_c \cdot \mathbb{GRAD}_c(Q) - Q_c \mathbb{DIV}_c(\boldsymbol{U})\right] = \sum_{p \in \mathcal{P}(c)} l_{pc} \left[(Q_{pc} - Q_c)(\boldsymbol{U}_p - \boldsymbol{U}_c)\right] \cdot \boldsymbol{n}_{pc}. \quad (3.54)$$

Here, Q_c and $\boldsymbol{U}_c$ denote the mean values of Q and $\boldsymbol{U}$ over cell ω_c. This result has been obtained using the geometric identity (3.49). Formula (3.54) consists of the discrete counterpart of the identity (3.53). The right-hand side can be viewed as a residual term, which cancels only for uniform scalar and vector fields.

3.3.3 Sub-cell force-based discretization

It remains to discretize momentum and total energy equations. To this end, we utilize the fundamental concept of sub-cell force initially introduced in [18] in the context of staggered discretization. Let us introduce some notations. Being given a polygonal cell, ω_c, for each vertex $p \in \mathcal{P}(c)$, we define the sub-cell ω_{pc} by connecting the centroid of ω_c to the midpoints of edges $[p^-, p]$ and $[p, p^+]$ impinging at node p, refer to Fig. 3.5. In two dimensions the sub-cell, as just defined, is always a quadrilateral regardless of the type of cells that compose the underlying grid. Using the sub-cell definition, cell ω_c and its boundary $\partial\omega_c$ can be decomposed as

$$\omega_c = \bigcup_{p \in \mathcal{P}(c)} \omega_{pc}, \quad \partial\omega_c = \bigcup_{p \in \mathcal{P}(c)} \partial\omega_{pc} \cap \partial\omega_c. \quad (3.55)$$

This decomposition allows to rewrite the momentum flux as a summation of contributions coming from each sub-cell boundary

$$\int_{\partial\omega_c} P\boldsymbol{n} \, \mathrm{d}s = \sum_{p \in \mathcal{P}(c)} \int_{\partial\omega_{pc} \cap \partial\omega_c} P\boldsymbol{n} \, \mathrm{d}s.$$

Hence, we define the sub-cell force related to cell ω_c and point p

$$\boldsymbol{F}_{pc} = \int_{\partial\omega_{pc} \cap \partial\omega_c} P\boldsymbol{n} \, \mathrm{d}s. \quad (3.56)$$

This definition enables us to rewrite momentum equation (3.41b) as

$$m_c \frac{\mathrm{d}\,\boldsymbol{U}_c}{\mathrm{d}t} + \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{pc} = \boldsymbol{0},$$

which is the Newton law applied to a particle of mass m_c moving with velocity $\boldsymbol{U}_c$. We also use the sub-cell-based decomposition to write the total energy flux

$$\int_{\partial\omega_c} P\boldsymbol{U}\,.\,\boldsymbol{n}\,\mathrm{d}s = \sum_{p\in\mathcal{P}(c)} \int_{\partial\omega_{pc}\cap\partial\omega_c} P\boldsymbol{U}\,.\,\boldsymbol{n}\,\mathrm{d}s.$$

The contribution of the sub-cell, ω_{pc} to the total energy flux is expressed in terms of sub-cell force, $\boldsymbol{F}_{pc}$, by means of the approximation

$$\begin{aligned}\int_{\partial\omega_{pc}\cap\partial\omega_c} P\boldsymbol{U}\,.\,\boldsymbol{n}\,\mathrm{d}s &= \left(\int_{\partial\omega_{pc}\cap\partial\omega_c} P\boldsymbol{n}\,\mathrm{d}s\right).\boldsymbol{U}_p \\ &= \boldsymbol{F}_{pc}\,.\,\boldsymbol{U}_p, \quad \text{using sub-cell force definition (3.56).}\end{aligned}$$

Here, we have made the assumption that the velocity field over the outer sub-cell boundary, $\partial\omega_{pc}\cap\partial\omega_c$ is constant and equal to the nodal velocity, $\boldsymbol{U}_p$. Finally, total energy equation (3.41c) rewrites

$$m_c \frac{\mathrm{d}}{\mathrm{d}t} E_c + \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{pc}\,.\,\boldsymbol{U}_p = 0.$$

This leads to the mechanical interpretation that the time variation of total energy results from the summation over the sub-cell of the rate of work done by sub-cell force $\boldsymbol{F}_{pc}$ applied on each vertex p. Combining the previous results, system (3.41) becomes

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho_c}\right) - \sum_{p\in\mathcal{P}(c)} l_{pc}\boldsymbol{n}_{pc}\,.\,\boldsymbol{U}_p = 0, \tag{3.57a}$$

$$m_c \frac{\mathrm{d}\,\boldsymbol{U}_c}{\mathrm{d}t} + \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{pc} = \boldsymbol{0}, \tag{3.57b}$$

$$m_c \frac{\mathrm{d}\,E_c}{\mathrm{d}t} + \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{pc}\,.\,\boldsymbol{U}_p = 0. \tag{3.57c}$$

The cell-centered discrete unknowns $(\frac{1}{\rho_c}, \boldsymbol{U}_c, E_c)$ satisfy a system of semi-discrete evolution equations wherein the numerical fluxes are expressed as functions of the nodal velocity, $\boldsymbol{U}_p$, and the sub-cell force $\boldsymbol{F}_{pc}$. Let us recall that thermodynamic closure is given by the equation of state (3.42) and the grid motion is governed by the semi-discrete trajectory equation (3.43).
We conclude this section by giving an interpretation of the above sub-cell force-based discretization in terms of the discrete divergence and gradient operators previously introduced. Applying the definitions (3.50), (3.51) of the discrete divergence and gradient operators for vectors and scalars to the system (3.41) directly yields

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho_c}\right) - v_c \mathbb{DIV}_c(\boldsymbol{U}) = 0, \tag{3.58a}$$

$$m_c \frac{\mathrm{d}\,\boldsymbol{U}_c}{\mathrm{d}t} + v_c \mathbb{GRAD}_c(P) = \boldsymbol{0}, \tag{3.58b}$$

$$m_c \frac{\mathrm{d}\,E_c}{\mathrm{d}t} + v_c \mathbb{DIV}_c(P\boldsymbol{U}) = 0. \tag{3.58c}$$

Thanks to (3.51), the discrete gradient of the pressure reads

$$\mathbb{GRAD}_c(P) = \frac{1}{v_c} \sum_{p \in \mathcal{P}(c)} l_{pc} P_{pc} \boldsymbol{n}_{pc},$$

where P_{pc} is the pointwise value of P at point p of cell ω_c. Bearing this in mind, the comparison between (3.58b) and (3.57b) shows that the sub-cell force, $\boldsymbol{F}_{pc}$, and the pointwise pressure value, P_{pc}, are linked by

$$\boldsymbol{F}_{pc} = l_{pc} P_{pc} \boldsymbol{n}_{pc}. \tag{3.59}$$

To complete the discretization, it remains to compute the nodal velocity and construct an approximation of the sub-cell force. This goal will be achieved by investigating the properties of the scheme regarding its thermodynamic consistency and its conservation for momentum and total energy.

3.3.4 Thermodynamic consistency of the semi-discrete scheme

We derive a general form of the sub-cell force requiring that the semi-discrete scheme (3.57) satisfies a semi-discrete entropy inequality within cell ω_c

$$m_c \theta_c \frac{\mathrm{d}\,\eta_c}{\mathrm{d}t} \geq 0, \tag{3.60}$$

where $\theta_c > 0$ denotes the averaged temperature and η_c the specific entropy within cell ω_c. This semi-discrete entropy inequality by mimicking its continuous counterpart, will ensure that kinetic energy will be dissipated into internal energy through irreversible processes such as shock waves. We compute the time rate of change of η_c utilizing the Gibbs fundamental relation which writes

$$m_c \theta_c \frac{\mathrm{d}\,\eta_c}{\mathrm{d}t} = m_c \left[\frac{\mathrm{d}\,\varepsilon_c}{\mathrm{d}t} + P_c \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho_c}) \right]. \tag{3.61}$$

Now, it remains to express the right hand-side if the above equation by means of the semi-discrete equations of system (3.57). First, we compute the semi-discrete version of the time rate of change of specific internal energy within cell ω_c. Recalling that the specific internal energy within cell ω_c is given by $\varepsilon_c = E_c - \frac{1}{2}\boldsymbol{U}_c^2$, we get

$$m_c \frac{\mathrm{d}\,\varepsilon_c}{\mathrm{d}t} = m_c \frac{\mathrm{d}\,E_c}{\mathrm{d}t} - m_c \boldsymbol{U}_c \cdot \frac{\mathrm{d}\,\boldsymbol{U}_c}{\mathrm{d}t}.$$

Bearing this in mind, we dot multiply the momentum equation (3.58b) by $\boldsymbol{U}_c$ and subtract it to the total energy equation (3.58c) to obtain

$$m_c \frac{\mathrm{d}\,\varepsilon_c}{\mathrm{d}t} + v_c \left[\mathbb{DIV}_c(P\boldsymbol{U}) - \boldsymbol{U}_c \cdot \mathbb{GRAD}_c(P) \right] = 0. \tag{3.62}$$

Then, we compute the pressure rate of work by multiplying (3.58a) by P_c

$$m_c P_c \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho_c}) - v_c P_c \mathbb{DIV}_c(\boldsymbol{U}) = 0. \tag{3.63}$$

Summing equations (3.62) and (3.63), we get

$$m_c [\frac{\mathrm{d}\,\varepsilon_c}{\mathrm{d}t} + P_c \frac{d}{dt}(\frac{1}{\rho_c})] + v_c \left[\mathbb{DIV}_c(P\boldsymbol{U}) - \boldsymbol{U}_c \cdot \mathbb{GRAD}_c(P) - P_c \mathbb{DIV}_c(\boldsymbol{U}) \right] = 0.$$

Finally, using the Gibbs relation, the above equation rewrites

$$m_c\theta_c\frac{\mathrm{d}\,\eta_c}{\mathrm{d}t} = -v_c\left[\mathbb{DIV}_c(P\boldsymbol{U}) - \boldsymbol{U}_c\,\boldsymbol{.}\,\mathbb{GRAD}_c(P) - P_c\mathbb{DIV}_c(\boldsymbol{U})\right].$$

Expressing the right hand-side by means of the vectorial identity (3.54) leads to

$$m_c\theta_c\frac{\mathrm{d}\,\eta_c}{\mathrm{d}t} = -\sum_{p\in\mathcal{P}(c)} l_{pc}\left[(P_{pc}-P_c)(\boldsymbol{U}_p-\boldsymbol{U}_c)\right]\boldsymbol{.}\,\boldsymbol{n}_{pc}. \tag{3.64}$$

Utilizing the definition (3.59), the rate of entropy production can also be expressed in terms of the sub-cell force, $\boldsymbol{F}_{pc}$, as follows

$$m_c\theta_c\frac{\mathrm{d}\,\eta_c}{\mathrm{d}t} = -\sum_{p\in\mathcal{P}(c)} (\boldsymbol{F}_{pc} - l_{pc}P_c\boldsymbol{n}_{pc})\,\boldsymbol{.}\,(\boldsymbol{U}_p-\boldsymbol{U}_c). \tag{3.65}$$

To be consistent with the second law of thermodynamics, the rate of entropy production of the semi-discrete scheme has to be non-negative, a sufficient condition to obtain this consists in defining the sub-cell force so that the residual term in the right-hand side of (3.65) remains always non-negative. To this end, we set

$$\boldsymbol{F}_{pc} = l_{pc}P_c\boldsymbol{n}_{pc} - \mathsf{M}_{pc}(\boldsymbol{U}_p-\boldsymbol{U}_c), \tag{3.66}$$

where M_{pc} is a sub-cell-based 2×2 matrix which must satisfy the following properties

- M_{pc} is positive semi-definite, i.e. $\mathsf{M}_{pc}\boldsymbol{U}\,\boldsymbol{.}\,\boldsymbol{U}\geq 0, \quad \forall\,\boldsymbol{U}\in\mathbb{R}^2$.
- M_{pc} has a physical dimension equal to a length times a mass density times a velocity, i.e. $[\mathsf{M}_{pc}] = L\,\rho\,U$.

Substituting (3.66) into (3.65) leads to

$$m_c\theta_c\frac{\mathrm{d}\,\eta_c}{\mathrm{d}t} = \sum_{p\in\mathcal{P}(c)} \mathsf{M}_{pc}\,(\boldsymbol{U}_p-\boldsymbol{U}_c)\,\boldsymbol{.}\,(\boldsymbol{U}_p-\boldsymbol{U}_c)\geq 0. \tag{3.67}$$

We remark that the numerical dissipation rate within cell ω_c is directly governed by the positive semi-definite sub-cell matrix M_{pc} and the velocity jump between the nodal and the cell-centered velocity, $\Delta\boldsymbol{U}_{pc} = \boldsymbol{U}_p - \boldsymbol{U}_c$.

3.3.5 Summary of the semi-discrete scheme

Having determined the generic sub-cell force form consistent with the entropy inequality inequality (3.60), we present a summary of the semi-discrete equations that govern the evolution of the primary variables $(\frac{1}{\rho_c}, \boldsymbol{U}_c, E_c)$

$$\begin{aligned}
&m_c\frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho_c}) - \sum_{p\in\mathcal{P}(c)} l_{pc}\boldsymbol{n}_{pc}\,\boldsymbol{.}\,\boldsymbol{U}_p = 0,\\
&m_c\frac{\mathrm{d}\,\boldsymbol{U}_c}{\mathrm{d}t} + \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{pc} = \boldsymbol{0},\\
&m_c\frac{\mathrm{d}\,E_c}{\mathrm{d}t} + \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{pc}\,\boldsymbol{.}\,\boldsymbol{U}_p = 0.
\end{aligned}$$

Here, the sub-cell force reads

$$\boldsymbol{F}_{pc} = l_{pc} P_c \boldsymbol{n}_{pc} - \mathsf{M}_{pc}(\boldsymbol{U}_p - \boldsymbol{U}_c),$$

where M_{pc} is a positive semi-definite matrix. The thermodynamical closure of the above system is ensured by means of the following equation of state

$$P_c = P(\rho_c, \varepsilon_c), \quad \text{where} \quad \epsilon_c = E_c - \frac{1}{2}\boldsymbol{U}_c^2.$$

We have derived a compatible semi-discrete cell-centered Lagrangian scheme by means of a sub-cell force formalism. This scheme satisfies a semi-discrete local dissipation inequality consistent with the second law of thermodynamics. To achieve the discretization it remains to compute the nodal velocity $\boldsymbol{U}_p$ and also to provide an expression for the sub-cell matrix M_{pc} which is the cornerstone of the scheme.

3.3.6 Node-centered solver for the grid velocity

Total energy and momentum conservation

We investigate the conservation of our cell-centered scheme regarding total energy and momentum. Total energy over the whole grid is defined as $\mathcal{E}(t) = \sum_c m_c E_c(t)$, its conservation involves

$$\frac{\mathrm{d}\,\mathcal{E}}{\mathrm{d}t} = -\int_{\partial\mathcal{D}} P\boldsymbol{U} \cdot \boldsymbol{n}\,\mathrm{d}s,$$

where the right-hand side expresses the time rate of pressure work on the boundary, $\partial\mathcal{D}$, of the domain, $\mathcal{D}$, occupied by the fluid. By definition of total energy, this last equation rewrites

$$\sum_c m_c \frac{\mathrm{d}\,E_c}{\mathrm{d}t} = -\int_{\partial\mathcal{D}} P\boldsymbol{U} \cdot \boldsymbol{n}\,\mathrm{d}s. \tag{3.68}$$

Before proceeding further, we discretize the right-hand side. To this end, let us introduce some specific notations, assuming that the boundary is a closed contour. Let p be a node located on the boundary $\partial\mathcal{D}$, we denote by p^- and p^+ the previous and next points on the boundary with respect to p in the counterclockwise ordered list of points located on $\partial\mathcal{D}$. The curvilinear boundary $\partial\mathcal{D}$ is discretized using the decomposition $\partial\mathcal{D} = \cup_p \partial\mathcal{D}_p$. Here, $\partial\mathcal{D}_p = [i^-, p] \cup [p, i^+]$, where $i^\pm$ is the midpoint of the segment $[p^\pm, p]$. Using this decomposition, boundary term contribution is discretized as

$$\begin{aligned}\int_{\partial\mathcal{D}} P\boldsymbol{U} \cdot \boldsymbol{n}\,\mathrm{d}s &= \sum_{p\in\partial\mathcal{D}} \int_{\partial\mathcal{D}_p} P\boldsymbol{U} \cdot \boldsymbol{n}\,\mathrm{d}s \\ &= \sum_{p\in\partial\mathcal{D}} \left(\int_{\partial\mathcal{D}_p} P\boldsymbol{n}\,\mathrm{d}s\right) \cdot \boldsymbol{U}_p.\end{aligned}$$

The term between parentheses in the right-hand side of the second line represents a corner force that acts from the exterior boundary onto boundary points. Then, it is natural to set

$$\boldsymbol{F}_p^\star = \int_{\partial\mathcal{D}_p} P\boldsymbol{n}\,\mathrm{d}s, \tag{3.69}$$

where $\boldsymbol{F}_p^\star$ is the boundary corner force acting onto point p. Combining the previous notations and substituting the specific total energy equation (3.57c) into (3.68) yields the balance of total energy over the entire domain

$$\sum_c \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{pc} \cdot \boldsymbol{U}_p = \sum_{p\in\partial\mathcal{D}} \boldsymbol{F}_p^\star \cdot \boldsymbol{U}_p.$$

Interchanging the order in the double sum in the left-hand side yields

$$\sum_p \left(\sum_{c \in \mathcal{C}(p)} \boldsymbol{F}_{pc} \right) . \boldsymbol{U}_p = \sum_{p \in \partial\mathcal{D}} \boldsymbol{F}_p^\star . \boldsymbol{U}_p,$$

where $\mathcal{C}(p)$ is the set of cells surrounding point p. Finally, left-hand side of the above equation is divided into two parts depending on the points location

$$\sum_{p \in \mathcal{D}^o} \left(\sum_{c \in \mathcal{C}(p)} \boldsymbol{F}_{pc} \right) . \boldsymbol{U}_p + \sum_{p \in \partial\mathcal{D}} \left(\sum_{c \in \mathcal{C}(p)} \boldsymbol{F}_{pc} \right) . \boldsymbol{U}_p = \sum_{p \in \partial\mathcal{D}} \boldsymbol{F}_p^\star . \boldsymbol{U}_p, \tag{3.70}$$

where $\mathcal{D}^o$ is the interior of the domain $\mathcal{D}$. **Knowing that the total energy balance** (3.70) **must hold whatever point velocity is, total energy conservation is ensured if and only if**

$$\forall\, p \in \mathcal{D}^o, \quad \sum_{c \in \mathcal{C}(p)} \boldsymbol{F}_{pc} = \boldsymbol{0}, \tag{3.71a}$$

$$\forall\, p \in \partial\mathcal{D}, \quad \sum_{c \in \mathcal{C}(p)} \boldsymbol{F}_{pc} = \boldsymbol{F}_p^\star. \tag{3.71b}$$

It remains to check that the previous conditions also lead to momentum conservation. Let $\mathcal{Q}$ denotes the total momentum over the entire domain, i.e. $\mathcal{Q} = \sum_c m_c \boldsymbol{U}_c$. We compute its time rate of change

$$\begin{aligned}
\frac{\mathrm{d}\,\mathcal{Q}}{\mathrm{d}t} &= \sum_c m_c \frac{\mathrm{d}\,\boldsymbol{U}_c}{\mathrm{d}t} \\
&= -\sum_c \sum_{p \in \mathcal{P}(c)} \boldsymbol{F}_{pc}, \quad \text{thanks to (3.57b)} \\
&= -\sum_{p \in \mathcal{D}^o} \sum_{c \in \mathcal{C}(p)} \boldsymbol{F}_{pc} - \sum_{p \in \partial\mathcal{D}} \sum_{c \in \mathcal{C}(p)} \boldsymbol{F}_{pc}, \quad \text{by interchanging the double sum} \\
&= -\sum_{p \in \partial\mathcal{D}} \boldsymbol{F}_p^\star, \quad \text{thanks to (3.71)} \\
&= -\sum_{p \in \partial\mathcal{D}} \int_{\partial\mathcal{D}_p} P\boldsymbol{n}\,\mathrm{d}s, \quad \text{thanks to (3.69).}
\end{aligned}$$

We conclude that up to the boundary terms contribution, momentum is conserved over the entire domain. Hence, conditions (3.71a) and (3.71b) turn out to ensure not only total energy but also momentum conservation. Moreover, as we shall show it in next section, they also provide a vectorial equation that enables us to determine the nodal velocity.

Comment 4 *The fundamental equation* (3.70) *can be interpreted as discrete variational formulation wherein the point velocity* $\boldsymbol{U}_p$ *is a test function.*

Generic formulation of the nodal solver

Remembering that the general sub-cell force form reads $\boldsymbol{F}_{pc} = l_{pc} P_c \boldsymbol{n}_{pc} - \mathsf{M}_{pc}(\boldsymbol{U}_p - \boldsymbol{U}_c)$, where M_{pc} is a 2×2 positive semi-definite matrix, and using the conservation condition (3.71), we are

now in position to write the 2×2 system that solves the nodal velocity $\boldsymbol{U}_p$. In the general case this system writes

$$\forall\, p \in \mathcal{D}^o, \quad \mathsf{M}_p \boldsymbol{U}_p = \sum_{c \in \mathcal{C}(p)} \left(l_{pc} P_c \boldsymbol{n}_{pc} + \mathsf{M}_{pc} \boldsymbol{U}_c \right), \tag{3.72a}$$

$$\forall\, p \in \partial\mathcal{D}, \quad \mathsf{M}_p \boldsymbol{U}_p = \sum_{c \in \mathcal{C}(p)} \left(l_{pc} P_c \boldsymbol{n}_{pc} + \mathsf{M}_{pc} \boldsymbol{U}_c \right) - \boldsymbol{F}_p^{\star}, \tag{3.72b}$$

where M_p denotes the 2×2 node-centered matrix defined as

$$\mathsf{M}_p = \sum_{c \in \mathcal{C}(p)} \mathsf{M}_{pc}. \tag{3.73}$$

We emphasize that we have divided the nodal velocity determination into two cases depending on the node location. As M_{pc} is positive semi-definite, M_p also shares the same property. To enforce the solvability of equation (3.72a) and equation (3.72b), we assume that the matrix M_{pc} is positive **definite**. This yields to a M_p matrix which is always invertible. Therefore, provided that the M_{pc} matrix is defined, the nodal velocity, $\boldsymbol{U}_p$ is always uniquely determined by inverting equations (3.72a) and (3.72b). We recall that M_{pc} has the physical dimension of a length times a density times a velocity. If M_{pc} does not depend on the nodal velocity, hence (3.72a) and (3.72b) are linear equations whose solutions are easily obtained.

Boundary conditions

In this section, we describe briefly boundary conditions implementation for the gas dynamics equations. In Lagrangian formalism, this task is quite simple as we have to consider only two types of natural boundary conditions. Namely, on the boundary of the domain, $\partial\mathcal{D}$, either the pressure or the normal component of the velocity is prescribed. Let us point out that we present a boundary conditions implementation which is fully compatible with the node-centered solver previously developed. To this end, let us consider a generic point p located on the boundary, we denote p^- and p^+ its previous and next neighbor with respect to p in the counterclockwise ordered list of boundary points. Without loss of generality we make the assumption that $\partial\mathcal{D}$ is a closed contour. The two outward normals to the edges located on the boundary and connected to point p are $l_p^- \boldsymbol{n}_p^-$ and $l_p^+ \boldsymbol{n}_p^+$, where $l_p^{\pm}$ denotes one half of the length of the segment $[p, p^{\pm}]$, refer to Fig. 3.6. These outward normals are linked to the corner vectors $l_{pc}\boldsymbol{n}_{pc}$ as follows

$$\sum_{c \in \mathcal{C}(p)} l_{pc} \boldsymbol{n}_{pc} = l_p^- \boldsymbol{n}_p^- + l_p^+ \boldsymbol{n}_p^+. \tag{3.74}$$

This result is due to the fact that the dual cell contour (red dashed line in Fig. 3.6) is closed . Now, let us investigate the two following cases.

Prescribed pressure. The boundary corner force acting onto point p defining by (3.69) writes

$$\boldsymbol{F}_p^{\star} = l_p^- \Pi_p^{-,\star} + l_p^+ \Pi_p^{+,\star},$$

where $\Pi_p^{\pm,\star}$ are the prescribed pressures on the two half-edges impinging at point p. Substituting this expression of the boundary corner force into (3.72b) leads to

$$\mathsf{M}_p \boldsymbol{U}_p = \sum_{c \in \mathcal{C}(p)} \left(l_{pc} P_c \boldsymbol{n}_{pc} + \mathsf{M}_{pc} \boldsymbol{U}_c \right) - \left(l_p^- \Pi_p^{-,\star} \boldsymbol{n}_p^{-,\star} + l_p^+ \Pi_p^{+,\star} \boldsymbol{n}_p^{+,\star} \right). \tag{3.75}$$

Due to (3.74), we note that this formula preserves uniform fluid flows. Namely, if $(P_c, \boldsymbol{U}_c) = (P^0, \boldsymbol{U}^0), \quad \forall\, c$, and $\Pi_p^{\pm,\star} = P^0 \boldsymbol{n}_p^{\pm}, \quad \forall\, p$, then $\boldsymbol{U}_p = \boldsymbol{U}^0$.

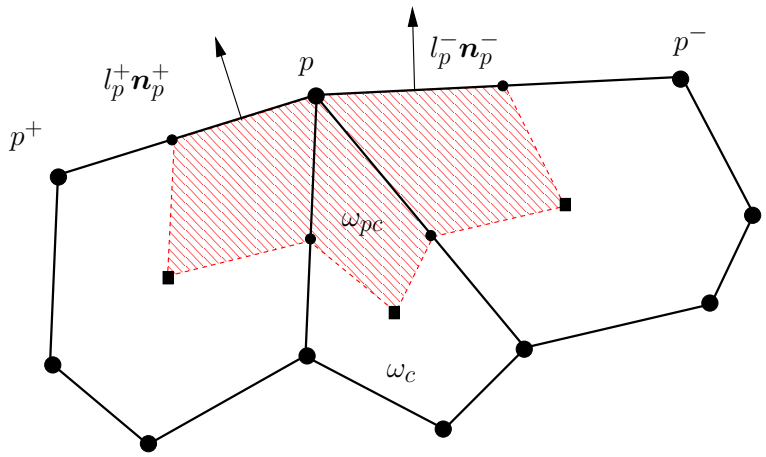


Figure 3.6: Fragment of a polygonal grid in the vicinity of point p located on the boundary $\partial\mathcal{D}$.

Prescribed normal velocity. Let $\mathcal{V}_p^{\pm,*}$ denotes the prescribed normal velocity on sides of point p, that is $(\boldsymbol{U} \cdot \boldsymbol{n}_p^{\pm}) = \mathcal{V}_p^{\pm,*}$. Let us note that the nodal velocity, $\boldsymbol{U}_p$, satisfies (3.72b)

$$\mathsf{M}_p \boldsymbol{U}_p = \sum_{c \in \mathcal{C}(p)} (l_{pc} P_c \boldsymbol{n}_{pc} + \mathsf{M}_{pc} \boldsymbol{U}_c) - \boldsymbol{F}_p^\star,$$

where $\boldsymbol{F}_p^\star$ is an unknown pressure force acting at node p. We point out that the nodal velocity is constrained by the normal velocity boundary conditions as follows

$$\left(l_p^- \boldsymbol{n}_p^- + l_p^+ \boldsymbol{n}_p^+\right) \cdot \boldsymbol{U}_p = l_p^- \mathcal{V}_p^{-,*} + l_p^+ \mathcal{V}_p^{+,*}. \tag{3.76}$$

It remains to determine the expression of the boundary force $\boldsymbol{F}_p^\star$. Recalling that we want to ensure total energy conservation (3.70) leads to propose the following expression for the boundary corner force, $\boldsymbol{F}_p^\star$,

$$\boldsymbol{F}_p^\star = (l_p^- \boldsymbol{n}_p^- + l_p^+ \boldsymbol{n}_p^+) \Pi_p^\star,$$

where $\Pi_p^\star$ stands for an averaged pressure acting onto point p. Substituting this expression into (3.72b) and making use of (3.76) leads to

$$\mathsf{M}_p \boldsymbol{U}_p + (l_p^- \boldsymbol{n}_p^- + l_p^+ \boldsymbol{n}_p^+) \Pi_p^\star = \sum_{c \in \mathcal{C}(p)} l_{pc} P_c \boldsymbol{n}_{pc} + \mathsf{M}_{pc} \boldsymbol{U}_c. \tag{3.77}$$

Thanks to the above equation, we express the nodal velocity in terms of the pressure $\Pi_p^\star$, then dot multiplying this resulting equation by the corner normal $\boldsymbol{n}_p = l_p^- \boldsymbol{n}_p^- + l_p^+ \boldsymbol{n}_p^+$, and using the boundary condition (3.76), we finally get the equation satisfied by the pressure, $\Pi_p^\star$

$$(\mathsf{M}_p^{-1} \boldsymbol{n}_p \cdot \boldsymbol{n}_p) \Pi_p^\star = -\left(l_p^- \mathcal{V}_p^{-,*} + l_p^+ \mathcal{V}_p^{+,*}\right) + \sum_{c \in \mathcal{C}(p)} \mathsf{M}_p^{-1} \left(l_{pc} P_c \boldsymbol{n}_{pc} + \mathsf{M}_{pc} \boldsymbol{U}_c\right) \cdot \boldsymbol{n}_p. \tag{3.78}$$

It is clear that this equation has always a unique solution provided that M_p^{-1} is positive definite. Computing $\Pi_p^\star$ by means of (3.78), it remains to obtain $\boldsymbol{U}_p$ by solving

$$\mathsf{M}_p \boldsymbol{U}_p = \sum_{c \in \mathcal{C}(p)} (l_{pc} P_c \boldsymbol{n}_{pc} + \mathsf{M}_{pc} \boldsymbol{U}_c) - \Pi_p^\star \boldsymbol{n}_p,$$

At this point, we can conclude that the sub-cell force formalism provides a general framework that allows to construct a family of compatible cell-centered schemes that share good physical

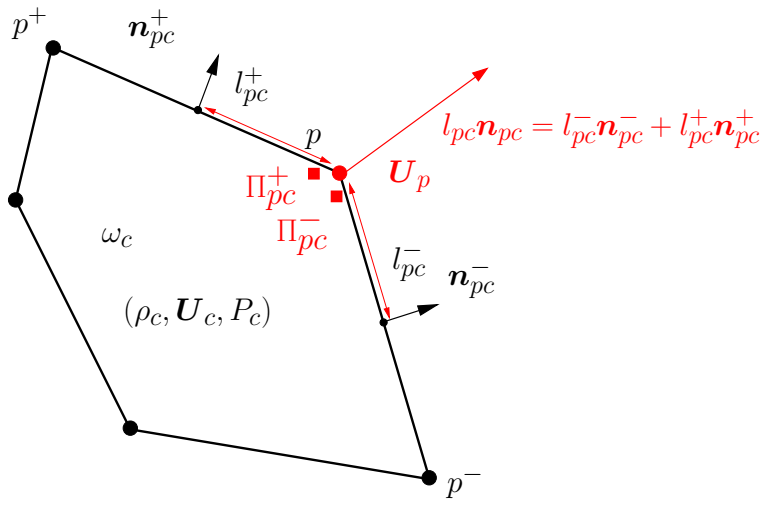


Figure 3.7: Notation related to the corner of a polygonal cell.

properties such as thermodynamic consistency and conservation. The numerical fluxes and the nodal velocity are computed in a compatible manner by means of a node-centered solver which enables us to consistently derive boundary conditions. The key point in designing these schemes lies in constructing the corner matrix M_{pc}. We shall investigate this latter point in the next paragraph, by presenting one example of such a construction and making the link with approximate Riemann solvers.

Definition of the corner matrix

The definition of the corner matrix is based on a methodology that has been developed initially in [78] and revisited in [74, 82]. Let us recall that the abstract formulation of the sub-cell force reads

$$\boldsymbol{F}_{pc} = l_{pc} P_c \boldsymbol{n}_{pc} - \mathsf{M}_{pc}(\boldsymbol{U}_p - \boldsymbol{U}_c),$$

where M_{pc} must be a 2×2 positive semi-definite matrix that remains to express. First, let us introduce Π_{pc}^- and Π_{pc}^+ the two nodal pressures located at node p within cell ω_c. Using this notation, the sub-cell force, $\boldsymbol{F}_{pc}$, writes

$$\boldsymbol{F}_{pc} = l_{pc}^- \Pi_{pc}^- \boldsymbol{n}_{pc}^- + l_{pc}^+ \Pi_{pc}^+ \boldsymbol{n}_{pc}^+,$$

where $l_{pc}^-\boldsymbol{n}_{pc}^-$ and $l_{pc}^+\boldsymbol{n}_{pc}^+$ are the outward normals to edges $[p^-, p]$, $[p, p^+]$. Let us note that $l_{pc}^\pm$ are one half of the length of the corresponding edges, refer to Fig. 3.7. These two nodal pressures are expressed as functions of the cell pressure and the jump between cell and node velocity by means of the two following Riemann invariants written along each half-edge normal directions

$$\begin{aligned} \Pi_{pc}^- - P_c &= -z_c(\boldsymbol{U}_p - \boldsymbol{U}_c) \cdot \boldsymbol{n}_{pc}^-, \\ \Pi_{pc}^+ - P_c &= -z_c(\boldsymbol{U}_p - \boldsymbol{U}_c) \cdot \boldsymbol{n}_{pc}^+, \end{aligned}$$

where $z_c = \rho_c a_c$ denotes the acoustic impedance, knowing that a_c is the thermodynamical sound speed computed from the equation of state of the material as

$$a_c^2 = \left(\frac{\partial P}{\partial \rho}\right)_\varepsilon + \frac{P}{\rho^2}\left(\frac{\partial P}{\partial \varepsilon}\right)_\rho.$$

Noticing that $l_{pc}^{-}\boldsymbol{n}_{pc}^{-} + l_{pc}^{+}\boldsymbol{n}_{pc}^{+} = l_{pc}\boldsymbol{n}_{pc}$ and combining the two above equations leads to the following expression of the sub-cell force

$$\boldsymbol{F}_{pc} = l_{pc}P_c\boldsymbol{n}_{pc} - z_c\left[l_{pc}^{-}(\boldsymbol{n}_{pc}^{-}\otimes\boldsymbol{n}_{pc}^{-}) + l_{pc}^{+}(\boldsymbol{n}_{pc}^{+}\otimes\boldsymbol{n}_{pc}^{+})\right](\boldsymbol{U}_p - \boldsymbol{U}_c).$$

From the comparison between this last expression and the generic form of the sub-cell force, we deduce that the corner matrix writes

$$\mathsf{M}_{pc} = z_c\left[l_{pc}^{-}(\boldsymbol{n}_{pc}^{-}\otimes\boldsymbol{n}_{pc}^{-}) + l_{pc}^{+}(\boldsymbol{n}_{pc}^{+}\otimes\boldsymbol{n}_{pc}^{+})\right]. \tag{3.79}$$

The corner matrix defined by (3.79) is symmetric definite positive and has been initially introduced in [75, 76, 77].

Now, we present a second-order extension of the scheme presented. Firstly, we describe the classical second-order MUSCL method. Then, we present an alternative procedure allowing the reconstruction of the slopes, using the nodal solvers defined previously.

3.3.7 Second-order MUSCL scheme

Pierre-Henri Maire and Jérome Breil presented in [79] a second-order extension of the EUCCLHYD scheme, using classical MUSCL-type approach. In this frame, the procedure consists in a least square procedure minimizing the jump between the extrapolated values of the linear reconstruction in the cell at the centroid of the cells surrounding the considered one, and the mean values in these cells [5]. After having briefly introduce this procedure, we present an alternative MUSCL method [20]. Instead of using the mean values of the neighboring cells we make use of $\boldsymbol{U}_p$ and $\boldsymbol{F}_{pc}$, defined previously, to evaluate the approximate gradients needed in the linear reconstruction.

Let recall the first-order discretization (3.57) of the Lagrangian integral gas dynamics equation (3.38) presented before. The mass averaged values of the volume, velocity and total energy respectively ρ_c, $\boldsymbol{U}_c$ and E_c at time t on the moving cell ω_c are governed by the following semi-discrete equations

$$\begin{aligned}
&m_c\frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho_c}) - \sum_{p\in\mathcal{P}(c)}\boldsymbol{U}_p\cdot l_{pc}\boldsymbol{n}_{pc} = 0,\\
&m_c\frac{\mathrm{d}\,\boldsymbol{U}_c}{\mathrm{d}t} + \sum_{p\in\mathcal{P}(c)}\boldsymbol{F}_{pc} = \boldsymbol{0},\\
&m_c\frac{\mathrm{d}\,E_c}{\mathrm{d}t} + \sum_{p\in\mathcal{P}(c)}\boldsymbol{U}_p\cdot\boldsymbol{F}_{pc} = 0,
\end{aligned}$$

where $\boldsymbol{U}_p$ are the node velocity and $\boldsymbol{F}_{pc}$ the nodal subcell forces.

Classical MUSCL method

Being given ϕ_c the mass averaged value of the function ϕ over the cell ω_c, we define its linear reconstruction as

$$\widetilde{\phi}_c(\boldsymbol{x}) = \phi_c + (\nabla_x\Phi)_c\cdot(\boldsymbol{x} - \boldsymbol{x}_c), \tag{3.80}$$

where $\boldsymbol{x}_c$ is the centroid of the cell ω_c and $(\boldsymbol{\nabla}_x\phi)_c$ represents the gradient of the solution. In the case of a vector valued function $\boldsymbol{\Phi}$, the linear reconstruction reads

$$\widetilde{\boldsymbol{\Phi}}_c(\boldsymbol{x}) = \boldsymbol{\Phi}_c + (\nabla_x\boldsymbol{\Phi})_c(\boldsymbol{x} - \boldsymbol{x}_c), \tag{3.81}$$

where $(\nabla_x\boldsymbol{\Phi})_c$ represents the gradient tensor of $\boldsymbol{\Phi}$.

The least square procedure compute $(\boldsymbol{\nabla}_x\phi)_c$ and $(\nabla_x\boldsymbol{\Phi})_c$ imposing the following conditions

$$\phi_d = \widetilde{\phi}_c(\boldsymbol{x}_d), \quad \forall\omega_d \in \mathcal{N}(c), \tag{3.82}$$

$$\boldsymbol{\Phi}_d = \widetilde{\boldsymbol{\Phi}}_c(\boldsymbol{x}_d), \quad \forall\omega_d \in \mathcal{N}(c), \tag{3.83}$$

where ϕ_d and $\boldsymbol{\Phi}_d$ are the mass averaged value of the function ϕ and $\boldsymbol{\Phi}$ over the cell ω_d, $\boldsymbol{x}_d$ is the centroid of this cell and $\mathcal{N}(c)$ is the set of the neighboring cells of ω_c. The neighborhood set $\mathcal{N}(c)$ of the cell ω_c could be defined in different manners. We decide to select the cells sharing a common face with ω_c. An alternative solution is to extend the stencil to the cells surrounding each nodes of $\mathcal{P}(c)$, the node set of ω_c.

Condition (3.82) or (3.83) is satisfied in a least squares sense by minimizing the cost function

$$L_c = \sum_{d\in\mathcal{N}(c)} \frac{1}{2}\left(\widetilde{\phi}_c(\boldsymbol{x}_d) - \phi_d\right)^2 = \sum_{d\in\mathcal{N}(c)} \frac{1}{2}\left[\phi_c - \phi_d + (\boldsymbol{\nabla}_x\phi)_c \cdot (\boldsymbol{x}_d - \boldsymbol{x}_c)\right]^2,$$

where ϕ_d is the mass averaged value of the function ϕ over the cell ω_d. And the minimum cost is reached for

$$\left(\sum_{d\in\mathcal{N}(c)} (\boldsymbol{x}_d - \boldsymbol{x}_c)\otimes(\boldsymbol{x}_d - \boldsymbol{x}_c)\right)(\boldsymbol{\nabla}_x\phi)_c = \sum_{d\in\mathcal{N}(c)} (\boldsymbol{x}_d - \boldsymbol{x}_c)(\phi_d - \phi_c). \tag{3.84}$$

The matrix $\sum_{d\in\mathcal{N}(c)}(\boldsymbol{x}_d-\boldsymbol{x}_c)\otimes(\boldsymbol{x}_d-\boldsymbol{x}_c)$ being symmetric positive definite, it is thus invertible. This formula gives us a definition of the approximate gradient $(\boldsymbol{\nabla}_x\phi)_c$. Applying this procedure to the pressure and on the two components of the velocity vector, one gets the piecewise linear reconstructions in the cells of the considered variables. In the second-order scheme, the nodal solvers $\boldsymbol{U}_p$ and $\boldsymbol{F}_{pc}$ are computed by means of the extrapolated values located at the node p of these linear approximations, the aim being to decrease the dissipation and to increase the accuracy of the scheme. This approach has the main feature to preserve linear fields regardless the shape of the cell. However, the treatment of the boundary conditions is not obvious and one must introduce ghost cells to take them into account properly. A more straightforward approach consists in using a Green-Gauss based approach to derive the gradients approximations, as it been described in [20].

Alternative MUSCL method

This time, in order to define $(\boldsymbol{\nabla}_x\psi)_c$ and $(\nabla_x\boldsymbol{\Psi})_c$, we use a similar approach to the one presented in Section 3.3.2. Considering the function ψ, we recall the discrete gradient operator $\mathbb{GRAD}_c(\psi)_c$ definition (3.51)

$$\mathbb{GRAD}_c(\psi) = \frac{1}{v_c}\sum_{p\in\mathcal{P}(c)} \psi_p\, l_{pc}\boldsymbol{n}_{pc}. \tag{3.85}$$

Here, ψ_p is a single value function derived from the numerical solvers. Using this definition to go to second-order, the stencil is reduced to the considered cell. The linear reconstruction is local. The information coming from the neighboring cells or the boundary conditions consideration as well are naturally contained in the nodal solvers. We are also able to derive a gradient definition of a vector valued function $\boldsymbol{\Psi}$

$$\mathbb{GRAD}_c(\boldsymbol{\Psi}) = \frac{1}{v_c} \sum_{p \in \mathcal{P}(c)} \boldsymbol{\Psi}_p \otimes l_{pc} \boldsymbol{n}_{pc}. \tag{3.86}$$

We make use of these gradient definitions in the slope reconstruction of variables. Thus, we identify the pressure gradient vector $(\boldsymbol{\nabla}_x P)_c$ to $\mathbb{GRAD}_c(P)$, and the velocity gradient matrix $(\nabla_x \boldsymbol{U})_c$ to $\mathbb{GRAD}_c(\boldsymbol{U})$

$$(\boldsymbol{\nabla}_x P)_c = \frac{1}{v_c} \sum_{p \in \mathcal{P}(c)} P_p \, l_{pc} \boldsymbol{n}_{pc}, \tag{3.87}$$

$$(\nabla_x \boldsymbol{U})_c = \frac{1}{v_c} \sum_{p \in \mathcal{P}(c)} \boldsymbol{U}_p \otimes l_{pc} \boldsymbol{n}_{pc}, \tag{3.88}$$

where $\boldsymbol{U}_p$ is the velocity of the moving node p, and P_p is the single node pressure computed with the use of the sub-cell forces. It remains to determine the nodal pressure P_p knowing $\boldsymbol{F}_{pc}$. Firstly, we recall the sub-cell force into a left and a right contributions

$$\boldsymbol{F}_{pc} = l^-_{pc} \Pi^-_{pc} \boldsymbol{n}^-_{pc} + l^+_{pc} \Pi^+_{pc} \boldsymbol{n}^+_{pc}, \tag{3.89}$$

where Π^-_{pc} and Π^+_{pc} are the two nodal pressure, refer to Figure 3.7. Equation (3.89) defines a 2×2 linear system on the nodal pressures which leads to following solution

$$\Pi^-_{pc} = \boldsymbol{F}_{pc} \cdot \frac{\boldsymbol{n}^-_{pc} - \cos\theta_{pc} \, \boldsymbol{n}^+_{pc}}{l^-_{pc} \, \sin^2 \theta_{pc}},$$

$$\Pi^+_{pc} = \boldsymbol{F}_{pc} \cdot \frac{\boldsymbol{n}^+_{pc} - \cos\theta_{pc} \, \boldsymbol{n}^-_{pc}}{l^+_{pc} \, \sin^2 \theta_{pc}},$$

where θ_{pc} denotes the measure of the angle between the two half-edges of sub-cell ω_{pc}, refer to Figure 3.8.

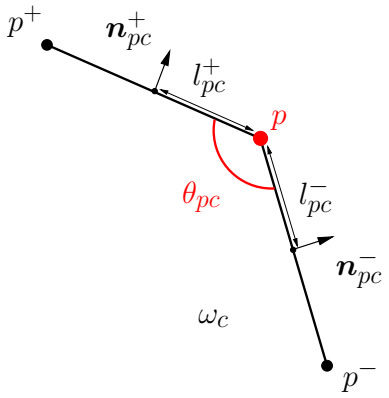


Figure 3.8: Corner angle of a polygonal cell.

We end up defining the single nodal pressure P_p as the average of all the sub-cell nodal pressures $\Pi^\pm_{pc}$ in the cell surrounding the node p

$$P_p = \frac{1}{\mathrm{N}_p} \sum_{c \in \mathcal{C}(p)} \frac{1}{2} (\Pi^-_{pc} + \Pi^+_{pc}).$$

where N_p is the number of cell surrounding the node p.

As previously, the linear reconstruction of the velocity and pressure are used in the construction of the nodal solvers. $\boldsymbol{U}_p$ and $\boldsymbol{F}_{pc}$ are computed by means of the extrapolated values located at the node p through these linear approximations. A convergence analysis has been performed on this particular method, in the smooth case of a Taylor-Green vortex. The results are displayed in Section 4.2.4 in Table 4.3, and compared with the ones obtained with our second-order DG scheme.

To reach a global second-order accuracy, we have to use a second-order time discretization. We could use for example, the classical second-order TVD Runge-Kutta discretization. An alternative choice is the second-order one-step time discretization, described by Maire in [74]. This time discretization derives from a prolongation of works initiated by Ben-Artzi and Falcovitz [6] concerning the Generalized Riemann Problem. Now, we briefly described the time step control to ensure the stability of the scheme. Finally, the robustness and the accuracy of these Finite Volume schemes have been assessed using several test cases taken from the literature. The interested reader may refer to the following papers [78, 79, 74, 73, 75, 76] presenting numerical results obtained with the Finite Volume EUCCLHYD scheme over a large number of test cases.

3.3.8 Time step control

For numerical applications, the time step is evaluated following two criteria. The first one is a standard CFL criterion [31]. The second one is more intuitive, but reveals very useful in practice: we limit the variation of the volume of cells over one time step.

CFL criterion. We introduce a CFL like criterion in order to ensure a positive entropy production in cell ω_c during the time step. At time t^n, for each cell ω_c we denote by λ_c^n the minimal value of the distance between two points of the cell. We define

$$\Delta t_e = C_e \min_c \frac{\lambda_c^n}{a_c^n},$$

where C_e is a strictly positive coefficient and a_c is the sound speed in the cell. The coefficient C_e is computed heuristically and we provide no rigorous analysis which allows such formula. However, extensive numerical experiments show that $C_e = 0.25$ is a value which provides stable numerical results in the second-order case. A rigorous derivation of this criterion can be obtained by computing the time step which ensures a positive entropy production in cell ω_c during the time interval $[t^n, t^{n+1}]$. Note that in [31], an optimization of the time step is performed to ensure a positive entropy dissipation at the discrete level.

Criterion on the volume variation. We estimate the volume of cell ω_c at $t = t^{n+1}$ using the Taylor expansion

$$v_c^{n+1} = v_c^n + \frac{\mathrm{d}}{\mathrm{d}t} v_c(t^n) \Delta t.$$

Here, the time derivative $\dfrac{\mathrm{d}}{\mathrm{d}t} v_c$ is computed by using (3.48) page 105. Let C_v be a positive coefficient, $C_v \in]0, 1[$. We look for Δt such that

$$\frac{|v_c^{n+1} - v_c^n|}{v_c^n} \leq C_v.$$

To this end, let us define

$$\Delta t_v = C_v \min_c \left\{ \frac{v_c^n}{|\frac{d}{dt} v_c(t^n)|} \right\}.$$

For numerical applications, we choose $C_v = 0.1$. Last, the estimation of the next time step Δt^{n+1} is given by

$$\Delta t^{n+1} = \min \left(\Delta t_e, \Delta t_v, C_m \Delta t^n \right), \tag{3.90}$$

where Δt^n is the current time step and C_m is a multiplicative coefficient which allows the time step to increase. We generally set $C_m = 1.01$.

In this section, we have recalled the cell-centered first and second-order EUCCLHYD schemes. The scheme was the starting point of the implementation of high-order DG schemes. One of the main objective was to recovers the EUCCLHYD scheme in the first-order case. In the next chapter, conserving the same notations, we present the general framework of the high-order DG discretization for the Lagrangian gas dynamics system of equations written using the Lagrangian coordinates. In the end, we emphasize that in its first-order version, our DG method naturally recovers the first-order EUCCLHYD scheme.

Chapter 4

High-order DG Cell-centered Lagrangian schemes

This Chapter aims at describing high-order DG schemes for the resolution of the Lagrangian hydrodynamics written under the Lagrangian coordinates. The choice of working on the initial configuration has been made in order to avoid the consideration of curve geometries. However, one has to follow the time evolution of the gradient deformation tensor associated to the Lagrange-Euler flow map. A fundamental requirement is the satisfaction of the Piola compatibility condition. Loubère in [1, 70] has successfully developed a high-order Discontinuous Galerkin Lagrangian method. Nevertheless, the use of this method is limited to a representation in the initial configuration. It was due to the loss of compatibility between the two configurations based on Eulerian and Lagrangian coordinates. Following the approach used by Loubère, a high-order discontinuous Galerkin spectral finite element method has been developed by Jia and Zhang [51]. However, the Piola compatibility condition is not ensured properly. This is due to the loss of consistency in the definition of the deformation gradient tensor in the cells and on the boundaries of the cells. Moreover, the deformation tensor being discretized by means of a Finite Volume method over the cells, this scheme is limited to the representation of computational meshes with straight-line edges. In [22] Cheng and Shu demonstrate that the accuracy degeneracy phenomenon observed for the high order Lagrangian type scheme is due to the error from the quadrilateral mesh with straight-line edges, which restricts the accuracy of the resulting scheme to at most second order. In [22] they present a third-order Lagrangian scheme for solving the Euler equations of compressible gas dynamics on curvilinear meshes. This scheme is based on a high order ENO reconstruction. In this particular scheme, the physical variables are computed through the use of a high order ENO conservative reconstruction. The determination of the vertex velocity is obtained by means of the conserved variables. Here, we present cell-centered DG schemes solving the gas dynamics equation in a total Lagrangian formalism. The flow map is discretized by means of continuous mapping, using a finite element basis. This provides an approximation of the deformation gradient tensor which satisfies the Piola identity. The discretization of the physical conservation laws for the specific volume, the momentum and the total energy relies on a discontinuous Galerkin method ensuring the respect of the GCL.

In the Section 4.1, we present the general framework of the DG discretization of the gas dynamics equations in the total Lagrangian formalism. Then, in Section 4.2, the particular second order case is detailled. Several results are displayed assessing the robustness and the accuracy of the second order DG scheme. Finally, in Section 4.3, we extend the methodology presented in the former sections to the third-order. In this case, the Lagrangian-Eulerian map is spanned by a P_2 Finite Element representation. Thus, the velocity field is quadratic with respect to the space

variables. The mesh edges in the actual configuration are parametrized by means of Bezier curves. In this case, a particular treatment of the GCL is required.

4.1 DG discretization general framework

In this section, we intend to solve the gas dynamics equations in its local Lagrangian form written using the Lagrangian coordinates presented in Section 3.2

$$\frac{\mathrm{d}\,\mathsf{F}}{\mathrm{d}t} = \nabla_X \boldsymbol{U}, \tag{4.1a}$$

$$\rho^0 \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho}) - \nabla_X \cdot (J\mathsf{F}^{-1}\boldsymbol{U}) = 0, \tag{4.1b}$$

$$\rho^0 \frac{\mathrm{d}\,\boldsymbol{U}}{\mathrm{d}t} + \nabla_X \cdot (P\, J\mathsf{F}^{-\mathrm{t}}) = \mathbf{0}, \tag{4.1c}$$

$$\rho^0 \frac{\mathrm{d}\,E}{\mathrm{d}t} + \nabla_X \cdot (J\mathsf{F}^{-1}P\boldsymbol{U}) = 0, \tag{4.1d}$$

where $J = \det \mathsf{F}$ is the Jacobian of F. The thermodynamical closure of this system is obtained through the use of an equation of state, which writes $P = P(\rho, \varepsilon)$, where ε is the specific internal energy, $\varepsilon = E - \frac{1}{2}\boldsymbol{U}^2$. These equations are valid provided that the Lagrangian-Eulerian flow map exists, that is $J > 0$. Here, we introduce G the cofactor matrix of F, *i.e.*, $\mathsf{G} = J\mathsf{F}^{-\mathrm{t}}$. Through the use of this new definition, system (4.1) rewrites

$$\frac{\mathrm{d}\,\mathsf{F}}{\mathrm{d}t} = \nabla_X \boldsymbol{U}, \tag{4.2a}$$

$$\rho^0 \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho}) - \nabla_X \cdot (\mathsf{G}^{\mathrm{t}}\boldsymbol{U}) = 0, \tag{4.2b}$$

$$\rho^0 \frac{\mathrm{d}\,\boldsymbol{U}}{\mathrm{d}t} + \nabla_X \cdot (P\,\mathsf{G}) = \mathbf{0}, \tag{4.2c}$$

$$\rho^0 \frac{\mathrm{d}\,E}{\mathrm{d}t} + \nabla_X \cdot (\mathsf{G}^{\mathrm{t}}P\boldsymbol{U}) = 0. \tag{4.2d}$$

In this framework, the computational grid is fixed, however one has to follow the time evolution of the Jacobian matrix associated to the Lagrange-Euler flow map. We made the choice of working on the initial configuration of the flow to avoid some difficulties inherent to the moving mesh scheme, as dealing with curvilinear geometries, in the case of third-order scheme. Furthermore, in this frame the basis function are time independent and defined on the initial mesh. This mesh being perfectly known, geometrical quantities required by the DG discretization can be computed and stored initially and used during the whole calculation.

Let $\{\Omega_c\}_c$ be a partition of the domain Ω into non-overlapping polygonal cells. We also partitionate the time domain in intermediate times $(t^n)_n$ with $\Delta t^n = t^{n+1} - t^n$ the nth time step. We use a DG discretization in order to develop on each cells our unknowns onto $\mathbb{P}^{\,\alpha}(\Omega_c)$, the set of polynomials of degree up to α. This space approximation leads to a $(\alpha+1)^{th}$ space order accurate scheme. Let ϕ_h^c be the restriction of ϕ_h, the polynomial approximation of the function ϕ, over the cell Ω_c

$$\phi_h^c(\boldsymbol{X}, t) = \sum_{k=0}^{K} \phi_k^c(t)\, \sigma_k^c(\boldsymbol{X}), \tag{4.3}$$

where the ϕ_k^c are the $K+1$ successive components of ϕ_h over the polynomial basis, and σ_k^c the polynomial basis functions. Recalling that the dimension of the polynomial space $\mathbb{P}^{\,\alpha}(\Omega_c)$ is

$\frac{(\alpha+1)(\alpha+2)}{2}$, we have to determine the set of the $\frac{(\alpha+1)(\alpha+2)}{2} = K+1$ basis functions. We make the choice of the two-dimensional extension of the basis (2.53) introduced in the one-dimensional gas dynamics Section 2.3.2. These particular basis functions comes from a Taylor expansion on the cell, located at the center of mass $\boldsymbol{\mathcal{X}}_c = (\mathcal{X}_c, \mathcal{Y}_c)^{\mathrm{t}}$ of the cell Ω_c defined as follows

$$\boldsymbol{\mathcal{X}}_c = \frac{1}{m_c} \int_{\Omega_c} \rho^0(\boldsymbol{X})\, \boldsymbol{X}\, \mathrm{d}V, \tag{4.4}$$

where m_c is the constant mass of the cell Ω_c.

We set the first basis element to 1, *i.e.*, $\sigma_0^c = 1$. Going further in space discretization, the $q+1$ basis functions of degree q, with $0 < q \leq \alpha$, write

$$\sigma^c_{\frac{q(q+1)}{2}+j} = \frac{1}{j!(q-j)!} \left(\frac{X - \mathcal{X}_c}{\Delta X_c}\right)^{q-j} \left(\frac{Y - \mathcal{Y}_c}{\Delta Y_c}\right)^{j}, \tag{4.5}$$

where $j = 0 \ldots q$, $\Delta X_c = \frac{X_{\max} - X_{\min}}{2}$ and $\Delta Y_c = \frac{Y_{\max} - Y_{\min}}{2}$ are the scaling factors where

$$X_{\max\,/\,\min} = \max\,/\,\min_{p \in \mathcal{P}(c)} \{X_p\} \quad \text{and} \quad Y_{\max\,/\,\min} = \max\,/\,\min_{p \in \mathcal{P}(c)} \{Y_p\},$$

knowing that $\mathcal{P}(c)$ is the set of vertices of the cell ω_c. The starting index $\frac{q(q+1)}{2}$ in (4.5) corresponds to the number of polynomial basis functions of degree strictly inferior to q. Let us introduce $\langle \phi \rangle_c$, the mean value of ϕ over the cell Ω_c weighted by the initial density

$$\langle \phi \rangle_c = \frac{1}{m_c} \int_{\Omega_c} \rho^0(\boldsymbol{X})\, \phi(\boldsymbol{X})\, \mathrm{d}V, \tag{4.6}$$

and the associated scalar product

$$\langle \phi, \psi \rangle_c = \int_{\omega_c} \rho^0(\boldsymbol{X})\, \phi(\boldsymbol{X})\, \psi(\boldsymbol{X})\, \mathrm{d}V. \tag{4.7}$$

This particular scalar product is the one used in the projection procedure, refer to 1.1.6, required by the initialization of the functions.

In our polynomial discretization, we want the mass averaged value to be preserved. Consequently, we identify the first component of ϕ_h^c with $\langle \phi \rangle_c$, *i.e.*, $\phi_0^c = \langle \phi \rangle_c$. This definition leads to a particular constraint on the successive basis functions writing

$$\begin{aligned}
\phi_0^c &= \frac{1}{m_c} \int_{\Omega_c} \rho^0\, \phi_h^c\, \mathrm{d}V, \\
&= \frac{1}{m_c} \sum_{k=0}^{K} \phi_k \int_{\Omega_c} \rho^0\, \sigma_k^c\, \mathrm{d}V, \\
&= \phi_0^c + \sum_{k=1}^{K} \phi_k\, \frac{1}{m_c} \int_{\Omega_c} \rho^0\, \sigma_k^c\, \mathrm{d}V, \\
&= \phi_0^c + \sum_{k=1}^{K} \phi_k\, \langle \sigma_k^c \rangle_c\,.
\end{aligned} \tag{4.8}$$

In order to satisfy equation (4.8), we impose $\langle \sigma_k^c \rangle_c = 0,\ \forall k \neq 0$. Consequently, we set a new definition of the $q+1$ basis functions of degree q, with $0 < q \leq \alpha$

$$\sigma^c_{\frac{q(q+1)}{2}+j} = \frac{1}{j!(q-j)!}\left[\left(\frac{X-\mathcal{X}_c}{\Delta X_c}\right)^{q-j}\left(\frac{Y-\mathcal{Y}_c}{\Delta Y_c}\right)^{j} - \left\langle\left(\frac{X-\mathcal{X}_c}{\Delta X_c}\right)^{q-j}\left(\frac{Y-\mathcal{Y}_c}{\Delta Y_c}\right)^{j}\right\rangle_c\right]. \quad (4.9)$$

We point out that the only geometry contributions presents in (4.9) are the center of mass position and the scaling factors. Consequently, regardless the shape of the cell is, we could use the same class of basis functions. We also note that the first moment ϕ_0 associated to the first basis element 1 is the mass averaged value of the function ϕ over the cell Ω_c. If the discretization ends here, the scheme would correspond to the Finite Volumes method. Going further, the successive moments can be identified as the successive derivatives of the function expressed at the center of mass of the cell. Here, to simplify the notation, we identify the functions $\frac{1}{\rho}$, $\boldsymbol{U}$, E and P to their polynomial approximations over the considered cell Ω_c, respectively $(\frac{1}{\rho})^c_h$, $\boldsymbol{U}^c_h$, E^c_h and P^c_h.

4.1.1 Variational formulations

The discontinuous Galerkin discretization of the gas dynamics equations results from a local variational formulation on each cell of equations (4.2b), (4.2c) and (4.2d)

$$\int_{\Omega_c}\rho^0\frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})\sigma_q^c\,\mathrm{d}V = \sum_{k=0}^{K}\frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})^c_k\int_{\Omega_c}\rho^0\sigma_q^c\sigma_k^c\,\mathrm{d}V = -\int_{\Omega_c}\boldsymbol{U}\,.\,\mathsf{G}\,\boldsymbol{\nabla}_X\sigma_q^c\,\mathrm{d}V + \int_{\partial\Omega_c}\overline{\boldsymbol{U}}\,.\,\sigma_q^c\,\mathsf{G}\boldsymbol{N}\mathrm{d}S, \quad (4.10)$$

$$\int_{\Omega_c}\rho^0\frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t}\sigma_q^c\,\mathrm{d}V = \sum_{k=0}^{K}\frac{\mathrm{d}\boldsymbol{U}^c_k}{\mathrm{d}t}\int_{\Omega_c}\rho^0\sigma_q^c\sigma_k^c\,\mathrm{d}V = \int_{\Omega_c}P\,\mathsf{G}\,\boldsymbol{\nabla}_X\sigma_q^c\,\mathrm{d}V - \int_{\partial\Omega_c}\overline{P}\,\sigma_q^c\,\mathsf{G}\boldsymbol{N}\mathrm{d}S, \quad (4.11)$$

$$\int_{\Omega_c}\rho^0\frac{\mathrm{d}E}{\mathrm{d}t}\sigma_q^c\,\mathrm{d}V = \sum_{k=0}^{K}\frac{\mathrm{d}E^c_k}{\mathrm{d}t}\int_{\Omega_c}\rho^0\sigma_q^c\sigma_k^c\,\mathrm{d}V = \int_{\Omega_c}P\boldsymbol{U}\,.\,\mathsf{G}\,\boldsymbol{\nabla}_X\sigma_q^c\,\mathrm{d}V - \int_{\partial\Omega_c}\overline{P\boldsymbol{U}}\,.\,\sigma_q^c\,\mathsf{G}\boldsymbol{N}\mathrm{d}S, \quad (4.12)$$

where σ_q^c is a function picked into the chosen basis $\{\sigma_k^c\}_{k=0\ldots K}$ of $\mathbb{P}^K$, and $\overline{\boldsymbol{U}}$, $\overline{P}$ and $\overline{P\boldsymbol{U}}$ are the continuous numerical flux functions defined at cell interfaces. Here, we can identify $\int_{\Omega_c}\rho^0\sigma_q^c\sigma_k^c\,\mathrm{d}V = \langle\sigma_q^c,\sigma_k^c\rangle_c$ as the generic coefficient of the symmetric positive definite mass matrix. Recalling that $\int_{\Omega_c}\rho^0\,\sigma_k^c\,\mathrm{d}V = 0,\ \forall k \neq 0$, the equations corresponding to mass averaged values are independent of the other polynomial basis components equations.

These variational formulations are written on the initial configuration. Thanks to the identities presented in Section 3.2, $\mathrm{d}v = J\mathrm{d}V$ and $\boldsymbol{n}\mathrm{d}s = J\mathsf{F}^{-\mathrm{t}}\boldsymbol{N}\mathrm{d}S$, we are able to write (4.10), (4.11) and (4.12) over the actual configuration $\omega_c = \boldsymbol{\Phi}(\Omega_c, t)$, at time $t > 0$

$$\int_{\omega_c}\rho\frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})\sigma_q^c\,\mathrm{d}v = -\int_{\omega_c}\boldsymbol{U}\,.\,\boldsymbol{\nabla}_x\sigma_q^c\,\mathrm{d}v + \int_{\partial\omega_c}\overline{\boldsymbol{U}}\,.\,\sigma_q^c\,\boldsymbol{n}\mathrm{d}s, \quad (4.13)$$

$$\int_{\omega_c}\rho\frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t}\sigma_q^c\,\mathrm{d}v = \int_{\omega_c}P\,\boldsymbol{\nabla}_x\sigma_q^c\,\mathrm{d}v - \int_{\partial\omega_c}\overline{P}\,\sigma_q^c\,\boldsymbol{n}\mathrm{d}s, \quad (4.14)$$

$$\int_{\omega_c}\rho\frac{\mathrm{d}E}{\mathrm{d}t}\sigma_q^c\,\mathrm{d}v = \int_{\omega_c}P\boldsymbol{U}\,.\,\boldsymbol{\nabla}_x\sigma_q^c\,\mathrm{d}v - \int_{\partial\omega_c}\overline{P\boldsymbol{U}}\,.\,\sigma_q^c\,\boldsymbol{n}\mathrm{d}s. \quad (4.15)$$

Remark 2 *The equation* (4.13) *has an interesting interpretation. Rewriting it for* $\sigma_0^c = 1$*, one gets*

$$\int_{\omega_c} \rho \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})\,\mathrm{d}v = \int_{\partial\omega_c} \boldsymbol{U}\,.\,\boldsymbol{n}\mathrm{d}s.$$

And thanks to (3.29)*, the left-hand side of this last equations yields*

$$\int_{\omega_c} \rho \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})\,\mathrm{d}v = \frac{\mathrm{d}}{\mathrm{d}t}\int_{\omega_c} \rho(\frac{1}{\rho})\,\mathrm{d}v = \int_{\omega_c} \mathrm{d}v.$$

Consequently, equation (4.13) *is equivalent to the following equation*

$$\int_{\omega_c} \mathrm{d}v = \int_{\partial\omega_c} \boldsymbol{U}\,.\,\boldsymbol{n}\mathrm{d}s,$$

which is nothing but the geometrical conservation law (3.44)*. Thus,* (4.13) *in the general case can be viewed as an extension of the GCL that is strongly linked to the Reynolds transport theorem. Indeed, let us note that equation* (4.13) *is equivalent to*

$$\int_{\omega_c} \rho \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})\sigma_q^c\,\mathrm{d}v = \int_{\omega_c} \sigma_q^c \nabla_x\,.\,\boldsymbol{U}\,\mathrm{d}v. \tag{4.16}$$

Using the fact that $\frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})\,\sigma_q^c = \frac{\mathrm{d}}{\mathrm{d}t}(\sigma_q^c(\frac{1}{\rho})) - (\frac{1}{\rho})\frac{\mathrm{d}\,\sigma_q^c}{\mathrm{d}t}$ *and again relation* (3.29)*, equation* (4.16) *rewrites*

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\omega_c} \sigma_q^c\,\mathrm{d}v = \int_{\omega_c} \left(\frac{\mathrm{d}\,\sigma_q^c}{\mathrm{d}t} + \sigma_q^c \nabla_x\,.\,\boldsymbol{U}\right)\mathrm{d}v, \tag{4.17}$$

which is nothing but the Reynolds transport formula. Hence (4.13) *is equivalent to* (4.17)*, where* σ_q^c *is a polynomial function picked into the chosen basis* $\{\sigma_k^c\}_{k=0\ldots K}$*.*

Remark 3 *Let us recall that in the particular case of an uniform flow, meaning the pressure and the velocity fields are constant, the gas dynamics equations reduce to the simple advection of the density field. Considering* (4.13)*,* (4.14) *and* (4.15)*, in the uniform flow case the left-hand side of these equations must be equal to zero. This means that the following relationship must hold for all basis functions*

$$\int_{\omega_c} \boldsymbol{\nabla}_x \sigma_q^c\,\mathrm{d}v = \int_{\partial\omega_c} \sigma_q^c\,\boldsymbol{n}\mathrm{d}s. \tag{4.18}$$

This above equation is nothing but a consequence of the Green-Gauss formula. It is clear that to ensure relation (4.18)*, the quadrature formulas employed to discretize interior and boundary terms must be compatible in the sense that they ensure the identity* (4.18) *at the discrete level. This compatibility requirement has a strong impact on the ability of the numerical scheme to preserve uniform flows.*

B. Cockburn, S. Hou and C.-W. Shu demonstrate in their article [25] that to design a $(\alpha+1)^{th}$ *order numerical scheme, a quadrature rule over the faces being exact for polynomials of degree* $2\alpha+1$ *is needed, as a quadrature rule over the elements being exact for polynomials of degree* 2α*. For the numerical applications, to evaluate properly the interior terms we make use of the two-dimensional quadrature formula defined on triangles, found in [37]. This quadrature is exact for polynomials up to degree* 5*. Thus, this formula is accurate enough for the implementation*

quad. points	quad. weights
$(\frac{1}{3}, \frac{1}{3})$	$\frac{9}{80}$
(a_-, a_-)	w_-
$(a_-, 1-2a_-)$	w_-
$(1-2a_-, a_-)$	w_-
(a_+, a_+)	w_+
$(a_+, 1-2a_+)$	w_+
$(1-2a_+, a_+)$	w_+

Table 4.1: Quadrature points and weights of the chosen two-dimensional quadrature rule, where $a_\pm = \frac{6\pm\sqrt{15}}{21}$ and $w_\pm = \frac{155\pm\sqrt{15}}{2400}$.

of a third-order scheme where $\alpha = 2$. This quadrature rule defined by its quadrature points and weights write as following on the reference triangle $((0,0),(1,0),(0,1))$, refer to Table 4.1. Concerning the boundary terms, we perform a special treatment on the face integrals enforcing relation (4.18)*. This point will be detailed in the second-order and third-order cases.*

Identity (4.18) *on the actual configuration corresponds to the following equation on the initial configuration*

$$\int_{\Omega_c} \mathsf{G}\boldsymbol{\nabla}_X \sigma_q^c \,\mathrm{d}V = \int_{\partial\Omega_c} \sigma_q^c \,\mathsf{G}\boldsymbol{N}\mathrm{d}S, \tag{4.19}$$

which is nothing but a generalization of the weak form of the Piola compatibility condition which writes

$$\int_{\Omega_c} \sigma_q^c \,(\nabla_X \cdot \mathsf{G})\,\mathrm{d}V = \mathbf{0}. \tag{4.20}$$

Considering identity (4.19)*, we clearly see that some consistency is required on the discretization of the deformation gradient tensor F inside the cell and on the boundaries. Moreover, one knows that the vector $\mathsf{G}\boldsymbol{N}$, where $\boldsymbol{N}$ is the unit outward normal of an edge in the initial configuration, corresponds to the outward normal $\boldsymbol{n}$ in the actual configuration. Thus, to maintain the coherence between the two configurations and ensure the geometry continuity, the vector $\mathsf{G}\boldsymbol{N}$ has to be continuous at the interfaces of the cells. These are the reasons that have motivated our choice of discretizing the tensor F by means of a mapping using a Finite Element basis on triangular cells.*

4.1.2 Deformation gradient tensor discretization

Let Ω_c be a generic polygonal cell in the initial configuration, partitioned into non-overlapping triangles $\mathcal{T}_i^c$ as $\Omega_c = \bigcup_{i=1}^{ntri} \mathcal{T}_i^c$, refer to Figure 4.1. Now, recalling the mapping formulation expressed in (3.1), we develop $\boldsymbol{\Phi}$ on the Finite Element basis functions λ_p

$$\boldsymbol{\Phi}_h^i(\boldsymbol{X}, t) = \sum_p \lambda_p(\boldsymbol{X})\; \boldsymbol{\Phi}_p(t),$$

where the points p are control points including vertices in a generic triangle $\mathcal{T}_i$. We identify $\boldsymbol{\Phi}_p(t) = \boldsymbol{\Phi}(\boldsymbol{X}_p, t)$ as the position at time t of the control point initially located at $\boldsymbol{X}_p$. The functions λ_p are the Finite Element basis functions. Thus, they identify with the P_1 barycentric

coordinate basis functions in the case of our second-order scheme. For the third-order scheme, we make use of the P_2 Finite Elements basis functions.

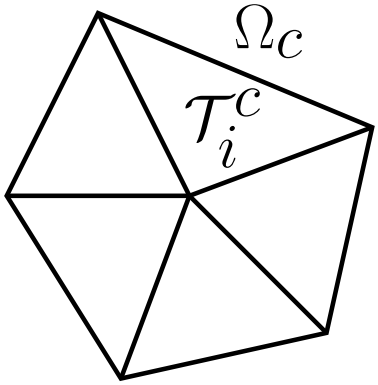


Figure 4.1: Triangular subdivision of a generic polygonal cell Ω.

Using this continuous polynomial mapping approximation and the definition (3.9) of F leads to the expression of this tensor in the triangle $\mathcal{T}_i$

$$\mathsf{F}_i(\boldsymbol{X},t) = \sum_p \boldsymbol{\Phi}_p(t) \otimes \boldsymbol{\nabla}_X \lambda_p(\boldsymbol{X}). \tag{4.21}$$

Let us check that this definition of the discrete deformation gradient tensor satisfies the Piola condition which ensures the compatibility between the initial and the actual configuration. Here, we recall that G is the cofactor matrix of F, *i.e.*, $\mathsf{G} = J\mathsf{F}^{-\mathrm{t}}$. In the two-dimensional case, the use of (4.21) to express G_i yields

$$\mathsf{G}_i = \begin{pmatrix} \sum_p \Phi_p^Y \partial_Y \lambda_p & -\sum_p \Phi_p^Y \partial_X \lambda_p \\ -\sum_p \Phi_p^X \partial_Y \lambda_p & \sum_p \Phi_p^X \partial_X \lambda_p \end{pmatrix} = \sum_p \begin{pmatrix} \Phi_p^Y \partial_Y \lambda_p & -\Phi_p^Y \partial_X \lambda_p \\ -\Phi_p^X \partial_Y \lambda_p & \Phi_p^X \partial_X \lambda_p \end{pmatrix}. \tag{4.22}$$

Taking the divergence of equation (4.22), one gets

$$\nabla_X \cdot \mathsf{G}_i = \sum_p \begin{pmatrix} \Phi_p^Y (\partial_{YX}\lambda_p - \partial_{XY}\lambda_p) \\ \Phi_p^X (\partial_{XY}\lambda_p - \partial_{YX}\lambda_p) \end{pmatrix} = \mathbf{0}. \tag{4.23}$$

This equation shows that the Piola compatibility condition is satisfied by construction. This result can be generalized to the three dimensional case with a similar procedure. If F_{jk} is the $(j,k)^{th}$ entry of matrix F_i, we compute the cofactor of F starting with the diagonal elements

$$\mathsf{G}_{jj} = \mathsf{F}_{kk}\mathsf{F}_{ll} - \mathsf{F}_{kl}\mathsf{F}_{lk},$$

where j, k and l are an ordered set of the integers 1, 2 and 3, corresponding to the three directions $\boldsymbol{e}_X$, $\boldsymbol{e}_Y$ and $\boldsymbol{e}_Z$. For the off-diagonal entries, G_{jk} and G_{kj} write

$$\mathsf{G}_{jk} = \mathsf{F}_{lj}\mathsf{F}_{kl} - \mathsf{F}_{ll}\mathsf{F}_{kj},$$
$$\mathsf{G}_{kj} = \mathsf{F}_{lk}\mathsf{F}_{jl} - \mathsf{F}_{ll}\mathsf{F}_{jk}.$$

Inserting in these elements formulas the development of the tensor F (4.21), the G_i matrix entries rewrite

$$\begin{aligned}
\mathsf{G}_{jj} &= \sum_{p,q} \left(\Phi_p^k \partial_k \lambda_p \, \Phi_q^l \partial_l \lambda_q - \Phi_p^k \partial_l \lambda_p \, \Phi_q^l \partial_k \lambda_q \right), \\
&= \sum_{p,q} \Phi_p^k \Phi_q^l \left(\partial_k \lambda_p \, \partial_l \lambda_q - \partial_l \lambda_p \, \partial_k \lambda_q \right), \\
&= \sum_{p,q} \Phi_p^k \Phi_q^l \left(\boldsymbol{\nabla}_X \lambda_p \times \boldsymbol{\nabla}_X \lambda_q \right)_j,
\end{aligned}$$

where $\sum_{p,q}$ is a double sum on the control points, Φ_p^j corresponds to j^{th} component of the vector $\boldsymbol{\Phi}_p$ and $(\boldsymbol{\nabla}_X \lambda_p)_j = \partial_j \lambda_p$. Likewise for the off-diagonal entries, one gets

$$\begin{aligned}
\mathsf{G}_{jk} &= \sum_{p,q} \Phi_p^k \Phi_q^l \left(\boldsymbol{\nabla}_X \lambda_p \times \boldsymbol{\nabla}_X \lambda_q \right)_k, \\
\mathsf{G}_{kj} &= \sum_{p,q} \Phi_p^j \Phi_q^l \left(\boldsymbol{\nabla}_X \lambda_p \times \boldsymbol{\nabla}_X \lambda_q \right)_j.
\end{aligned}$$

Thus, we can express the cofactor matrix G_i in the following matricial form

$$\mathsf{G}_i = \sum_{p,q} \boldsymbol{\Psi}_{pq} \otimes \left(\boldsymbol{\nabla}_X \lambda_p \times \boldsymbol{\nabla}_X \lambda_q \right), \tag{4.24}$$

where the vector $\boldsymbol{\Psi}_{pq}$ is defined such as its jth component reads $\boldsymbol{\Psi}_{pq}^j = \Phi_p^k \Phi_q^l$. Taking the divergence of equation (4.24) yields

$$\begin{aligned}
\nabla_X \cdot \mathsf{G}_i &= \sum_{p,q} \left(\nabla_X \cdot \left(\boldsymbol{\nabla}_X \lambda_p \times \boldsymbol{\nabla}_X \lambda_q \right) \right) \boldsymbol{\Psi}_{pq}, \\
&= \sum_{p,q} \left(\boldsymbol{\nabla}_X \lambda_q \cdot \underbrace{\left(\nabla_X \times \boldsymbol{\nabla}_X \lambda_p \right)}_{=\mathbf{0}} \right) \boldsymbol{\Psi}_{pq}, \\
&= \mathbf{0}.
\end{aligned} \tag{4.25}$$

Again the Piola compatibility condition is satisfied by construction of the scheme.

Now the use of the trajectory equation $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Phi}_p = \boldsymbol{U}_p$ leads to a semi-discrete equation of the deformation gradient tensor

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathsf{F}_i(\boldsymbol{X}, t) = \sum_p \boldsymbol{U}_p(t) \otimes \boldsymbol{\nabla}_X \lambda_p(\boldsymbol{X}), \tag{4.26}$$

where $\boldsymbol{U}_p$ is the velocity of the control point p. This is the chosen semi-discrete equation we use to determine the deformation gradient tensor on the triangles constituting the polygonal cells. The deformation gradient tensor is discretized through the same equation in [56], in their cell-centered Lagrangian scheme for the hyperelasticity.

At this point, we did not make any assumption on the order of the discretization. The numerical accuracy deriving from equation (4.26) depends on the choice of the Finite Element basis. Regarding (3.10), we make the assumption that the spatial approximation order of the deformation gradient tensor could be one less than the velocity, and consequently than the polynomial

approximation coming from the DG discretization. Consequently, F would be piecewise constant over the triangles for a second-order scheme, and piecewise linear for a third-order scheme.

Now, we want to show that the discretization of F ensures the continuity of $\mathsf{G}\boldsymbol{N}$ at cell interfaces, where $\boldsymbol{N}$ is the interface unite outward normal in the initial configuration. But first, we prove that the properties of the $\mathsf{G}\boldsymbol{N}$ are equivalent to the ones of $\mathsf{F}\boldsymbol{T}$, where $\boldsymbol{T}$ is the unit tangent vector, orthogonal to $\boldsymbol{N}$. Let us introduce the infinitesimal displacement in the actual configuration $\mathrm{d}\boldsymbol{x}$ corresponding through the flow motion to $\mathrm{d}\boldsymbol{X}$ in the initial configuration. We define the two tangent vectors $\boldsymbol{t}$ and $\boldsymbol{T}$ such as $\mathrm{d}\boldsymbol{x} = \boldsymbol{t}\,\mathrm{d}l$ and $\mathrm{d}\boldsymbol{X} = \boldsymbol{T}\mathrm{d}L$. We have $\mathrm{d}\boldsymbol{x} = \mathsf{F}\mathrm{d}\boldsymbol{X}$ by definition of F (3.14), and hence

$$\boldsymbol{t}\,\mathrm{d}l = \mathsf{F}\boldsymbol{T}\mathrm{d}L. \tag{4.27}$$

This is another form of the Nanson formula, expressing the transport of a tangent vector through the flow motion of the fluid. Now, in the two-dimensional particular case, one knows that

$$\boldsymbol{n}\mathrm{d}l = (\boldsymbol{t}\mathrm{d}l)^{\perp},$$

where $\boldsymbol{a}^{\perp} = (a_2, -a_1)^{\mathrm{t}}$ with $\boldsymbol{a} = (a_1, a_2)^{\mathrm{t}}$. Finally, using the classical Nanson relation (3.17) in this last identity yields

$$\mathsf{G}\boldsymbol{N} = (\mathsf{F}\boldsymbol{T})^{\perp}\,. \tag{4.28}$$

This relation ensures the equivalency between the properties of $\mathsf{G}\boldsymbol{N}$ and $(\mathsf{F}\boldsymbol{T})^{\perp}$.

Remark 4 *In some numerical applications, the positivity assumption on the Jacobian, $J > 0$, may not be satisfied. Indeed, in strong shock regime and vortices, the Jacobian J may become negative. In this method we compute the tensor F, although the one used is G, the cofactor matrix. The question of the invertibility of F in the calculation of G is not crucial because the function $\mathsf{F} \longrightarrow J\mathsf{F}^{-t}$ is continuous (linear in two-dimensions and quadratic in three-dimensions regarding its components). That is the reason why the numerical calculation would not stop in this extreme case. But obviously, to work on the fixed initial mesh does not solve the problem of mesh entangling, since the difficulty is then transferred to the numerical solution of F. Thus, in some extreme cases, an ALE approach could be needed, not regarding the shape of the moving mesh but its deformation gradient tensor.*

Now, we want demonstrate that the deformation gradient tensor discretization respects the transport formulas (3.17) and (4.27). In the case of infinitesimal displacements, the Eulerian tangent $\boldsymbol{t}\,\mathrm{d}l$ writes

$$\begin{aligned}
\boldsymbol{t}\,\mathrm{d}l = \mathrm{d}\boldsymbol{x} &= \mathrm{d}\boldsymbol{\Phi}, \\
&= \mathrm{d}\left(\sum_p \lambda_p(\boldsymbol{X})\,\boldsymbol{\Phi}_p\right) \\
&= \sum_p \left(\boldsymbol{\nabla}_X \lambda_p \,\boldsymbol{\cdot}\, \mathrm{d}\boldsymbol{X}\right)\boldsymbol{\Phi}_p \\
&= \sum_p \left(\boldsymbol{\Phi}_p \otimes \boldsymbol{\nabla}_X \lambda_p\right)\mathrm{d}\boldsymbol{X} \\
&= \mathsf{F}_i\,\mathrm{d}\boldsymbol{X} = \mathsf{F}_i\,\boldsymbol{T}\,\mathrm{d}L.
\end{aligned} \tag{4.29}$$

Finally, the discretization (4.21) of the tensor F preserves the transport relations (4.27).

Finally, we set up the general framework of the discretization of the deformation gradient tensor using Finite Element basis functions. We partitionate our polygonal cells Ω_c into $ntri$ triangles $\mathcal{T}_i^c$ with $i = 1, \ldots, ntri$ and make use of the semi-discrete equation (4.26). Now, coming back to the discontinuous Galerkin discretization of the specific volume, the velocity and the total energy equations, we derive the numerical fluxes in order to ensure a local entropy inequality.

4.1.3 Entropic analysis

Let η be the specific entropy and θ the absolute temperature defined by means of the Gibbs identity as follows

$$\begin{aligned}\theta \mathrm{d}\eta &= \mathrm{d}\varepsilon + P\mathrm{d}(\frac{1}{\rho}),\\ &= \mathrm{d}E - \boldsymbol{U}\,.\,\mathrm{d}\boldsymbol{U} + P\mathrm{d}(\frac{1}{\rho}).\end{aligned} \tag{4.30}$$

We are going to write the time rate of change of the specific entropy. Here, we remind that we identify the functions $\frac{1}{\rho}$, $\boldsymbol{U}$, E and P to their polynomial approximation over the considered cell Ω_c, respectively $(\frac{1}{\rho})_h^c$, $\boldsymbol{U}_h^c$, E_h^c and P_h^c. The method is quite similar to the one used in the one-dimensional case, refer to Section 2.3.3. We perform variational formulations on the gas dynamics equations (4.2b),(4.2c) and (4.2d) with respectively P, $\boldsymbol{U}$ and 1 as test functions

$$\int_{\Omega_c} \rho^0\, P \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})\,\mathrm{d}V = \int_{\partial\Omega_c} P\,\overline{\boldsymbol{U}}\,.\,\mathsf{G}\boldsymbol{N}\mathrm{d}S - \int_{\Omega_c} \boldsymbol{U}\,.\,\mathsf{G}\boldsymbol{\nabla}_X P\,\mathrm{d}V, \tag{4.31a}$$

$$\int_{\Omega_c} \rho^0\, \boldsymbol{U}\,.\,\frac{\mathrm{d}\,\boldsymbol{U}}{\mathrm{d}t}\,\mathrm{d}V = -\int_{\partial\Omega_c} \overline{P}\,\boldsymbol{U}\,.\,\mathsf{G}\boldsymbol{N}\mathrm{d}S + \int_{\Omega_c} P\ (\mathsf{G} : \nabla_X \boldsymbol{U})\ \mathrm{d}V, \tag{4.31b}$$

$$\int_{\Omega_c} \rho^0 \frac{\mathrm{d}\,E}{\mathrm{d}t}\,\mathrm{d}V = -\int_{\partial\Omega_c} \overline{P\boldsymbol{U}}\,.\,\mathsf{G}\boldsymbol{N}\mathrm{d}S. \tag{4.31c}$$

We identify these equations respectively to the pressure work, the kinetic energy and the total energy semi-discrete equations. Making use of the Gibbs formula (4.30), the (4.31c)-(4.31b)+(4.31a) combination leads to

$$\begin{aligned}\int_{\Omega_c} \rho^0\,\theta \frac{\mathrm{d}\,\eta}{\mathrm{d}t}\,\mathrm{d}V = &\int_{\partial\Omega_c} \left[\overline{P}\,\boldsymbol{U} + P\,\overline{\boldsymbol{U}} - \overline{P\boldsymbol{U}}\right]\,.\,\mathsf{G}\boldsymbol{N}\mathrm{d}S\\ &- \int_{\Omega_c} \underbrace{\left[P\ (\mathsf{G} : \nabla_X \boldsymbol{U}) + \boldsymbol{U}\,.\,\mathsf{G}\boldsymbol{\nabla}_X P\right]}_{\mathrm{A}}\,\mathrm{d}V.\end{aligned}$$

The use of the Piola compatibility condition gives us

$$\begin{aligned}\nabla_X\,.\,(\mathsf{G}^{\mathrm{t}} P\,\boldsymbol{U}) &= \boldsymbol{U}\,.\,\mathsf{G}\boldsymbol{\nabla}_X P + P\,\nabla_X\,.\,(\mathsf{G}^{\mathrm{t}}\boldsymbol{U}),\\ &= \boldsymbol{U}\,.\,\mathsf{G}\boldsymbol{\nabla}_X P + P\,\boldsymbol{U}\,.\,\underbrace{(\boldsymbol{\nabla}_X\,.\,\mathsf{G})}_{=\mathbf{0}\ \text{Piola}} + P\ (\mathsf{G} : \nabla_X \boldsymbol{U})\,,\\ &= \mathrm{A}.\end{aligned}$$

We finally get an expression of the time rate of change of the specific entropy

$$\int_{\Omega_c} \rho^0\,\theta \frac{\mathrm{d}\,\eta}{\mathrm{d}t}\,\mathrm{d}V = \int_{\partial\Omega_c} \left[\overline{P}\,\boldsymbol{U} + P\,\overline{\boldsymbol{U}} - \overline{P\boldsymbol{U}} - P\,\boldsymbol{U}\right]\,.\,\mathsf{G}\boldsymbol{N}\mathrm{d}S. \tag{4.32}$$

At this point, it remains to express the numerical fluxes in such way that an entropy inequality is satisfied. To this end, we first make the following fundamental assumption

$$\overline{P\boldsymbol{U}} = \overline{P}\,\overline{\boldsymbol{U}}. \tag{4.33}$$

This assumption allows to factorize the right-hand side of equation (4.32)

$$\int_{\Omega_c} \rho^0\,\theta \frac{\mathrm{d}\,\eta}{\mathrm{d}t}\,\mathrm{d}V = \int_{\partial\Omega_c} (\overline{P} - P)(\boldsymbol{U} - \overline{\boldsymbol{U}})\,\boldsymbol{.}\,\mathsf{G}\boldsymbol{N}\mathrm{d}S. \tag{4.34}$$

This relation transforms into the following expression when coming back to the actual configuration

$$\int_{\omega_c} \rho\,\theta \frac{\mathrm{d}\,\eta}{\mathrm{d}t}\,\mathrm{d}v = \int_{\partial\omega_c} (\overline{P} - P)(\boldsymbol{U} - \overline{\boldsymbol{U}})\,\boldsymbol{.}\,\boldsymbol{n}\mathrm{d}s. \tag{4.35}$$

To enforce a local entropy inequality at the semi-discrete level, we set the following sufficient condition on the numerical fluxes

$$\begin{aligned} \overline{P} - P &= -Z\,(\overline{\boldsymbol{U}} - \boldsymbol{U})\,\boldsymbol{.}\,\boldsymbol{n}, \\ &= -Z\,(\overline{\boldsymbol{U}} - \boldsymbol{U})\,\boldsymbol{.}\,\frac{\mathsf{G}\boldsymbol{N}}{\|\mathsf{G}\boldsymbol{N}\|}, \end{aligned} \tag{4.36}$$

where Z is a positive constant with a physical dimension of a density times a velocity. For the numerical applications, we use $Z = \rho\,a$ which corresponds to the acoustic impedance.

Finally, we have derived a particular form of the numerical fluxes ensuring a local entropy inequality for the semi-discrete scheme. We have now identified most of the difficulties and presented the main ingredients inherent of the DG discretization of the Lagrangian gas dynamics equations written using the Lagrangian coordinates. Before detailing the two cases of the second-order and the third-order schemes, we present the two-dimensional extension of the limitation presented in Section 2.3.4.

4.1.4 Limiting procedure based on the characteristic variables

In the case of discontinuous problem, without a specific treatment, high-order numerical schemes produce solutions containing spurious oscillations. On the other hand, if we apply a limiting procedure directly to the polynomial approximation of the physical variables $(\frac{1}{\rho})_h$, $\boldsymbol{U}_h$ and E_h, we cannot enforce totally the monotonicity of the solutions. To correct this flaw, as we did in the one-dimensional case, we construct a limiting procedure based on the Riemann invariants.

Let us assume that the fluid variables are sufficiently smooth to compute their partial derivatives. Here, we recall the Lagrangian gas dynamics equations written under the non-conservative form

$$\rho\frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho}) - \nabla_x\,\boldsymbol{.}\,\boldsymbol{U} = 0, \tag{4.37a}$$

$$\rho\frac{\mathrm{d}\,\boldsymbol{U}}{\mathrm{d}t} + \boldsymbol{\nabla}_x P = \boldsymbol{0}, \tag{4.37b}$$

$$\rho\frac{\mathrm{d}\,E}{\mathrm{d}t} + \nabla_x\,\boldsymbol{.}\,(P\boldsymbol{U}) = 0. \tag{4.37c}$$

Dot-multiplying the momentum equation (4.37b) by the velocity and subtracting it from the total energy equation (4.37c) leads to

$$\rho \frac{\mathrm{d}\,\varepsilon}{\mathrm{d}t} + P \nabla_x \cdot \boldsymbol{U} = 0,$$

where ε is the specific internal energy. Now, substituting the volume equation (4.37a) in the above equation leads to

$$\rho \frac{\mathrm{d}\,\varepsilon}{\mathrm{d}t} + P\,\rho \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho}) = 0.$$

Recalling the definition of the specific entropy, η, thanks to the Gibbs formula

$$\theta \mathrm{d}\eta = \mathrm{d}\varepsilon + P\,\mathrm{d}(\frac{1}{\rho}),$$

where θ denotes the temperature, allows to write the time rate of change of the specific entropy as

$$\frac{\mathrm{d}\,\eta}{\mathrm{d}t} = 0. \tag{4.38}$$

This equation shows that the specific entropy is conserved along the trajectory equation. Thanks to this result, one gets the following relation between the pressure and the density material derivatives

$$\frac{\mathrm{d}\,P}{\mathrm{d}t} = -\rho^2 a^2\, \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho}). \tag{4.39}$$

Gathering the previous results, we obtain the new non-conservative form of the gas dynamics equations

$$\frac{\mathrm{d}\,P}{\mathrm{d}t} + \rho a^2 \nabla_x \cdot \boldsymbol{U} = 0, \tag{4.40a}$$
$$\frac{\mathrm{d}\,\boldsymbol{U}}{\mathrm{d}t} + \frac{1}{\rho}\boldsymbol{\nabla}_x P = 0, \tag{4.40b}$$
$$\frac{\mathrm{d}\,\eta}{\mathrm{d}t} = 0. \tag{4.40c}$$

Expanding the above system in the two dimensional Cartesian frame yields

$$\frac{\partial\,P}{\partial t} + U\frac{\partial\,P}{\partial x} + V\frac{\partial\,P}{\partial y} + \rho a^2(\frac{\partial\,U}{\partial x} + \frac{\partial\,V}{\partial y}) = 0, \tag{4.41a}$$
$$\frac{\partial\,U}{\partial t} + U\frac{\partial\,U}{\partial x} + V\frac{\partial\,U}{\partial y} + \frac{1}{\rho}\frac{\partial\,P}{\partial x} = 0, \tag{4.41b}$$
$$\frac{\partial\,V}{\partial t} + U\frac{\partial\,V}{\partial x} + V\frac{\partial\,V}{\partial y} + \frac{1}{\rho}\frac{\partial\,P}{\partial y} = 0, \tag{4.41c}$$
$$\frac{\partial\,\eta}{\partial t} + U\frac{\partial\,\eta}{\partial x} + V\frac{\partial\,\eta}{\partial y} = 0, \tag{4.41d}$$

where U and V are respectively the x and y components of $\boldsymbol{U}$. Setting $\mathrm{W} = (P, U, V, \eta)^{\mathrm{t}}$, we can rewrite system (4.40) as

$$\frac{\partial\,\mathrm{W}}{\partial t} + \mathsf{A}_x\frac{\partial\,\mathrm{W}}{\partial x} + \mathsf{A}_y\frac{\partial\,\mathrm{W}}{\partial y} = 0, \tag{4.42}$$

where the matrix 4×4 matrices A_x and A_y are given by

$$\mathsf{A}_x = \begin{pmatrix} U & \rho\, a^2 & 0 & 0 \\ \frac{1}{\rho} & U & 0 & 0 \\ 0 & 0 & U & 0 \\ 0 & 0 & 0 & U \end{pmatrix} \quad \text{and} \quad \mathsf{A}_y = \begin{pmatrix} V & 0 & \rho\, a^2 & 0 \\ 0 & V & 0 & 0 \\ \frac{1}{\rho} & 0 & V & 0 \\ 0 & 0 & 0 & V \end{pmatrix}.$$

If $\boldsymbol{n}$ denotes a unit vector, we define $\mathsf{A}(\boldsymbol{n}) = \mathsf{A}_x n_x + \mathsf{A}_y n_y$

$$\mathsf{A}(\boldsymbol{n}) = \begin{pmatrix} \boldsymbol{U} \cdot \boldsymbol{n} & \rho\, a^2\, n_x & \rho\, a^2\, n_y & 0 \\ \frac{1}{\rho}\, n_x & \boldsymbol{U} \cdot \boldsymbol{n} & 0 & 0 \\ \frac{1}{\rho}\, n_y & 0 & \boldsymbol{U} \cdot \boldsymbol{n} & 0 \\ 0 & 0 & 0 & \boldsymbol{U} \cdot \boldsymbol{n} \end{pmatrix}. \tag{4.43}$$

This matrix admits four real eigenvalues: $\lambda_1 = \boldsymbol{U} \cdot \boldsymbol{n} - a$, $\lambda_2 = \lambda_3 = \boldsymbol{U} \cdot \boldsymbol{n}$, $\lambda_4 = \boldsymbol{U} \cdot \boldsymbol{n} + a$. One knows that the system of equations (4.41) is hyperbolic if for all $\boldsymbol{n}$, $\mathsf{A}(\boldsymbol{n})$ admits four real eigenvalues. Consequently, the above system is hyperbolic. Now, we are able to introduce the four differential Riemann invariants associated to unit direction $\boldsymbol{n}$

$$\mathrm{d}\alpha_E = \mathrm{d}E - \boldsymbol{U} \cdot \mathrm{d}\boldsymbol{U} + P\, \mathrm{d}(\frac{1}{\rho}), \tag{4.44}$$

$$\mathrm{d}\alpha_- = \mathrm{d}P - \rho a\, \mathrm{d}\boldsymbol{U} \cdot \boldsymbol{n}, \tag{4.45}$$

$$\mathrm{d}\alpha_+ = \mathrm{d}P + \rho a\, \mathrm{d}\boldsymbol{U} \cdot \boldsymbol{n}, \tag{4.46}$$

$$\mathrm{d}\alpha_0 = \mathrm{d}\boldsymbol{U} \cdot \boldsymbol{t}, \tag{4.47}$$

where $\boldsymbol{t} = \boldsymbol{n}^{\perp}$. Recalling that for an isentropic flow the differential of the pressure expresses as $\mathrm{d}P = -\rho^2 a^2\, \mathrm{d}(\frac{1}{\rho})$, the Riemann invariants differentials definitions rewrite

$$\mathrm{d}\alpha_E = \mathrm{d}E - \boldsymbol{U} \cdot \mathrm{d}\boldsymbol{U} + P\, \mathrm{d}(\frac{1}{\rho}), \tag{4.48}$$

$$\mathrm{d}\alpha_- = \mathrm{d}(\frac{1}{\rho}) - \frac{1}{\rho a}\, \mathrm{d}\boldsymbol{U} \cdot \boldsymbol{n}, \tag{4.49}$$

$$\mathrm{d}\alpha_+ = \mathrm{d}(\frac{1}{\rho}) + \frac{1}{\rho a}\, \mathrm{d}\boldsymbol{U} \cdot \boldsymbol{n}, \tag{4.50}$$

$$\mathrm{d}\alpha_0 = \mathrm{d}\boldsymbol{U} \cdot \boldsymbol{t}. \tag{4.51}$$

Following the same procedure than in the one-dimensional case, we define the polynomial approximation of the Riemann invariants by linearizing equations (4.48), (4.49), (4.50) and (4.51) on each cells around the mean values in the cells

$$\alpha^c_{E,h} = \sum_{k=0}^{K} \alpha^c_{E,k} \sigma^c_k = E^c_h - \boldsymbol{U}^c_0 \cdot \boldsymbol{U}^c_h + P^c_0\, (\frac{1}{\rho})^c_h, \tag{4.52}$$

$$\alpha^c_{-,h} = \sum_{k=0}^{K} \alpha^c_{-,k} \sigma^c_k = (\frac{1}{\rho})^c_h - \frac{1}{Z_c}\, \boldsymbol{U}^c_h \cdot \boldsymbol{n}, \tag{4.53}$$

$$\alpha^c_{+,h} = \sum_{k=0}^{K} \alpha^c_{+,k} \sigma^c_k = (\frac{1}{\rho})^c_h + \frac{1}{Z_c}\, \boldsymbol{U}^c_h \cdot \boldsymbol{n}, \tag{4.54}$$

$$\alpha^c_{0,h} = \sum_{k=0}^{K} \alpha^c_{0,k} \sigma^c_k = \boldsymbol{U}^c_h \cdot \boldsymbol{t}, \tag{4.55}$$

where ϕ_h^i is the polynomial approximation of ϕ on the cell Ω_c and ϕ_0^c its mass averaged value. $Z_c = a_0^c \rho_0^c$ is the acoustic impedance. This procedure is equivalent to linearize the gas dynamics equations, on each cells, around a mean state. Applying the hierarchical high-order limitation procedure used in the previous cases, we obtain the limiting coefficients for the Riemann invariants polynomials. Hence, using definitions (4.52), (4.53), (4.54) and (4.55), we recover the limiting coefficients corresponding to the system variables polynomial approximations

$$
\begin{aligned}
&(\frac{1}{\rho})_k^c = \frac{1}{2}(\alpha_{+,k}^c + \alpha_{-,k}^c),\\
&\boldsymbol{U}_k^c = \frac{1}{2} Z_c (\alpha_{+,k}^c - \alpha_{-,k}^c)\boldsymbol{n} + \alpha_{0,k}^c \boldsymbol{t},\\
&E_k^c = \alpha_{E,k}^c + \frac{1}{2} Z_c (\alpha_{+,k}^c - \alpha_{-,k}^c)\boldsymbol{U}_0^c \cdot \boldsymbol{n} + \alpha_{0,k}^c \boldsymbol{U}_0^c \cdot \boldsymbol{t} - \frac{1}{2} P_0^c (\alpha_{+,k}^c + \alpha_{-,k}^c).
\end{aligned}
$$

Concerning the unit vector $\boldsymbol{n}$ and $\boldsymbol{t}$, we could define these projection vectors as the two orthogonal directions $\boldsymbol{e}_X$ and $\boldsymbol{e}_Y$. This option in the choice of $\boldsymbol{n}$ and $\boldsymbol{t}$ would not ensure symmetry preservation in the case of radial flows on polar grids. Finally, we decided to use the velocity mean value direction over the cell and its orthogonal vector, *i.e.*, $\boldsymbol{n} = \boldsymbol{U}_0^c \backslash ||\boldsymbol{U}_0^c||$ and $\boldsymbol{t} = (\boldsymbol{U}_0^c)^\perp \backslash ||\boldsymbol{U}_0^c||$. We numerically demonstrate in the numerical results Section 4.2.4 that this limitation procedure allows to preserve the cylindrical symmetry and of the apparition of most of the oscillations.

The discretization of the equations (4.2b), (4.2c) and (4.2d) by means of piecewise polynomials and of the deformation gradient tensor equation (3.10) by means of a mapping using Finite Element basis presented up to now is valid for any order of accuracy. From now on, we specify the proper implementation of our DG scheme in the two special cases of the second and third order of accuracy.

4.2 Second-order DG scheme

In this section, we present the second-order application of the general DG discretization presented previously. In this case, we consider that the continuous mapping characterizing the flow motion of the fluid is linear. Thus, the edges remain straight lines through the flow deformation. We start this part detailing the discretization of the deformation gradient tensor recalling that F would be piecewise constant in this second-order case.

4.2.1 Deformation gradient tensor discretization

The tensor F being piecewise constant over the triangles, we use the P_1 barycentric coordinate basis functions which write in a generic triangle $\mathcal{T}_i$

$$\lambda_p(\boldsymbol{X}) = \frac{1}{2|\mathcal{T}_i|}[X(Y_{p^+} - Y_{p^-}) - Y(X_{p^+} - X_{p^-}) + X_{p^+}Y_{p^-} - X_{p^-}Y_{p^+}], \tag{4.56}$$

where p, p^+ and p^- are the counterclockwise ordered triangle nodes and $|\mathcal{T}_i|$ the triangle volume, see Figure 4.2.

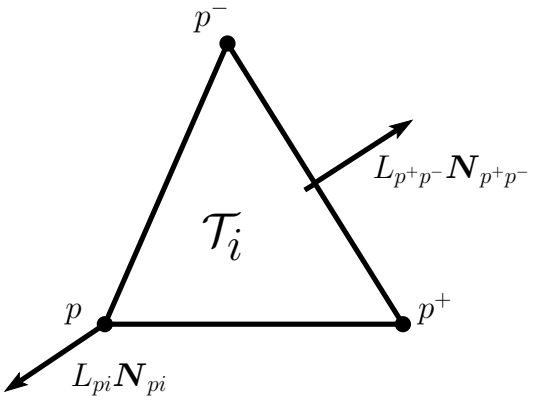


Figure 4.2: Generic triangle.

In this configuration, the gradient tensor definition (4.21) and the semi-discrete equation (4.26) rewrites

$$\mathsf{F}_i(t) = \frac{1}{|\mathcal{T}_i|} \sum_{p \in \mathcal{P}(\mathcal{T}_i)} \boldsymbol{x}_p(t) \otimes L_{pi}\boldsymbol{N}_{pi}, \tag{4.57}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathsf{F}_i(t) = \frac{1}{|\mathcal{T}_i|} \sum_{p \in \mathcal{P}(\mathcal{T}_i)} \boldsymbol{U}_p(t) \otimes L_{pi}\boldsymbol{N}_{pi}, \tag{4.58}$$

where $\mathcal{P}(\mathcal{T}_i)$ is the node set of $\mathcal{T}_i$, and $L_{pi}\boldsymbol{N}_{pi}$ is the corner normal at node p in the initial configuration, see Figure 4.2, since

$$\begin{aligned}
\boldsymbol{\nabla}_X \lambda_p(\boldsymbol{X}) &= \frac{1}{2|\mathcal{T}_i|} \begin{pmatrix} Y_{p^+} - Y_{p^-} \\ X_{p^-} - X_{p^+} \end{pmatrix}, \\
&= \frac{1}{2|\mathcal{T}_i|}(L_{p^-p}\boldsymbol{N}_{p^-p} + L_{pp^+}\boldsymbol{N}_{pp^+}), \\
&= \frac{L_{pi}\boldsymbol{N}_{pi}}{|\mathcal{T}_i|}.
\end{aligned} \tag{4.59}$$

This form of deformation gradient tensor discretization has also been presented by G. Kluth and B. Després in [56]. The difference lies in the reference configuration. In [56], they developed a cell-centered Lagrangian scheme on moving mesh for the hyperelasticity equations. Thus, they compute the flow deformation from the previous time step geometry. In our case, the whole scheme is developed on the initial configuration. Consequently, the flow deformation is computed from the initial geometry.

Recalling that the integral of the unit outward normal over a closed surface is equal to zero, the relation $L_{p^-p}\boldsymbol{N}_{p^-p} + L_{pp^+}\boldsymbol{N}_{pp^+} + L_{p^+p^-}\boldsymbol{N}_{p^+p^-} = \boldsymbol{0}$, holds in the case of a triangular cell. Hence, a new relation on the corner normals immediately comes

$$\begin{aligned} L_{pi}\boldsymbol{N}_{pi} &= \frac{1}{2}\left(L_{p^-p}\boldsymbol{N}_{p^-p} + L_{pp^+}\boldsymbol{N}_{pp^+}\right), \\ &= -\frac{1}{2}L_{p^+p^-}\boldsymbol{N}_{p^+p^-}. \end{aligned} \tag{4.60}$$

This relation means that in a triangular cell the corner normal at a specific node is colinear with the opposite face normal. A major problem inherent to the deformation gradient tensor discretization lies in the continuity of the geometry. We made the choice of discretizing the deformation gradient tensor by means of a mapping using Finite Element basis to assure continuity of the normal $\mathsf{G}\boldsymbol{N}$ through the flow deformation. This continuity property as well as the respect of the Piola condition come from the mapping discretization using the P_1 barycentric coordinate basis functions. Now, we can ensure this result from the discretized form of the deformation gradient tensor. Obviously, regarding (4.57) tensor F is piecewise constant on the triangular cells over the domain. Hence F is discontinuous. Nevertheless, $\mathsf{F}\boldsymbol{T}\mathrm{d}L$ and $\mathsf{G}\boldsymbol{N}\mathrm{d}L$, corresponding to $\boldsymbol{t}\mathrm{d}l$ and $\boldsymbol{n}\mathrm{d}l$ the Eulerian tangent and normal, have to be continuous on the boundaries of the triangular cells.

Continuity of the normals. Let $\mathcal{T}_L$ and $\mathcal{T}_R$ be two neighbor triangles sharing the face f_{pp^+}, refer to Figure 4.3. A correct discretization of the deformation tensor would lead to have the normal and tangent continuity on the face ,*i.e.*, $\mathsf{F}_L\, L_{pp^+}\boldsymbol{T}_{pp^+} = \mathsf{F}_R\, L_{pp^+}\boldsymbol{T}_{pp^+} = l_{pp^+}\boldsymbol{t}_{pp^+}$.

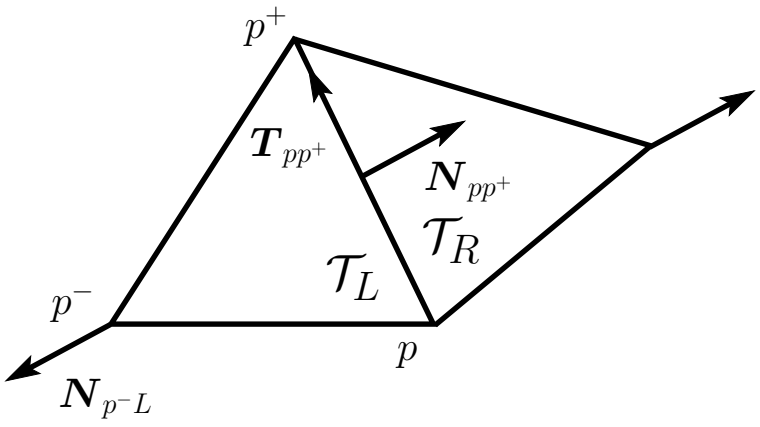


Figure 4.3: Neighbor triangles.

Considering the triangle $\mathcal{T}_L$, we know that $\boldsymbol{T}_{pp^+}$ is orthogonal to $\boldsymbol{N}_{pp^+}$ and consequently to $\boldsymbol{N}_{p^-L}$. This relation means that in the computation of the Eulerian normal of a mesh face, there is

no contribution of the opposite node. Hence, the transported Lagrangian normal $\mathsf{F}_L\, L_{pp^+}\boldsymbol{T}_{pp^+}$ writes

$$\begin{aligned}\mathsf{F}_L\, L_{pp^+}\boldsymbol{T}_{pp^+} &= \frac{1}{|\mathcal{T}_L|}\sum_{q\in\mathcal{P}(\mathcal{T}_L)} \boldsymbol{\Phi}_q\left(L_{qL}\boldsymbol{N}_{qL}\,\boldsymbol{.}\,L_{pp^+}\boldsymbol{T}_{pp^+}\right),\\ &= \frac{1}{|\mathcal{T}_L|}\left(\boldsymbol{\Phi}_p\left(L_{pL}\boldsymbol{N}_{pL}\,\boldsymbol{.}\,L_{pp^+}\boldsymbol{T}_{pp^+}\right)+\boldsymbol{\Phi}_{p^+}\left(L_{p^+L}\boldsymbol{N}_{p^+L}\,\boldsymbol{.}\,L_{pp^+}\boldsymbol{T}_{pp^+}\right)\right). \end{aligned}\tag{4.61}$$

Recalling the corner normals definitions $L_{pL}\boldsymbol{N}_{pL}=\frac{1}{2}(L_{p^-p}\boldsymbol{N}_{p^-p}+L_{pp^+}\boldsymbol{L}_{pp^+})$ and $L_{p^+L}\boldsymbol{N}_{p^+L}=\frac{1}{2}(L_{pp^+}\boldsymbol{N}_{pp^+}+L_{p^+p^-}\boldsymbol{L}_{p^+p^-})$, expression (4.61) rewrites

$$\begin{aligned}\mathsf{F}_L\, L_{pp^+}\boldsymbol{T}_{pp^+} &= \frac{1}{2|\mathcal{T}_L|}\left(\boldsymbol{\Phi}_p\left(L_{p^-p}\boldsymbol{N}_{p^-p}\,\boldsymbol{.}\,L_{pp^+}\boldsymbol{T}_{pp^+}\right)+\boldsymbol{\Phi}_{p^+}\left(L_{p^+p^-}\boldsymbol{N}_{p^+p^-}\,\boldsymbol{.}\,L_{pp^+}\boldsymbol{T}_{pp^+}\right)\right),\\ &= \boldsymbol{\Phi}_{p^+}-\boldsymbol{\Phi}_p=\boldsymbol{x}_{p^+}-\boldsymbol{x}_p=l_{pp^+}\boldsymbol{t}_{pp^+}. \end{aligned}\tag{4.62}$$

Again, it confirms that the initial tangent transported through the flow deformation definitively corresponds to the tangent into the actual configuration. Obviously, regarding (4.28), this relation holds considering the normals. The use of a similar process in the neighbor triangle $\mathcal{T}_R$ leads to an identical result. Consequently, we do have the required continuity

$$\mathsf{F}_L\, L_{pp^+}\boldsymbol{T}_{pp^+}=\mathsf{F}_R\, L_{pp^+}\boldsymbol{T}_{pp^+}=l_{pp^+}\boldsymbol{t}_{pp^+}.\tag{4.63}$$

Deformation gradient tensor initialization. Concerning the initialization of the deformation gradient tensor, one knows that initially, $\boldsymbol{\Phi}(\boldsymbol{X},0)=\boldsymbol{X}$ and $\mathsf{F}(\boldsymbol{X},0)=\mathsf{I_d}$. Let us recall the geometrical identity (3.52) in the particular case of $t=0$

$$\sum_{p\in\mathcal{P}(\mathcal{T}_i)} L_{pi}(\boldsymbol{N}_{pi}\otimes\boldsymbol{X}_p)=|\mathcal{T}_i|\mathsf{I_d}.$$

Recalling the definition of the discretized deformation gradient tensor (4.57), it immediately comes that

$$\mathsf{F}_i(0)=\frac{1}{|\mathcal{T}_i|}\sum_{p\in\mathcal{P}(\mathcal{T}_i)}\boldsymbol{X}_p\otimes L_{pi}\boldsymbol{N}_{pi}=\frac{|\mathcal{T}_i|}{|\mathcal{T}_i|}\mathsf{I_d}=\mathsf{I_d}.$$

This result shows that the deformation gradient tensor discretization ensures a correct initialization of the tensor. Now, with the help of (3.11) we want to derive a semi-discrete equation on the Jacobian.

Compatibility of tensor $\mathsf{F_i}$ discretization. Actually, substituting the tensor $\mathsf{F_i}$ semi-discrete equation (4.58) into $\frac{\mathrm{d}\,J_i}{\mathrm{d}t}=\mathsf{G}_i:\frac{\mathrm{d}\,\mathsf{F}_i}{\mathrm{d}t}$ yields

$$\begin{aligned}\frac{\mathrm{d}\,J_i}{\mathrm{d}t} &= \frac{1}{|\mathcal{T}_i|}\sum_{p\in\mathcal{P}(\mathcal{T}_i)}\mathsf{G}_i:(\boldsymbol{U}_p\otimes L_{pi}\boldsymbol{N}_{pi}),\\ &= \frac{1}{|\mathcal{T}_i|}\sum_{p\in\mathcal{P}(\mathcal{T}_i)}\boldsymbol{U}_p\,\boldsymbol{.}\,\mathsf{G}_i L_{pi}\boldsymbol{N}_{pi},\quad\text{thanks to (A.18c)}\end{aligned}$$

where J_i is the constant Jacobian over the triangle $\mathcal{T}_i$. At the end, reminding the previous relations (4.28) and (4.63), the Jacobian semi-discrete equation writes

$$\frac{\mathrm{d}\,J_i}{\mathrm{d}t}=\frac{1}{|\mathcal{T}_i|}\sum_{p\in\mathcal{P}(\mathcal{T}_i)}\boldsymbol{U}_p\,\boldsymbol{.}\,l_{pi}\boldsymbol{n}_{pi}=\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{|\tau_i|}{|\mathcal{T}_i|}\right),\tag{4.64}$$

where $|\tau_i|$ is the volume of the triangle τ_i, which corresponds to the transport of $\mathcal{T}_i$ trough the flow motion of the fluid. We will see in the next Section 4.2.2 that considering (4.64) the discretization of F is consistent with the discretization of the volume equation.

4.2.2 Variational formulations

In the previous general framework section, we make explicit the variational formulation of the system equations required by the DG discretization. As we already said, the interior terms are evaluated by means of the quadrature rule defined in Table 4.1. Here, we detail the specific treatment of the boundary terms ensuring the respect of the GCL condition.

Specific volume equation discretization. Let us recall the variational formulation over the cell Ω_c of the specific volume equation

$$\begin{aligned}\int_{\Omega_c} \rho^0 \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})\sigma_q^c \,\mathrm{d}V &= \sum_{k=0}^{K} \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})_k^c \int_{\Omega_c} \rho^0 \sigma_q^c \sigma_k^c \,\mathrm{d}V, \\ &= -\int_{\Omega_c} \boldsymbol{U} \cdot \mathsf{G}\, \boldsymbol{\nabla}_X \sigma_q^c \,\mathrm{d}V + \int_{\partial\Omega_c} \overline{\boldsymbol{U}} \cdot \sigma_q^c \,\mathsf{G}\boldsymbol{N}\mathrm{d}L, \end{aligned} \tag{4.65}$$

With the chosen polynomial basis (4.9) presented, the first moment of the specific volume reads

$$(\frac{1}{\rho})_0^c = \frac{1}{m_c}\int_{\Omega_c} \frac{\rho^0}{\rho}\,\mathrm{d}V = \frac{1}{m_c}\int_{\Omega_c} J\mathrm{d}V = \frac{1}{m_c}\int_{\omega_c} \mathrm{d}v = \frac{|\omega_c|}{m_c} = \frac{1}{\rho_c},$$

where ρ_c is the density mean value defined in EUCCLHYD scheme Section 1.2. This relation of the first moment $(\frac{1}{\rho})_0^c$ of the specific volume rewrites

$$(\frac{1}{\rho})_0^c = \frac{|\omega_c|}{m_c} = \frac{1}{m_c}\sum_{i=1}^{ntri} |\tau_i^c|,$$

where $|\tau_i^c|$ is the new volume at time t of the triangle in the initial configuration $\mathcal{T}_i^c$. Now, with a proper time integration of the Jacobian equation (4.64), the previous relation on $(\frac{1}{\rho})_0^c$ yields

$$(\frac{1}{\rho})_0^c = \frac{|\omega_c|}{m_c} = \frac{1}{m_c}\sum_{i=1}^{ntri} |\mathcal{T}_i^c|\, J_i^c. \tag{4.66}$$

This identity ensures the compatibility between the geometric discretization and the DG resolution of the specific volume.

For the second-order scheme the deformation gradient tensor F and so G are piecewise constants over the triangles $\mathcal{T}_i^c$, and $\boldsymbol{\nabla}_X \sigma_q^c$ are piecewise constants over the polygonal cells Ω_c. Thus, the previous equation (4.65) rewrite

$$\int_{\Omega_c} \rho^0 \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})\sigma_q^c \,\mathrm{d}V = -\sum_{i=1}^{ntri} \mathsf{G}_i^c\, \boldsymbol{\nabla}_X \sigma_q^c \cdot \int_{\mathcal{T}_i^c} \boldsymbol{U}\mathrm{d}V + \int_{\partial\Omega_c} \overline{\boldsymbol{U}} \cdot \sigma_q^c \,\mathsf{G}\boldsymbol{N}\mathrm{d}L,$$

Here, the main difficulty lies in the evaluation of the boundary term. We know that in second-order scheme case the mapping characterizing the flow motion of the fluid is linear over the domain. Consequently, we define a parametric representation of the edge velocity which is

consistent with the Finite Element mapping. Then, the continuous velocity $\overline{\boldsymbol{U}}$ reads as follows on face f_{pp^+}

$$\overline{\boldsymbol{U}}_{|_{pp^+}}(\zeta) = \boldsymbol{U}_p\,(1-\zeta) + \boldsymbol{U}_{p^+}\,\zeta, \tag{4.67}$$

where $\zeta \in [0,1]$ is the linear abscissa, $\boldsymbol{U}_p$ and $\boldsymbol{U}_{p^+}$ are the velocity of the nodes p and p^+ initially located at $\boldsymbol{X}_p$ and $\boldsymbol{X}_{p^+}$. Here, the linear functions $1-\zeta$ and ζ correspond to the trace of the P_1 barycentric coordinate functions (4.56) on the triangle edges.

In the second-order case, the polynomial basis functions σ_q^c are piecewise linear over the polygonal cells and thus piecewise linear on both sides of the mesh edges. Consequently, on the face f_{pp^+} the function σ_q^c can be defined by means of the linear abscissa and of its extrapolated values at the nodes p and p^+ as

$$\sigma_{q|_{pp^+}}^c(\zeta) = \sigma_q^c(\boldsymbol{X}_p)\,(1-\zeta) + \sigma_q^c(\boldsymbol{X}_{p^+})\,\zeta. \tag{4.68}$$

Combining the two definitions (4.67) and (4.68), we are able to express on face f_{pp^+} the product $\overline{\boldsymbol{U}}\sigma_q^c$ as

$$\overline{\boldsymbol{U}}\sigma_{q|_{pp^+}}^c(\zeta) = \boldsymbol{U}_p\sigma_q^c(\boldsymbol{X}_p)\,(1-\zeta)^2 + \boldsymbol{U}_{p^+}\sigma_q^c(\boldsymbol{X}_{p^+})\,\zeta^2 + (\boldsymbol{U}_{p^+}\sigma_q^c(\boldsymbol{X}_p) + \boldsymbol{U}_p\sigma_q^c(\boldsymbol{X}_{p^+}))\,(1-\zeta)\zeta. \tag{4.69}$$

Now, to evaluate the boundary term in (4.65), we analytically integrate the function defined in (4.69)

$$\begin{aligned}\int_{\partial\Omega_c} \overline{\boldsymbol{U}}\sigma_q^c \cdot \mathsf{G}\boldsymbol{N}\,\mathrm{d}L &= \sum_{p\in\mathcal{P}(c)} \int_p^{p^+} \overline{\boldsymbol{U}}\sigma_q^c \cdot \mathsf{G}\boldsymbol{N}\,\mathrm{d}L,\\ &= \sum_{p\in\mathcal{P}(c)} \left(\int_0^1 \overline{\boldsymbol{U}}\sigma_{q|_{pp^+}}^c(\zeta)\,\mathrm{d}\zeta\right) \cdot \mathsf{G}_{|_{pp^+}} L_{pp^+}\boldsymbol{N}_{pp^+},\end{aligned} \tag{4.70}$$

where $\mathsf{G}_{|_{pp^+}}$ is the constant value of the cofactor tensor G on face f_{pp^+}. Thanks to the discretization of the deformation gradient tensor described previously, we know that $\mathsf{G}_{|_{pp^+}} L_{pp^+}\boldsymbol{N}_{pp^+}$ strictly corresponds to the Eulerian normal $l_{pp^+}\boldsymbol{n}_{pp^+}$ in the actual configuration. Thus, analytically integrating equation (4.70), one gets

$$\begin{aligned}\int_{\partial\Omega_c} \overline{\boldsymbol{U}}\sigma_q^c \cdot \mathsf{G}\boldsymbol{N}\,\mathrm{d}L &= \sum_{p\in\mathcal{P}(c)} \left(\int_0^1 \overline{\boldsymbol{U}}\sigma_{q|_{pp^+}}^c(\zeta)\,\mathrm{d}\zeta\right) \cdot l_{pp^+}\boldsymbol{n}_{pp^+},\\ &= \sum_{p\in\mathcal{P}(c)} \frac{1}{6}[\boldsymbol{U}_p \cdot (2\sigma_q^c(\boldsymbol{X}_p) + \sigma_q^c(\boldsymbol{X}_{p^+}))l_{pp^+}\boldsymbol{n}_{pp^+}\\ &\qquad + \boldsymbol{U}_{p^+} \cdot (2\sigma_q^c(\boldsymbol{X}_{p^+}) + \sigma_q^c(\boldsymbol{X}_p))l_{pp^+}\boldsymbol{n}_{pp^+}].\end{aligned}$$

Finally, performing an index permutation on this last equation leads to

$$\int_{\partial\Omega_c} \overline{\boldsymbol{U}}\sigma_q^c \cdot \mathsf{G}\boldsymbol{N}\,\mathrm{d}L = \sum_{p\in\mathcal{P}(c)} \boldsymbol{U}_p \cdot [\frac{1}{3}(2\sigma_q^c(\boldsymbol{X}_p) + \sigma_q^c(\boldsymbol{X}_{p^+}))l_{pc}^+\boldsymbol{n}_{pc}^+ + \frac{1}{3}(2\sigma_q^c(\boldsymbol{X}_p) + \sigma_q^c(\boldsymbol{X}_{p^-}))l_{pc}^-\boldsymbol{n}_{pc}^-],$$

where $l_{pc}^-\boldsymbol{n}_{pc}^- = \frac{1}{2}l_{p^-p}\boldsymbol{n}_{p^-p}$ and $l_{pc}^+\boldsymbol{n}_{pc}^+ = \frac{1}{2}l_{pp^+}\boldsymbol{n}_{pp^+}$ are the half-left and half-right Eulerian corner normals. Now, for the sake of conciseness in the development of our DG scheme, let us

introduce the weighted lengths $l_{pc}^{\pm,q}$, balanced by the basis functions extrapolations and defined as

$$l_{pc}^{\pm,q} = \left(\frac{2\sigma_q^c(\boldsymbol{X}_p) + \sigma_q^c(\boldsymbol{X}_{p^\pm})}{3} \right) l_{pc}^{\pm}. \tag{4.71}$$

Hence, the boundary term of the equation (4.65) finally writes

$$\int_{\partial\Omega_c} \overline{\boldsymbol{U}} \sigma_q^c \,.\, \mathsf{G}\boldsymbol{N} \mathrm{d}L = \sum_{p\in\mathcal{P}(c)} \boldsymbol{U}_p \,.\, l_{pc}^q \boldsymbol{n}_{pc}^q, \tag{4.72}$$

where $l_{pc}^q \boldsymbol{n}_{pc}^q = l_{pc}^{-,q} \boldsymbol{n}_{pc}^- + l_{pc}^{+,q} \boldsymbol{n}_{pc}^+$. The first moment of this relation, *i.e.*, $q = 0$, confirms the respect of the GCL condition which reads

$$\int_{\partial\Omega_c} \overline{\boldsymbol{U}} \,.\, \mathsf{G}\boldsymbol{N} \mathrm{d}L = \int_{\partial\omega_c} \overline{\boldsymbol{U}} \,.\, \boldsymbol{n} \mathrm{d}l = \sum_{p\in\mathcal{P}(c)} \boldsymbol{U}_p \,.\, l_{pc} \boldsymbol{n}_{pc}.$$

Finally, we have derived the semi-discrete equations on the specific volume successive moments, in respect with the GCL and with an analytical integration of the boundary term. The semi-discrete equations writes

$$\int_{\Omega_c} \rho^0 \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho}) \sigma_q^c \,\mathrm{d}V = -\sum_{i=1}^{ntri} \mathsf{G}_i^c \boldsymbol{\nabla}_X \sigma_q^c \,.\, \int_{\mathcal{T}_i^c} \boldsymbol{U} \mathrm{d}V + \sum_{p\in\mathcal{P}(c)} \boldsymbol{U}_p \,.\, l_{pc}^q \boldsymbol{n}_{pc}^q. \tag{4.73}$$

For the first moment, we exactly recover the EUCCLHYD scheme presented in Section 3.2

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})_0^c = \sum_{p\in\mathcal{P}(c)} \boldsymbol{U}_p \,.\, l_{pc} \boldsymbol{n}_{pc}, \tag{4.74}$$

where $l_{pc}\boldsymbol{n}_{pc} = l_{pc}^0 \boldsymbol{n}_{pc}^0$.
For the momentum discretization, we use a similar procedure as the one presented for the specific volume equation, with a noteworthy difference on the property of the pressure numerical flux.

Momentum discretization. For the momentum equation, the local variational formulation of (4.2c) on Ω_c leads to

$$\int_{\Omega_c} \rho^0 \frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t} \sigma_q^c \,\mathrm{d}V = \sum_{i=1}^{ntri} \mathsf{G}_i^c \boldsymbol{\nabla}_X \sigma_q^c \int_{\mathcal{T}_i^c} P \mathrm{d}V - \int_{\partial\Omega_c} \overline{P} \sigma_q^c \, \mathsf{G}\boldsymbol{N} \mathrm{d}L. \tag{4.75}$$

To evaluate the boundary term of (4.75), we use a similar procedure as one presented for the specific volume equation, considering that the pressure numerical flux $\overline{P}$ is piecewise linear on the edges of the cell Ω_c. For the edge velocity $\overline{\boldsymbol{U}}$, we imposed this quantity to be continuous between cells and to be single valued at the nodes of the mesh. This hypothesis is strongly needed, the velocity being a kinematic variable deriving from the continuous mapping. But this time, the pressure being a thermodynamical variable, nothing impose to $\overline{P}$ to be continuous between neighboring cells. Thus, we generalize the study setting the edge pressure in the cell Ω_c as

$$\overline{P}\big|_{pp^+}^c(\zeta) = P_{pc}^+ (1-\zeta) + P_{p^+c}^- \zeta, \tag{4.76}$$

where P^+_{pc} and $P^-_{p^+c}$ are respectively the right and left nodal pressures in the cell Ω_c at the nodes p and p^+. These nodal pressures correspond to the Π^+_{pc} and $\Pi^-_{p^+c}$ introduced in Section 3.3. After an analytical integration, the boundary term of (4.75) writes

$$\int_{\partial\Omega_c} \overline{P}\sigma_q^c\, \mathsf{G}\boldsymbol{N}\mathrm{d}L = \sum_{p\in\mathcal{P}(c)} \left(P^-_{pc}\, l^{-,q}_{pc}\boldsymbol{n}^-_{pc} + P^+_{pc}\, l^{+,q}_{pc}\boldsymbol{n}^+_{pc}\right). \tag{4.77}$$

Finally, we have derived the semi-discrete equations on the successive moments of the velocity, with an analytical integration of the boundary term. The semi-discrete equations writes

$$\int_{\Omega_c} \rho^0 \frac{\mathrm{d}\,\boldsymbol{U}}{\mathrm{d}t}\sigma_q^c\,\mathrm{d}V = \sum_{i=1}^{ntri} \mathsf{G}_i^c \boldsymbol{\nabla}_X \sigma_q^c \int_{\mathcal{T}_i^c} P\mathrm{d}V - \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}^q_{pc}, \tag{4.78}$$

where $\boldsymbol{F}^q_{pc} = P^-_{pc}\, l^{-,q}_{pc}\boldsymbol{n}^-_{pc} + P^+_{pc}\, l^{+,q}_{pc}\boldsymbol{n}^+_{pc}$ identifies with the qth moment of the subcell forces. For the first moment, corresponding to $q=0$, we again recover the EUCCLHYD scheme

$$m_c \frac{\mathrm{d}\,\boldsymbol{U}^c_0}{\mathrm{d}t} = -\sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{pc}, \tag{4.79}$$

where $\boldsymbol{F}_{pc} = \boldsymbol{F}^0_{pc}$.

Total energy discretization. The procedure to discretize the total energy equation (4.2d) is identical as the one used for the momentum equation. Let us recall the local variational formulation of (4.2d) on Ω_c

$$\int_{\Omega_c} \rho^0 \frac{\mathrm{d}\,E}{\mathrm{d}t}\sigma_q^c\,\mathrm{d}V = \sum_{i=1}^{ntri} \mathsf{G}_i^c \boldsymbol{\nabla}_X \sigma_q^c \,.\int_{\mathcal{T}_i^c} P\,\boldsymbol{U}\mathrm{d}V - \int_{\partial\Omega_c} \overline{P\boldsymbol{U}}\,.\,\sigma_q^c \mathsf{G}\boldsymbol{N}\mathrm{d}L. \tag{4.80}$$

At this point, we assume that the numerical flux inherent to the total energy equation, $\overline{P\boldsymbol{U}}$, has the same linear property than the other numerical fluxes $\overline{\boldsymbol{U}}$ and $\overline{P}$. Thus, we consider that the polynomial degree of $\overline{P\boldsymbol{U}}$ is equivalent to the one of the total energy polynomial approximation E_h. Following the procedure used in the momentum equation discretization, the function $\overline{P\boldsymbol{U}}$ writes on the face f_{pp^+} as

$$\overline{P\boldsymbol{U}}^c_{|_{pp^+}}(\zeta) = (P\boldsymbol{U})^+_{pc}\,(1-\zeta) + (P\boldsymbol{U})^-_{p^+c}\,\zeta. \tag{4.81}$$

The analytical integration of the expression in the evaluation of the boundary term of (4.144) leads to the following semi-discrete equations for the total energy

$$\int_{\Omega_c} \rho^0 \frac{\mathrm{d}\,E}{\mathrm{d}t}\sigma_q^c\,\mathrm{d}V = \sum_{i=1}^{ntri} \mathsf{G}_i^c \boldsymbol{\nabla}_X \sigma_q^c \,.\int_{\mathcal{T}_i^c} P\,\boldsymbol{U}\mathrm{d}V - \sum_{p\in\mathcal{P}(c)} \left((P\boldsymbol{U})^-_{pc}\,.\,l^{-,q}_{pc}\boldsymbol{n}^-_{pc} + (P\boldsymbol{U})^+_{pc}\,.\,l^{+,q}_{pc}\boldsymbol{n}^+_{pc}\right). \tag{4.82}$$

At this point, we make the same fundamental assumption on the numerical flux $\overline{P\boldsymbol{U}}$ than the one used in the semi-discretization of the local entropy inequality (4.33)

$$\overline{P\boldsymbol{U}} = \overline{P}\;\overline{\boldsymbol{U}}. \tag{4.83}$$

This relation immediately yields

$$(P\boldsymbol{U})^-_{pc} = P^-_{pc}\,\boldsymbol{U}_p \text{ and } (P\boldsymbol{U})^+_{pc} = P^+_{pc}\,\boldsymbol{U}_p. \tag{4.84}$$

With the use of this fundamental assumption, the semi-discrete equations on the moments of the total energy finally write

$$\int_{\Omega_c} \rho^0 \frac{\mathrm{d}\, E}{\mathrm{d}t} \sigma_q^c \,\mathrm{d}V = \sum_{i=1}^{ntri} \mathsf{G}_i^c \boldsymbol{\nabla}_X \sigma_q^c \,.\, \int_{\mathcal{T}_i^c} P\, \boldsymbol{U} \mathrm{d}V - \sum_{p\in\mathcal{P}(c)} \boldsymbol{U}_p \,.\, \boldsymbol{F}_{pc}^q \tag{4.85}$$

For the first moment, we can identify the EUCCLHYD scheme

$$m_c \frac{\mathrm{d}\, E_0^c}{\mathrm{d}t} = - \sum_{p\in\mathcal{P}(c)} \boldsymbol{U}_p \,.\, \boldsymbol{F}_{pc}. \tag{4.86}$$

Assumption (4.83) allows us to uncouple the different nodes contributions. This is the only solution to recover the EUCCLHYD for the first moments of these DG polynomial discretizations, and also to avoid the need for a global linear system on the nodal velocity. That is we keep local nodal solvers. In the end, we have $3 \times (K+1)$ semi-discrete equations for the same number of unknowns. For $q = 0 \dots K$

$$\begin{aligned}
&\sum_{k=0}^{K} \frac{\mathrm{d}}{\mathrm{d}t} (\frac{1}{\rho})_k \int_{\Omega_c} \rho^0 \sigma_q^c \sigma_k^c \,\mathrm{d}V = - \sum_{i=1}^{ntri} \mathsf{G}_i^c \boldsymbol{\nabla}_X \sigma_q^c \,.\, \int_{\mathcal{T}_i^c} \boldsymbol{U} \mathrm{d}V + \sum_{p\in\mathcal{P}(c)} \boldsymbol{U}_p \,.\, l_{pc}^q \boldsymbol{n}_{pc}^q, \\
&\sum_{k=0}^{K} \frac{\mathrm{d}\, \boldsymbol{U}_k}{\mathrm{d}t} \int_{\Omega_c} \rho^0 \sigma_q^c \sigma_k^c \,\mathrm{d}V = \sum_{i=1}^{ntri} \mathsf{G}_i^c \boldsymbol{\nabla}_X \sigma_q^c \int_{\mathcal{T}_i^c} P \mathrm{d}V - \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{pc}^q, \\
&\sum_{k=0}^{K} \frac{\mathrm{d}\, E_k}{\mathrm{d}t} \int_{\Omega_c} \rho^0 \sigma_q^c \sigma_k^c \,\mathrm{d}V = \sum_{i=1}^{ntri} \mathsf{G}_i^c \boldsymbol{\nabla}_X \sigma_q^c \,.\, \int_{\mathcal{T}_i^c} P\, \boldsymbol{U} \mathrm{d}V - \sum_{p\in\mathcal{P}(c)} \boldsymbol{U}_p \,.\, \boldsymbol{F}_{pc}^q.
\end{aligned}$$

To complete the scheme construction, we need to define the nodal solvers to compute $\boldsymbol{U}_p$ and $\boldsymbol{F}_{pc}^q$. For that purpose, we make use of the entropy analysis done in the general framework Section 4.1.3, to ensure that the kinetic energy is correctly dissipated in internal energy through a shock.

4.2.3 Nodal solvers

The entropic analysis done in the previous general framework section gives us a specific relation between the numerical fluxes $\overline{P}$ and $\boldsymbol{U}$ ensuring a local entropy inequality. Let us recall this fundamental identity on the boundaries of cell Ω_c

$$\overline{P} = P_h^c - Z_c \,(\overline{\boldsymbol{U}} - \boldsymbol{U}_h^c) \,.\, \boldsymbol{n}, \tag{4.87}$$

where $Z_c = \rho_c \, a_c$ is the acoustic impedance. The use of this expression to calculate $\boldsymbol{F}_{pc}^q$ yields

$$\begin{aligned}
\boldsymbol{F}_{pc}^q &= P_{pc}^- \, l_{pc}^{-,q} \boldsymbol{n}_{pc}^- + P_{pc}^+ \, l_{pc}^{+,q} \boldsymbol{n}_{pc}^+, \\
&= P_h^c(\boldsymbol{X}, t) \, l_{pc}^q \boldsymbol{n}_{pc}^q \quad - Z_c \left[(\boldsymbol{U}_p - \boldsymbol{U}_h^c(\boldsymbol{X}_p, t)) \,.\, \boldsymbol{n}_{pc}^- \right] l_{pc}^{-,q} \boldsymbol{n}_{pc}^- \\
&\qquad\qquad\quad - Z_c \left[(\boldsymbol{U}_p - \boldsymbol{U}_h^c(\boldsymbol{X}_p, t)) \,.\, \boldsymbol{n}_{pc}^+ \right] l_{pc}^{+,q} \boldsymbol{n}_{pc}^+.
\end{aligned}$$

Finally, the qth moment of the subcell force writes

$$\boldsymbol{F}_{pc}^q = P_h^c(\boldsymbol{X}_p, t) \, l_{pc}^q \boldsymbol{n}_{pc}^q - \mathsf{M}_{pc}^q \,(\boldsymbol{U}_p - \boldsymbol{U}_h^c(\boldsymbol{X}_p, t)), \tag{4.88}$$

where the M_{pc}^{q} matrices are defined as

$$\mathsf{M}_{pc}^{q} = Z_c \left(l_{pc}^{-,q} \boldsymbol{n}_{pc}^{-} \otimes \boldsymbol{n}_{pc}^{-} + l_{pc}^{+,q} \boldsymbol{n}_{pc}^{+} \otimes \boldsymbol{n}_{pc}^{+} \right).$$

Similarly to the EUCCLHYD scheme development, refer to Section 3.3.6, the total energy conservation imposes, in the simple case with no boundary contribution that

$$\begin{aligned}
\sum_c \int_{\Omega_c} \rho^0 \frac{\mathrm{d}\, E}{\mathrm{d}t} \,\mathrm{d}V &= -\sum_c \sum_{p \in \mathcal{P}(c)} \boldsymbol{U}_p \cdot \boldsymbol{F}_{pc}, \\
&= \sum_p \boldsymbol{U}_p \cdot \sum_{c \in \mathcal{C}(p)} \boldsymbol{F}_{pc} \\
&= 0,
\end{aligned}$$

where $\mathcal{C}(p)$ is the set of cells surrounding the p node. Since this result holds for any velocity field, total energy conservation implies

$$\sum_{c \in \mathcal{C}(p)} \boldsymbol{F}_{pc} = \boldsymbol{0}. \tag{4.89}$$

This constraint also enforce the momentum conservation because

$$\begin{aligned}
\sum_c \int_{\Omega_c} \rho^0 \frac{\mathrm{d}\, \boldsymbol{U}}{\mathrm{d}t} \,\mathrm{d}V &= -\sum_c \sum_{p \in \mathcal{P}(c)} \boldsymbol{F}_{pc}, \\
&= \sum_{p \in \mathcal{P}(\Omega)} \sum_{c \in \mathcal{C}(p)} \boldsymbol{F}_{pc} \\
&= \boldsymbol{0}.
\end{aligned}$$

Thanks to (4.89), we finally have an explicit expression of the nodal velocity $\boldsymbol{U}_p$

$$\Big(\sum_{c \in \mathcal{C}(p)} M_{pc} \Big) \boldsymbol{U}_p = \sum_{c \in \mathcal{C}(p)} \left[P_h^c(\boldsymbol{X}_p, t)\, l_{pc} \boldsymbol{n}_{pc} + \mathsf{M}_{pc}\, \boldsymbol{U}_h^c(\boldsymbol{X}_p, t) \right], \tag{4.90}$$

where $\mathsf{M}_{pc} = Z_c \left(l_{pc}^{+} \boldsymbol{n}_{pc}^{+} \otimes \boldsymbol{n}_{pc}^{+} + l_{pc}^{-} \boldsymbol{n}_{pc}^{-} \otimes \boldsymbol{n}_{pc}^{-} \right)$ are positive semi-definite matrices with a physical dimension of a density times a velocity. This node velocity definition is nothing but the nodal solver derived in the previous EUCCLHYD Section 3.3. Thus, the boundary conditions are identical to the one used in the first-order scheme.

In the end, we obtain a second-order semi-discrete scheme respecting by construction the Piola compatibility condition and assuring the GCL, the momentum conservation and the total energy conservation. Finally, we apply to all these semi-discretized equations a classical second-order TVD Runge-Kutta time discretization scheme [27]. The time step is evaluated following the same criteria than in the first-order Section 3.3.8. The choice of the coefficient C_e is constrained by the condition introduced by B. Cockburn and C.-W. Shu in [27]. Thus, in this second-order frame, C_e reads

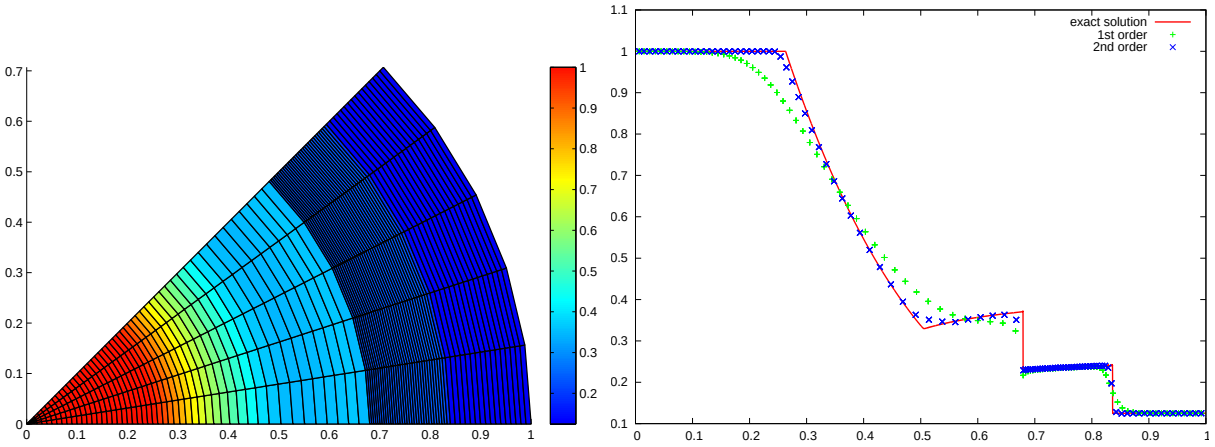
$$C_e = \frac{1}{2\alpha + 1} = \frac{1}{3}.$$

Here, the second-order DG scheme robustness and accuracy will be assessed using several relevant test cases taken from the literature.

4.2.4 Numerical results

To demonstrate the accuracy and the robustness of our scheme on the gas dynamics system, we have run test cases taken from the literature. During the whole calculation we are working on the fixed initial grid. However, plotting final solutions on the initial mesh, the results are difficult to analyze. Luckily, knowing the deformation gradient tensor everywhere and at anytime, we are able to observe the solution on the actual, deformed, mesh. For a better understanding of the results, all the problem solutions are displayed on the final mesh. For the numerical applications, the initial functions are projected over the polynomial space $\mathbb{P}^1$ on the cells. The slope limitation is performed using the Riemann invariants limitation procedure, refer to Section 4.1.4.

Polar Sod shock tube problem. Here, we consider the extension of the classical Sod shock tube [100] to the case of polar geometry. The present problem consists of a cylindrical shock tube of unity radius. The interface is located at $r = 0.5$. At the initial time, the states on the left and on the right sides of the interface are constant. The left state is a high pressure fluid characterized by $(\rho_L^0, P_L^0, \boldsymbol{U}_L^0) = (1, 1, \mathbf{0})$, the right state is a low pressure fluid defined by $(\rho_R^0, P_R^0, \boldsymbol{U}_R^0) = (0.125, 0.1, \mathbf{0})$. The gamma gas law is defined by $\gamma = \frac{7}{5}$. The computational domain is defined in polar coordinates by $(r, \theta) \in [0, 1] \times [0, \frac{\pi}{4}]$. The boundary conditions are symmetric for $\theta = 0$ and $\theta = \frac{\pi}{4}$, and wall, *i.e.*, the normal velocity is set to zero, at $r = 1$. The aim of this test case is to assess the scheme accuracy and the scheme symmetry preservation ability, refer Figure 4.4-(a). On Figure 4.4-(b), we note the strong accuracy difference between the first and the second-order. In this case, the limited solution is perfectly monotonic.



(a) Second-order DG scheme with limitation.

(b) Density profiles comparison.

Figure 4.4: Sod shock tube problem on a polar grid made of 100×5 cells.

A variant of the Gresho vortex problem. We propose a variant of the initial vortex problem defined by Gresho in [45]. The analytical solution is obtained by considering a steady vortex which is solution of the incompressible Euler equations. This solution is characterized by a balance between inertia and pressure gradient into the momentum equation. We recall hereafter the procedure to derive such a solution. Before proceeding any further we start by recalling the writing of the differential operators using cylindrical polar coordinates.

Let (x, y, z) denote the usual Cartesian coordinates and $(\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z)$ the Cartesian orthogonal basis. The polar cylindrical coordinates are (r, θ, z) where $r \geq 0$ and $\theta \in [0, 2\pi[$, and the corresponding basis in cylindrical polar geometry is $(\boldsymbol{e}_r, \boldsymbol{e}_\theta, \boldsymbol{e}_z)$. For $P = P(r, \theta, z)$ and $\boldsymbol{U} = u_r\, \boldsymbol{e}_r + u_\theta\, \boldsymbol{e}_\theta + u_z\, \boldsymbol{e}_z$, the gradient and divergence operators are expressed as follows in cylindrical

$$\boldsymbol{\nabla}\, P = \frac{\partial P}{\partial r}\boldsymbol{e}_r + \frac{1}{r}\frac{\partial P}{\partial \theta}\boldsymbol{e}_\theta + \frac{\partial P}{\partial z}\boldsymbol{e}_z,$$
$$\nabla \,.\, \boldsymbol{U} = \frac{1}{r}\frac{\partial (r\, u_r)}{\partial r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{\partial u_z}{\partial z}.$$

The curl of the velocity field $\boldsymbol{U}$ is given by

$$\boldsymbol{\nabla} \times \boldsymbol{U} = (\frac{1}{r}\frac{\partial u_z}{\partial \theta} - \frac{\partial u_\theta}{\partial z})\boldsymbol{e}_r + (\frac{\partial u_r}{\partial r} - \frac{\partial u_z}{\partial r})\boldsymbol{e}_\theta + \frac{1}{r}(\frac{\partial (r\, u_\theta)}{\partial r} - \frac{\partial u_r}{\partial \theta})\boldsymbol{e}_z. \tag{4.91}$$

Let $\psi = \psi(r, \theta, z, t)$ be a physical variable attached to the fluid, its material derivative writes as

$$\frac{\mathrm{d}\, \psi}{\mathrm{d}t} = \frac{\partial\, \psi}{\partial t} + \boldsymbol{U} \,.\, \boldsymbol{\nabla}\psi = \frac{\partial\, \psi}{\partial t} + u_r \frac{\partial \psi}{\partial r} + \frac{1}{r} u_\theta \frac{\partial \psi}{\partial \theta} + u_z \frac{\partial \psi}{\partial z}.$$

Being given the velocity vector, $\boldsymbol{U}$, the acceleration reads as

$$\frac{\mathrm{d}\, \boldsymbol{U}}{\mathrm{d}t} = \frac{\partial\, \boldsymbol{U}}{\partial t} + \left(u_r \frac{\partial}{\partial r} + \frac{u_\theta}{r}\frac{\partial}{\partial \theta} + u_z \frac{\partial}{\partial z} \right) \boldsymbol{U}. \tag{4.92}$$

Now, let us assume that the fluid flow obeys the following properties:

- The velocity field is orthoradial and its component depends only on the radius r

$$\boldsymbol{U} = u_\theta(r)\boldsymbol{e}_\theta. \tag{4.93}$$

- The thermodynamics variables, ρ and P are scalar valued functions which depends on r.
- The thermodynamical closure is provided by a gamma as law, $P = (\gamma - 1)\rho\varepsilon$.

Using the previous assumptions, we readily obtain

$$\boldsymbol{\nabla}\rho = \frac{\mathrm{d}\rho}{\mathrm{d}r}\boldsymbol{e}_r, \quad \boldsymbol{\nabla}P = \frac{\mathrm{d}P}{\mathrm{d}r}\boldsymbol{e}_r, \quad \boldsymbol{\nabla}\varepsilon = \frac{\mathrm{d}\varepsilon}{\mathrm{d}r}\boldsymbol{e}_r. \tag{4.94}$$

We also point out that the flow is incompressible since from (4.93) we deduce that $\nabla \,.\, \boldsymbol{U} = 0$. Moreover, the material derivative of a physical variable such that $\psi = \psi(r, t)$ coincides with its time derivative since the velocity is orthogonal to the gradient of this variable, $\frac{\mathrm{d}\, \psi}{\mathrm{d}t} = \frac{\partial\, \psi}{\partial t}$. These points involve that the internal energy and the mass continuity equations are automatically satisfied. Replacing the velocity field into (4.92) leads to following expression for the acceleration

$$\frac{\mathrm{d}\, \boldsymbol{U}}{\mathrm{d}t} = -\frac{u_\theta^2}{r}\boldsymbol{e}_r.$$

In writing this equation, we have used the fact that $\frac{\mathrm{d}\, \boldsymbol{e}_\theta}{\mathrm{d}t} = -\boldsymbol{e}_r$. Finally, the momentum equation reduces to the ordinary differential equation

$$\frac{\mathrm{d}P}{\mathrm{d}r} = -\rho\frac{u_\theta^2}{r}. \tag{4.95}$$

The time evolution of the specific kinetic energy, $k = \frac{1}{2}\boldsymbol{U}^2$, reads as

$$\rho \frac{\mathrm{d}\, k}{\mathrm{d}t} + \boldsymbol{\nabla} P \,.\, \boldsymbol{U} = 0. \tag{4.96}$$

Knowing that $k = k(r)$ and $\boldsymbol{\nabla} P \perp \boldsymbol{U}$ leads to the conservation of the kinetic energy. Finally, using (4.91) and the previous assumptions, the curl of the velocity field reduces to

$$\boldsymbol{\nabla} \times \boldsymbol{U} = (\frac{u_\theta}{r} + \frac{\mathrm{d}u_\theta}{\mathrm{d}r})\boldsymbol{e}_z. \tag{4.97}$$

Being given $\rho = \rho(r)$ and $u_\theta = u_\theta(r)$, the pressure field is obtained by solving the differential equation (4.95). Now, we propose to derive a smooth vortex solution by setting $\rho = \rho_0$ and

$$u_\theta(r) = \begin{cases} 2^{2n} U_0 \, (\frac{r}{r_v})^n (1 - \frac{r}{r_v})^n & \text{if } r \in [0, r_v], \\ 0 & \text{if } r \in]r_v, 1]. \end{cases} \tag{4.98}$$

Here, r_v denotes the radius of so vortex and n is a integer such that $n > 1$. The factor 2^{2n} is a normalization factor chosen such that the maximum of the orthoradial velocity is equal to one. Choosing the above form for the orthoradial velocity ensures that it is a smooth function with a compact support over $[0, 1]$. The integration of the momentum equation (4.95) gives

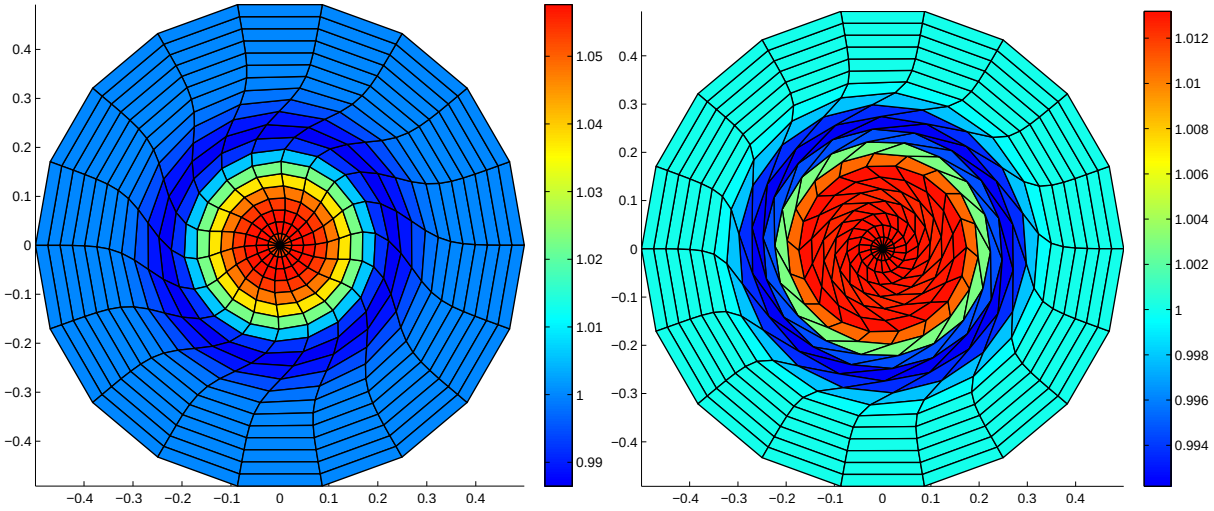
$$P(r) = \begin{cases} P(0) + 2^{4n} \rho_0 U_0^2 \, h(\frac{r}{r_v}) & \text{if } r \in [0, r_v], \\ P(0) + 2^{4n} \rho_0 U_0^2 \, h(1) & \text{if } r \in]r_v, 1]. \end{cases} \tag{4.99}$$

where $P(0)$ is an arbitrary integration constant and h is the real valued function defined by

$$h(\xi) = \int_0^\xi s^{2n-1} (1 - s)^{2n} \, \mathrm{d}s, \quad \text{for } \xi \in [0, 1]. \tag{4.100}$$

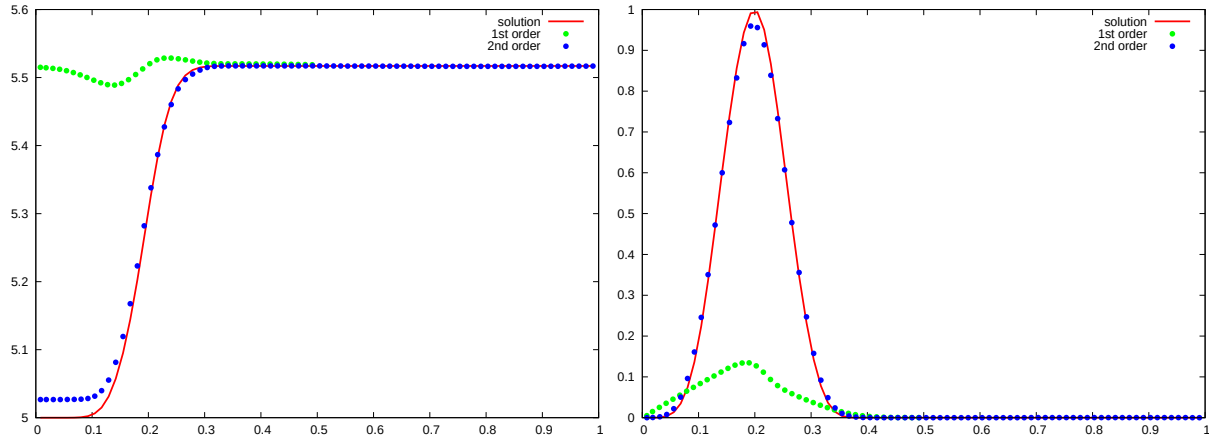
For numerical applications, we define the analytical solution setting the parameters to the following values: $U_0 = 1$, $\rho_0 = 1$, $P(0) = 5$, $r_v = 0.4$ and $n = 6$. The corresponding curves are displayed in Figure 4.6.

This variant of the Gresho problem is an interesting validation test case to assess the robustness and the accuracy of a Lagrangian scheme. On the first hand, the vorticity leads to a strong mesh rotation which can cause some problems as negative Jacobian determinants or negative densities. On the other hand, if the numerical diffusion is too important the flow motion stops very early. We run this Gresho problem on a polar grid made of 40×18 cells with our first and second-order DG schemes. The computational domain is defined in polar coordinates by $(r, \theta) \in [0, 1] \times [0, 2\pi]$. For the two-dimensional visualization, the solutions are displayed with a zoom in the zone $(r, \theta) \in [0, 0.5] \times [0, 2\pi]$, refer to Figure 4.5.
Firstly, we note that in two cases of first and second-order the symmetry is very well preserved. The second-order DG scheme being a lot less diffusive, the mesh is more deformed at the end than in the first-order case. We also notice that the incompressibility assumption is well satisfied in the second-order case, the density error being less than 1.3%. We have also displayed the pressure and velocity plots in the case of a polar grid made of 80×18 cells, refer to Figure 4.6. On Figure 4.6, we note the strong accuracy difference between the first and the second-order. This result confirms the very low diffusive contribution present in the second-order DG scheme in comparison to the first order. The small difference between the analytical solution and the second-order one shows the high accuracy of the numerical scheme. In Figure 4.7, we display the values the Jacobian of the second-order solution, on the subdivided triangular cells.



(a) First-order. (b) Second-order without limitation.

Figure 4.5: Gresho problem variant on a polar grid made of 40×18 cells at the final time $t = 0.36$: density map.



(a) Pressure profile. (b) Velocity profile.

Figure 4.6: Gresho problem variant on a polar grid made of 80×36 cells at the final time $t = 0.36$: comparison between the analytical solution and those obtained with the first and second-order numerical schemes.

We note that in this extreme vortex case, the Jacobian remains positive. This demonstrate the high robustness of the second-order scheme. If we perform the calculation further in time, the first-order solution stops very early. For the second-order DG scheme, the distortion of the mesh get even more important, refer to Figure 4.8.

We have run this variant of the Gresho problem with the second-order DG scheme until the final time $t = 2$, refer to Figure 4.8. We note that at the end of the calculation, the mesh is strongly deformed and presents some cross points and negative Jacobians. Anyway, as we are working on the initial configuration, the calculation does not stop. This shows the high robustness of our second-order scheme.

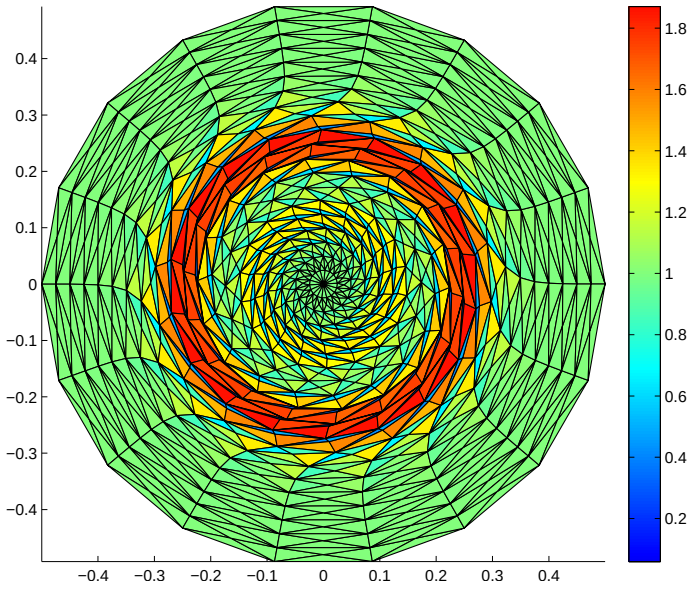
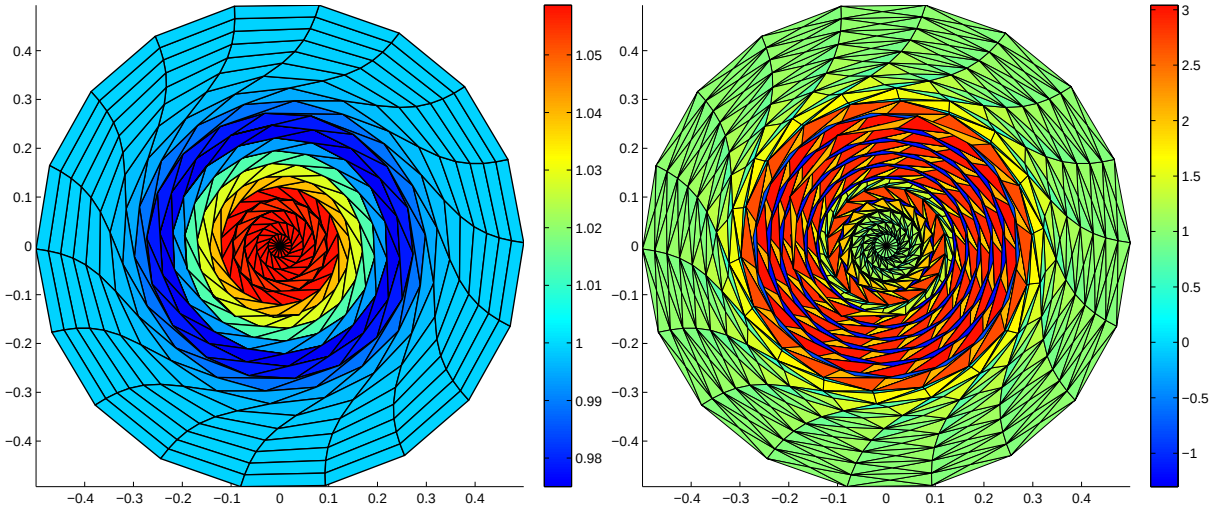


Figure 4.7: Gresho problem variant on a polar grid made of 40×18 cells at the final time $t = 0.36$ with our second-order DG scheme without limitation: Jacobian.

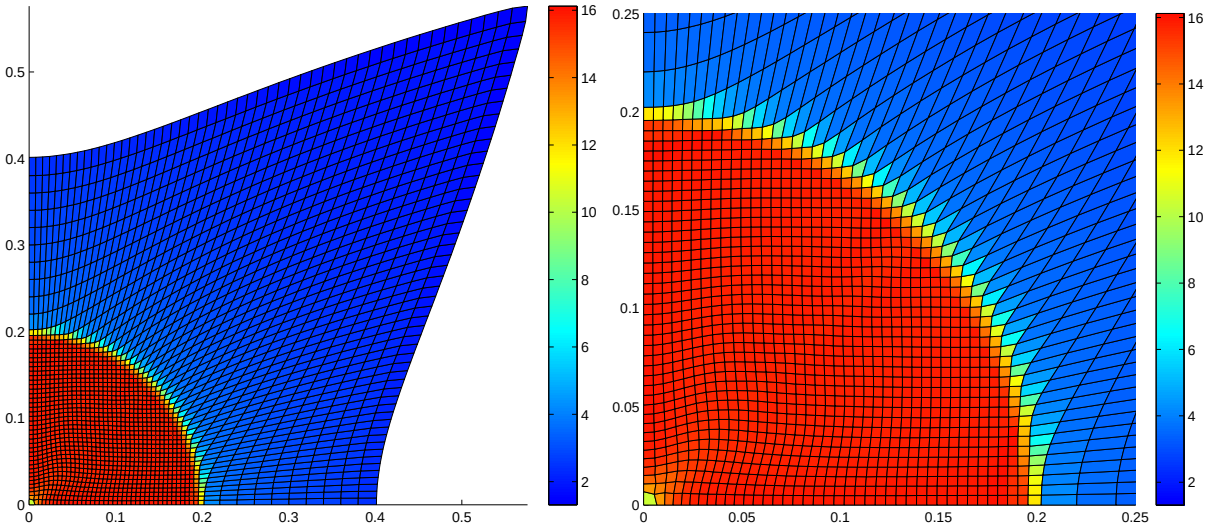


(a) Density.

(b) Jacobian.

Figure 4.8: Gresho problem variant on a polar grid made of 40×18 cells with our second-order DG scheme at the final time $t = 2$.

Noh problem. The Noh problem [85] is a well known test case used to validate Lagrangian schemes in the regime of infinite strength shock wave. In this test case, a cold gas with unit density is given an initial inward radial velocity of magnitude 1. The initial pressure is given by $P^0 = 10^{-6}$ and the polytropic index is equal to $\frac{5}{3}$. A diverging cylindrical shock wave is generated which propagates at speed $D = \frac{1}{3}$. The density plateau behind the shock wave reaches the value 16. The initial computational domain is defined by $(X, Y) = [0, 1] \times [0, 1]$. The boundary conditions on the X and Y axis are wall boundary conditions whereas a pressure given by $P^\star = P^0$ is prescribed at $X = Y = 1$. We run the Noh problem on a 50×50 Cartesian grid. This configuration leads to a severe test case since the mesh is not aligned with the flow.



(a) Global view.

(b) Zoom on the $[0, 0.25] \times [0, 0.25]$ zone.

Figure 4.9: Second-order DG scheme with limitation for a Noh problem on a 50×50 Cartesian grid: density map.

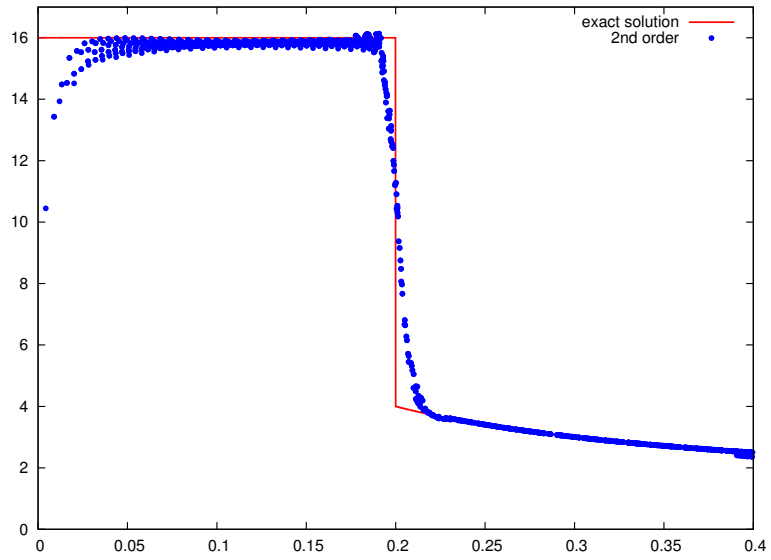


Figure 4.10: Density profiles comparison for a Noh problem on a 50×50 Cartesian grid with the second-order DG scheme with limitation.

We note that we have a very smooth and cylindrical solution, and that the shock is located at a circle whose radius is approximately 0.2, refer Figure 4.9. On Figure 4.10, we observe that the second-order plot is very sharp at the shock wave front and very similar to the one-dimensional cylindrical solution. Moreover the density at the shock plateau is not far from the analytical value.

Sedov point blast problem. We consider the Sedov problem for a point-blast in a uniform medium. An exact solution based on self-similarity arguments is available, see for instance

[53]. The initial conditions are characterized by $(\rho^0, P^0, \boldsymbol{U}^0) = (1, 10^{-6}, \mathbf{0})$, and the polytropic index is equal to $\frac{7}{5}$. We set an initial delta-function energy source at the origin prescribing the pressure in the cell containing the origin as follows, $P_{or} = (\gamma - 1)\rho_{or}\frac{\varepsilon^0}{v_{or}}$, where v_{or} denotes the volume of the cell containing the origin and ε^0 is the total amount of release energy. By choosing $\varepsilon^0 = 0.244816$, as suggested in [53], the solution consists of a diverging infinite strength shock wave whose front is located at radius $r = 1$ at $t = 1$, with a peak density reaching 6. First, we run Sedov problem with the second-order DG scheme with a 30×30 Cartesian grid on the domain $(X, Y) = [0, 1.2] \times [0, 1.2]$, refer to Figure 4.11-(a). Then, keeping the same conditions, we make use of an unstructured grid made of 1110 triangular cells, refer to Figure 4.12-(a), and an another unstructured grid 775 polygonal cells produced by Voronoi tessellation, as depicts Figure 4.13-(a). We point out that the triangular grid is made of completely anisotropic elements.
In the three cases, the numerical solution is very close to the one-dimensional analytical solution, see Figure 4.11-(c), 4.12-(c) and 4.13-(c). At the end of the computation, the shock wave front is correctly located at the end of the computation and is cylindrical, see Figure 4.11-(b), 4.12-(b) and 4.13-(b). And the density peak almost reaches 6. These results demonstrate the robustness and the accuracy of this scheme.

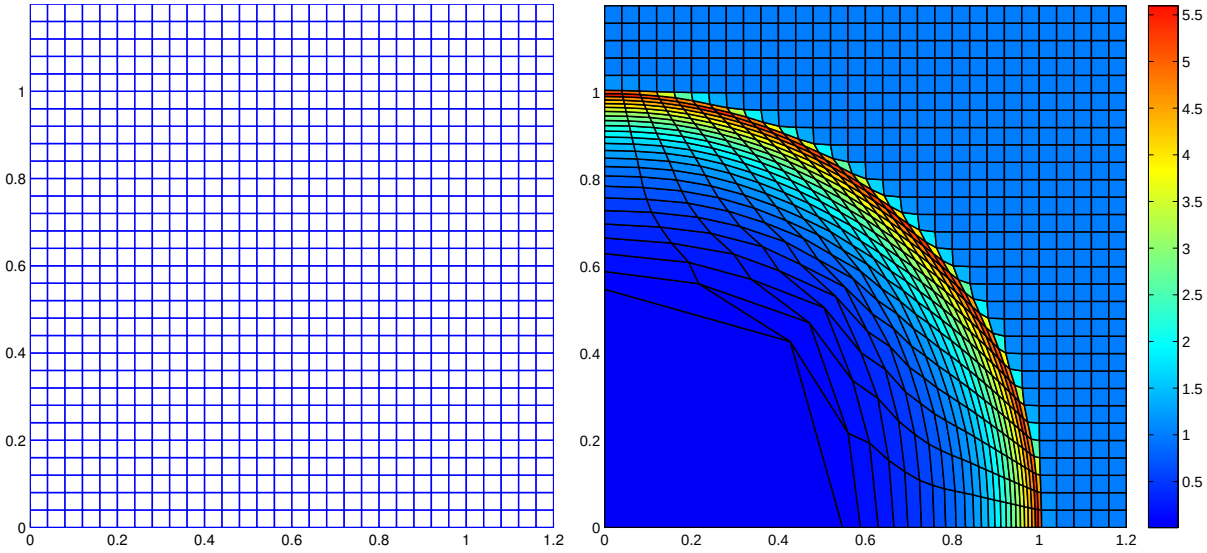
One knows that the presence of shocks and large deformations must affect the convergence rates. Nevertheless, plotting the numerical solutions obtained on different refined meshes, refer to Figure 4.14, we observe the positive impact of our high-order DG formulation. We also note that the solution correctly converges to the analytical solution meaning the shock front being well located at radius equal to 1 with a density reaching 6.

Saltzman problem. This test case taken from [36] is a well known difficult problem that allows to evaluate the robustness of Lagrangian schemes. It consists of a strong piston-driven shock wave calculated using an initially nonuniform mesh. The computational domain is defined by $(X, Y) \in [0, 1] \times [0, 0.1]$. The skewed initial mesh, displayed in Figure 4.15, is obtained transforming a uniform 100×10 Cartesian grid with the mapping

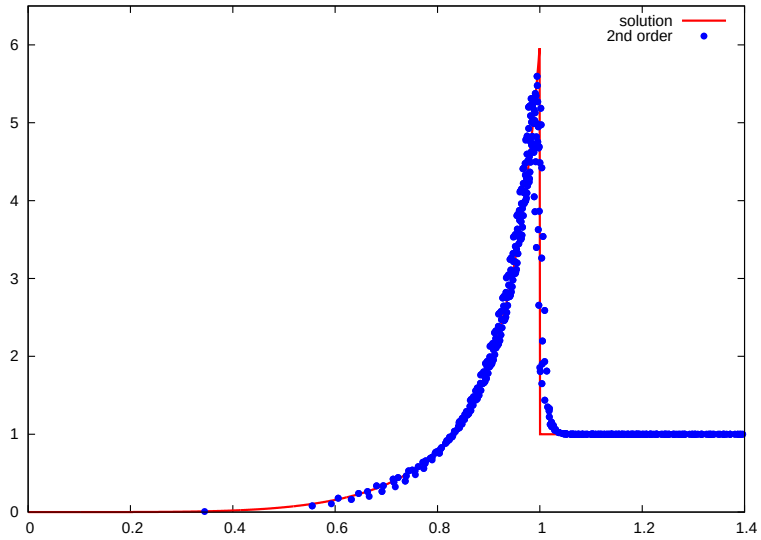
$$\begin{cases} X_{sk} = X + (0.1 - Y)\sin(\pi X), \\ Y_{sk} = Y. \end{cases}$$

The initial conditions are $(\rho^0, \varepsilon^0, \boldsymbol{U}^0) = (1, 10^{-6}, \mathbf{0})$, and the polytropic index is $\gamma = \frac{5}{3}$. At $X = 0$, a unit inward normal velocity is prescribed, the other boundaries are reflective walls. The analytical solution is a one-dimensional infinite strength shock wave that moves at speed $D = \frac{4}{3}$ in the right direction. Thus, the shock wave hits the face $x = 1$ at time $t = 0.75$. Behind the shock, the density is equal to 4.
We have displayed in Figure 4.16 the grid at time $t = 0.75$ which corresponds to the first bounce of the shock wave. We remark that the one-dimensional solution is very well preserved. Moreover, the location of the shock wave and the shock plateau are in good agreement with the analytical solution, refer to Figure 4.16. In Figure 4.17, we have plotted the grid and the density map at time $t = 0.9$, which corresponds to the time of the second bounce of the shock wave. Although the mesh is more wavy than before, it still exhibits a good quality. These results, in which no spurious modes appear, show the robustness of our second-order DG scheme.
To assess the accuracy of our DG scheme we compute the convergence order for the smooth solution of the Taylor-Green vortex test case, as in [35].

Taylor-Green vortex problem. We make use of the smooth Taylor-Green vortex test case in order to perform a convergence analysis on our DG scheme. This problem is derived by



(a) Initial grid. (b) Density map.



(c) Density profile.

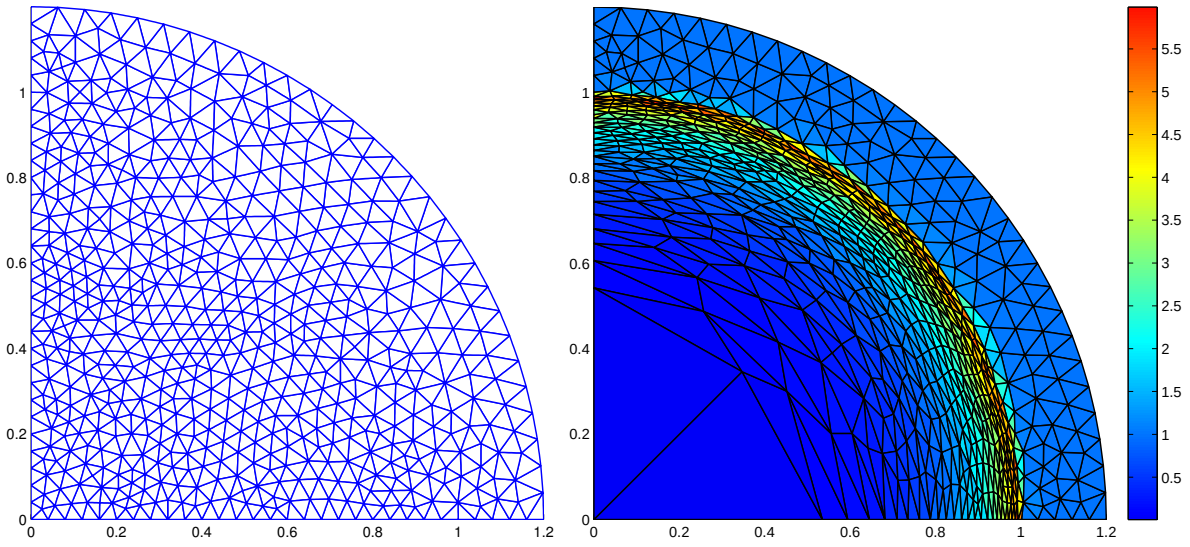
Figure 4.11: Point blast Sedov problem on a Cartesian grid made of 30×30 cells with the second-order DG scheme with limitation: density map.

considering an analytical solution of the incompressible Navier-Stokes equations, modify to the case of compressible Euler equations. The Taylor-Green vortex is characterized by the following conditions. The computational domain is defined by $(X, Y) = [0, 1] \times [0, 1]$. The initial density is uniform and denoted by ρ^0. The initial velocity field is divergence-free and reads

$$\boldsymbol{U}^0 = U^0 \begin{pmatrix} \sin(\pi x)\cos(\pi y) \\ -\cos(\pi x)\sin(\pi y) \end{pmatrix}.$$

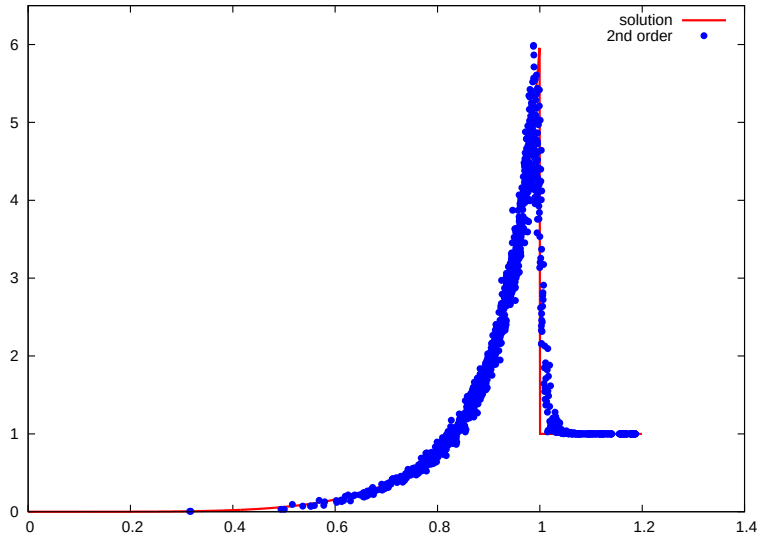
Let us point that with these two fields, the volume equation (3.31a) is automatically satisfied. Now, satisfying momentum equation (3.31b), P^0 is obtained by balancing the inertia term as

$$\boldsymbol{\nabla}_x P^0 = -\rho^0 \frac{\mathrm{d}\,\boldsymbol{U}^0}{\mathrm{d}t}. \tag{4.101}$$



(a) Initial grid.

(b) Second-order DG scheme with limitation.



(c) Density profiles comparison.

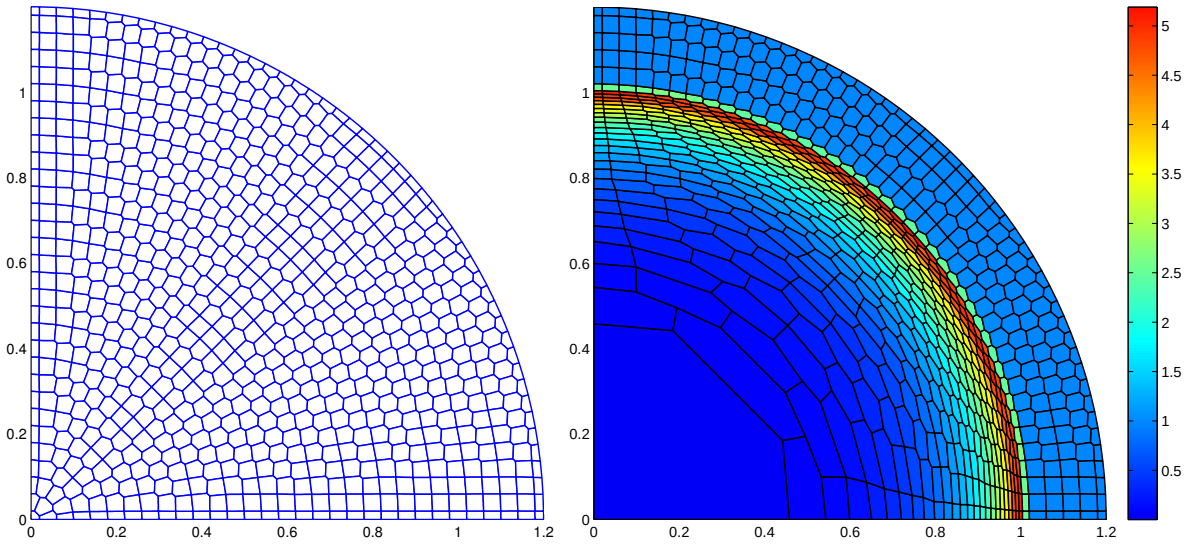
Figure 4.12: Point blast Sedov problem on a unstructured grid made of 1110 triangular cells with the second-order DG scheme with limitation: density map.

After some computations, we finally obtain

$$P^0 = \frac{1}{4}\rho^0 (U^0)^2 [\cos(2\pi x) + \cos(2\pi y)] + C^0,$$

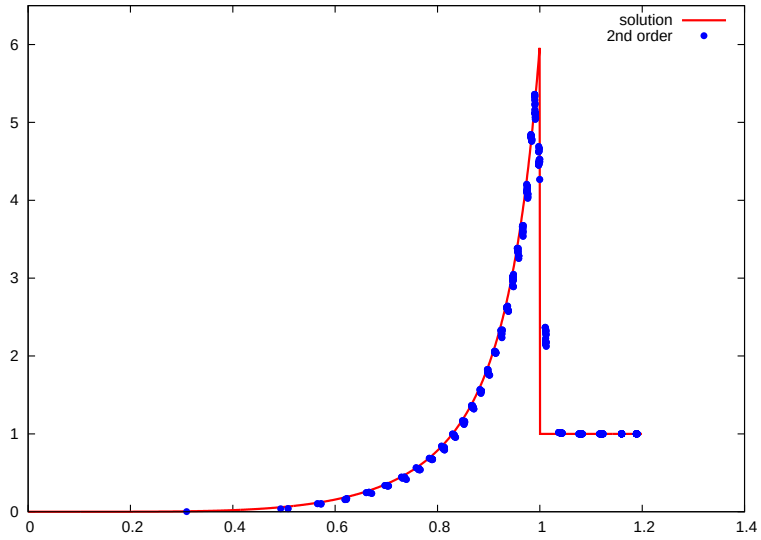
where C^0 is a constant that allows to define a non-negative pressure. Using these definitions of the density, velocity and pressure, the volume equation and the momentum equation are automatically satisfied. However, since we are computing this solution by solving the compressible Euler equations, it remains to check whether or not the total energy equation (3.31c) is satisfied. To this end, let us investigate the time rate of change of the specific internal energy ε

$$\rho^0 \frac{\mathrm{d}\,\varepsilon^0}{\mathrm{d}t} + P^0 \nabla_x \cdot \boldsymbol{U}^0 = \frac{1}{\gamma - 1} \boldsymbol{U}^0 \cdot \boldsymbol{\nabla}_x P^0,$$



(a) Initial grid.

(b) Second-order scheme with limitation.



(c) Density profiles comparison.

Figure 4.13: Point blast Sedov problem on a unstructured grid made of 775 polygonal cells with the second-order DG scheme with limitation: density map.

since $\varepsilon^0 = \frac{P^0}{(\gamma-1)\rho^0}$ and $\nabla_x \cdot \boldsymbol{U}^0 = 0$. Replacing $\boldsymbol{U}^0$ and P^0 by their analytical expressions, we finally obtain

$$\rho^0 \frac{\mathrm{d}\,\varepsilon^0}{\mathrm{d}t} + P^0 \nabla_x \cdot \boldsymbol{U}^0 = \frac{\pi}{4} \frac{\rho^0 (U^0)^3}{\gamma - 1} [\cos(3\pi x)\cos(\pi y) - \cos(3\pi y)\cos(\pi x)]. \tag{4.102}$$

Finally, adding the kinematic energy equation to the previous equation leads to the total energy equation

$$\rho^0 \frac{\mathrm{d}\,E^0}{\mathrm{d}t} + \nabla_x \cdot (P^0 \boldsymbol{U}^0) = \frac{\pi}{4} \frac{\rho^0 (U^0)^3}{\gamma - 1} [\cos(3\pi x)\cos(\pi y) - \cos(3\pi y)\cos(\pi x)]. \tag{4.103}$$

Finally, the numerical simulation of this test case solving the Lagrangian hydrodynamics equations requires the addition of the above source term in the energy (total or internal) equation.

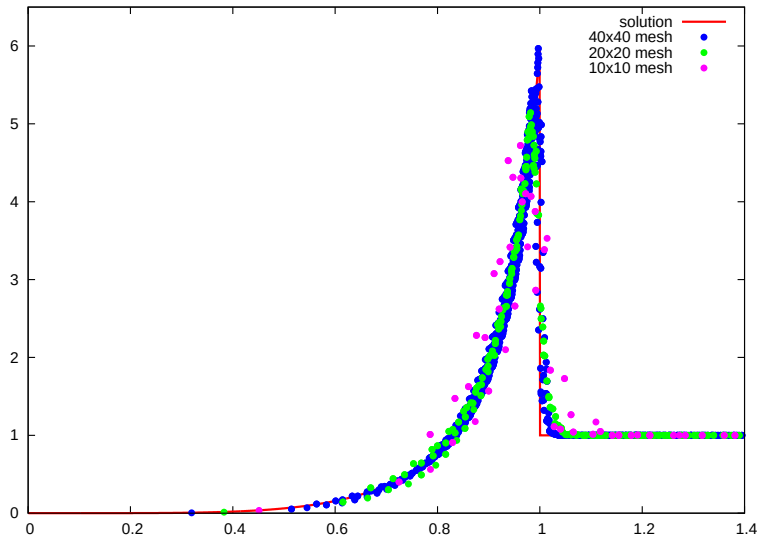


Figure 4.14: Second-order DG scheme with limitation for a point blast Sedov problem on Cartesian grids made of 10×10, 20×20 and 40×40, with the second-order DG scheme with limitation: density profiles.

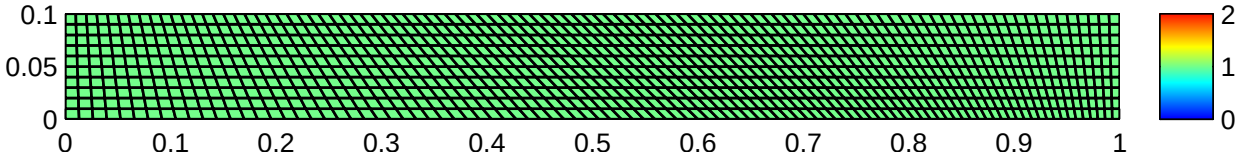


Figure 4.15: Initial computational domain for the Saltzman problem.

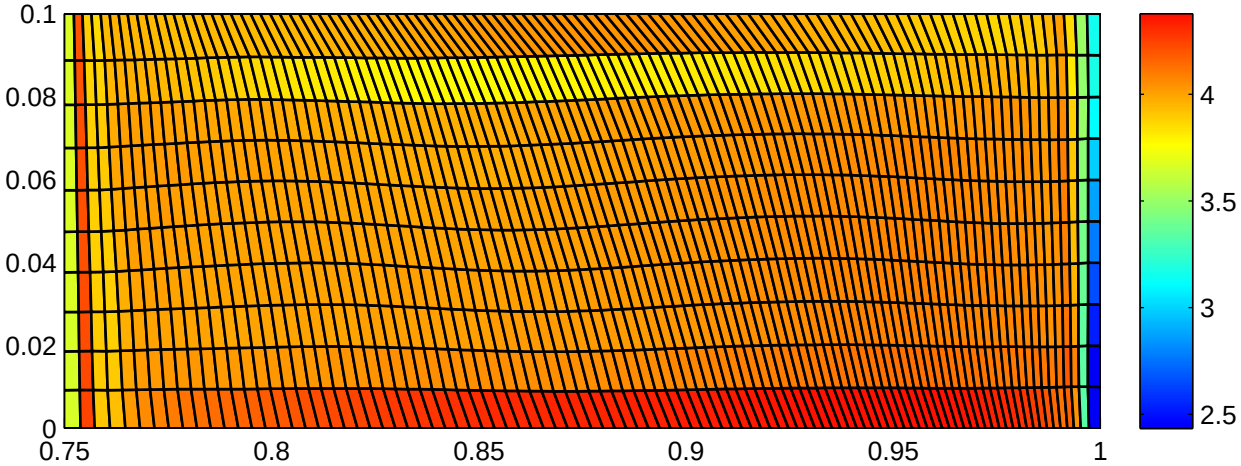
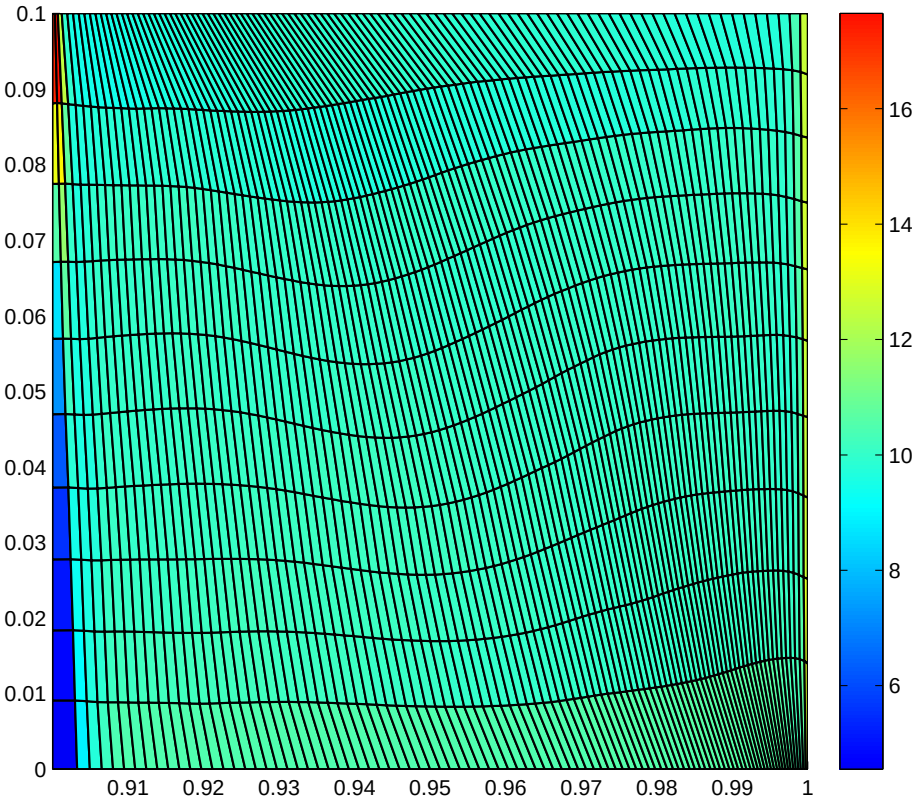
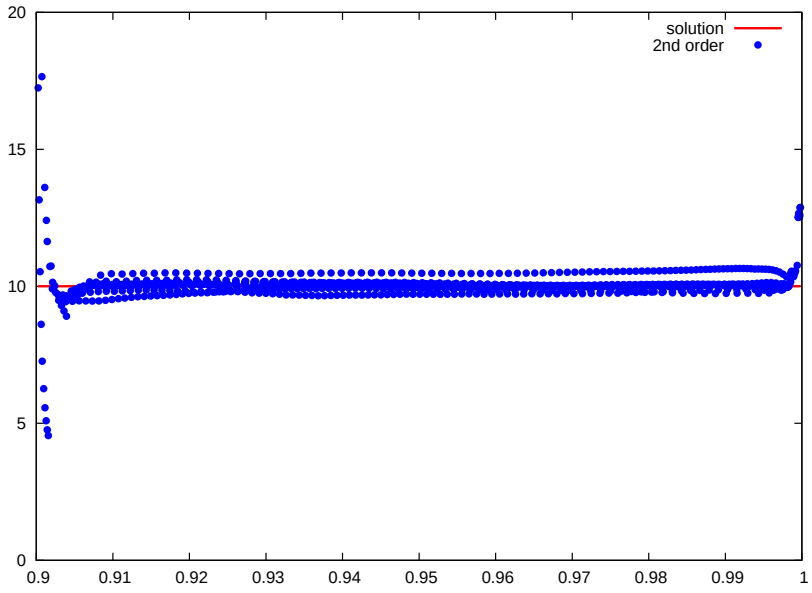


Figure 4.16: Saltzman problem. Grids and density map at time $t = 0.75$. Solution obtained with the second-order DG scheme with limitation.

For the numerical applications, we set $\rho^0 = 1$, $C^0 = 1$, $U^0 = 1$ and $\gamma = \frac{7}{5}$.
The results displayed in Figure 4.18 have been obtained by our first and second-order schemes on a 20×20 Cartesian grid.



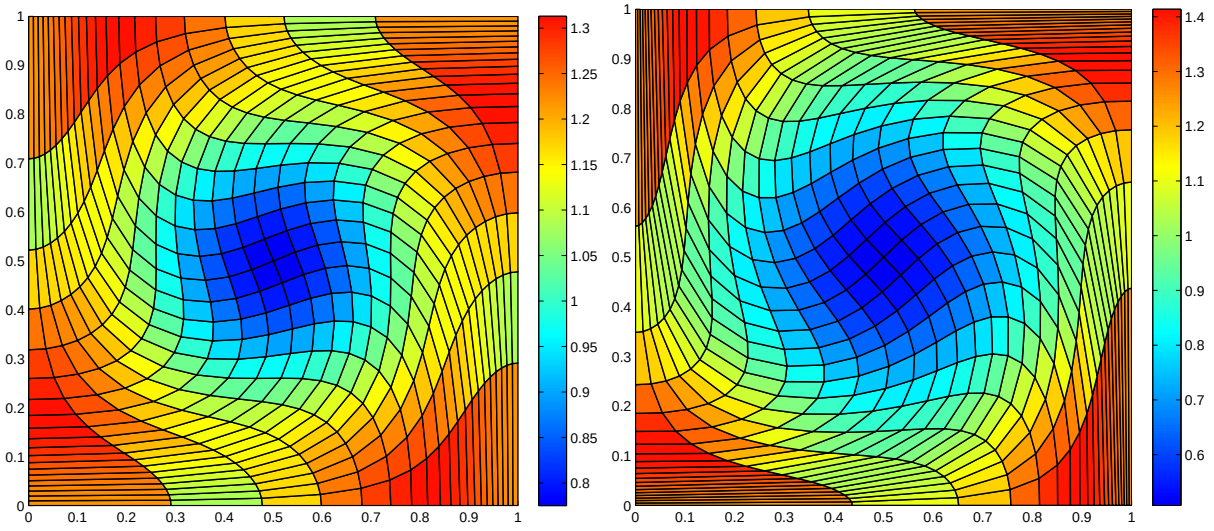
(a) Grids and density map.



(b) Density in the cells as function of the cell center x-coordinate versus analytical solution.

Figure 4.17: Solution of the Saltzman problem at time $t = 0.9$ obtained with the second-order DG scheme with limitation.

Using the analytical solution, we compute the global truncation error corresponding to our second DG scheme scheme and display it in Table 4.2. These results confirm the high accuracy



(a) First-order scheme.

(b) Second-order DG scheme without limitation.

Figure 4.18: Taylor-Green vortex problem on a 20×20 Cartesian grid at time $t = 0.75$: pressure map.

of this scheme.

	L_1		L_2		L_∞	
h	$E^h_{L_1}$	$q^h_{L_1}$	$E^h_{L_2}$	$q^h_{L_2}$	$E^h_{L_\infty}$	$q^h_{L_\infty}$
$\frac{1}{10}$	2.50E-2	1.48	3.71E-2	1.30	1.72E-1	1.35
$\frac{1}{20}$	8.98E-3	1.88	1.51E-2	1.75	6.73E-2	1.27
$\frac{1}{40}$	2.44E-3	1.94	4.48E-3	1.95	2.79E-2	1.68
$\frac{1}{80}$	6.36E-4	2.00	1.16E-3	2.00	8.68E-3	1.95
$\frac{1}{160}$	1.59E-4	2.01	2.90E-4	2.01	2.24E-3	2.01
$\frac{1}{320}$	3.94E-5	-	7.18E-5	-	5.54E-4	-

Table 4.2: Rate of convergence computed on the pressure in the case of the Taylor-Green vortex, with the second-order DG scheme without limitation.

We also compare these numerical errors with those obtained with the alternative second-order MUSCL method presented in the previous Section 3.3.7. As expected, the DG scheme presented is more accurate than the MUSCL scheme, refer to Table 4.3.
Now, we assess the accuracy of the limited DG scheme. Considering the results gathered in Table 4.4, we note that even if the slope limiters decrease somewhat the accuracy of the scheme, it remains second-order accurate, the rate of convergence being close to two.

Kidder isentropic compression. In [54], Kidder has constructed an analytical solution of the self-similar isentropic compression of a shell filled with perfect gas. Following [12, 77], we recall the main features of this solution in order to define the set up of the test case. Initially, the shell has the internal (resp. external) radius R_i (resp. R_e). Let P_i, P_e, ρ_i and ρ_e be the pressures and densities located at R_i and R_e. Since the compression is isentropic, we define $s = \frac{P_e}{\rho_e^\gamma}$, and we have $\rho_i = \rho_e \left(\frac{P_i}{P_e}\right)^{\frac{1}{\gamma}}$. Let $r(R,t)$ be the radius at time $t > 0$ of a fluid particle

	L_1		L_2		L_∞	
h	$E^h_{L_1}$	$q^h_{L_1}$	$E^h_{L_2}$	$q^h_{L_2}$	$E^h_{L_\infty}$	$q^h_{L_\infty}$
$\frac{1}{10}$	3.58E-2	1.44	5.08E-2	1.37	1.67E-1	1.17
$\frac{1}{20}$	1.32E-2	1.78	1.96E-2	1.56	7.41E-2	1.03
$\frac{1}{40}$	3.84E-3	1.93	6.66E-3	1.89	3.63E-2	1.58
$\frac{1}{80}$	1.01E-3	1.99	1.80E-3	1.98	1.21E-2	1.87
$\frac{1}{160}$	2.55E-4	2.00	4.57E-4	2.00	3.31E-3	1.97
$\frac{1}{320}$	6.38E-5	-	1.14E-4	-	8.47E-4	-

Table 4.3: Rate of convergence computed on the pressure in the case of the Taylor-Green vortex, with the second-order MUSCL scheme without limitation.

	L_1		L_2		L_∞	
h	$E^h_{L_1}$	$q^h_{L_1}$	$E^h_{L_2}$	$q^h_{L_2}$	$E^h_{L_\infty}$	$q^h_{L_\infty}$
$\frac{1}{10}$	5.80E-2	1.54	7.22E-2	1.31	2.24E-1	1.44
$\frac{1}{20}$	1.99E-2	2.33	2.92E-2	2.03	8.27E-2	1.34
$\frac{1}{40}$	3.96E-3	2.25	7.16E-3	2.20	3.26E-2	1.61
$\frac{1}{80}$	8.31E-4	2.17	1.56E-3	2.15	1.07E-2	1.52
$\frac{1}{160}$	1.85E-4	2.11	3.52E-4	2.14	3.73E-3	2.41
$\frac{1}{320}$	4.28E-5	-	8.01E-5	-	7.01E-4	-

Table 4.4: Rate of convergence computed on the pressure in the case of the Taylor-Green vortex, with the second-order DG scheme with limitation.

initially located at radius R. Looking for a solution of the gas dynamics equation under the form $r(R,t) = h(t)R$, using the isentropic feature of the flow and setting $\gamma = 1 + \frac{2}{\nu}$, where $\nu = 1, 2, 3$ indicates planar, cylindrical or spherical symmetry, we finally get the self-similar analytical solution for $t \in [0, \tau[$

$$\begin{aligned}
\rho(r(R,t),t) &= \rho^0(R)\, h(t)^{-\frac{2}{\gamma-1}}, \\
U(r(R,t),t) &= R\,\frac{\mathrm{d}\,h(t)}{\mathrm{d}t}, \\
P(r(R,t),t) &= P^0(R)\, h(t)^{-\frac{2\gamma}{\gamma-1}}.
\end{aligned}$$

Here, τ denotes the focusing time of the shell which is written

$$\tau = \sqrt{\frac{\gamma-1}{2}\frac{R_e^2 - R_i^2}{a_e^2 - a_i^2}},$$

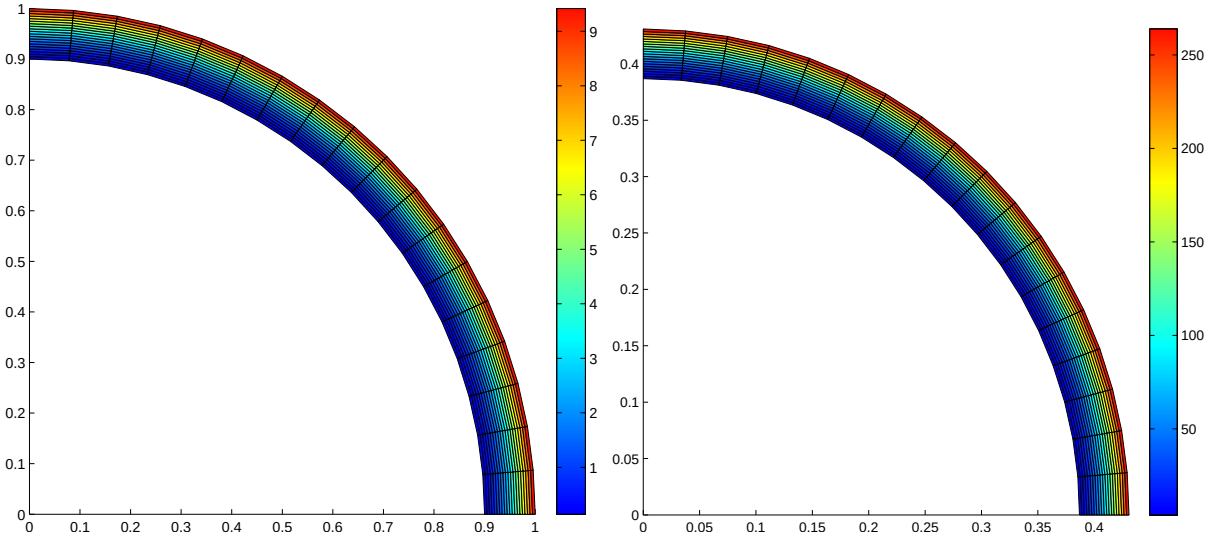
where $a^2 = s\gamma\rho^{\gamma-1}$ is the square of the isentropic sound speed. The particular form of the polytropic index enables us to get the analytical expression $h(t) = \sqrt{1 - \left(\frac{t}{\tau}\right)^2}$, which is valid for any $t \in [0, \tau[$. Note that $h(t)$ goes to zero when t goes to τ, hence τ corresponds to the collapse of the shell on itself. for $R \in [R_i, R_e]$, the initial density and pressure, ρ^0 and P^0, are defined by

$$\begin{aligned}
\rho^0(R) &= \left(\frac{R_e^2 - R^2}{R_e^2 - R_i^2}\rho_i^{\gamma-1} + \frac{R^2 - R_i^2}{R_e^2 - R_i^2}\rho_e^{\gamma-1}\right)^{\frac{1}{\gamma-1}}, \\
P^0(R) &= s\,(\rho^0(R))^\gamma.
\end{aligned}$$

Note that the initial velocity is equal to zero since the shell is assumed to be initially at rest. The isentropic compression is obtained imposing the following pressure laws at the internal and external faces of the shell:

$$P(r(R_i,t),t) = P_i\, h(t)^{-\frac{2\gamma}{\gamma-1}},$$
$$P(r(R_e,t),t) = P_e\, h(t)^{-\frac{2\gamma}{\gamma-1}}.$$

We point out that the velocity field is a linear function of the radius r which is a typical property of self-similar isentropic compression. For numerical applications, we consider the cylindrical shell characterized by $R_i = 0.9$ and $R_e = 1$. We set $P_i = 0.1$, $P_e = 10$ and $\rho_e = 10^{-2}$. Due to the cylindrical symmetry we have $\nu = 2$, hence $\gamma = 2$. The previous values lead to $\rho_i = 10^{-3}$, $s = 10^5$ and $\tau \simeq 7.265\,10^{-3}$. The initial computational domain is defined in polar coordinates by $(r,\theta) \in [0.9,1] \times [0,\frac{\pi}{2}]$. This analytical solution is useful in order to assess the scheme accuracy and the ability of a Lagrangian scheme to properly compute an isentropic compression with radial symmetry.



(a) At time $t = 0$.

(b) At time $t = 0.9\,\tau$.

Figure 4.19: Second-order DG scheme without limitation for a Kidder isentropic compression problem on a polar grid made of 20×18 cells: pressure map.

The results displayed in Figure 4.19 show that the symmetry is perfectly preserved, and those in Table 4.5 that the correct order is reached.

	L_1		L_2		L_∞	
$N_r \times N_\theta$	$E^h_{L_1}$	$q^h_{L_1}$	$E^h_{L_2}$	$q^h_{L_2}$	$E^h_{L_\infty}$	$q^h_{L_\infty}$
10×9	3.92E-2	1.82	1.55E-1	1.89	1.22E0	1.97
20×18	1.11E-2	1.95	4.18E-2	1.97	3.12E-1	1.98
40×36	2.88E-3	1.98	1.07E-2	1.99	7.92E-2	1.99
80×72	7.28E-4	1.99	2.69E-3	2.00	2.00E-2	1.99
160×144	1.83E-4	2.00	6.74E-4	2.00	5.03E-3	1.99
320×288	4.58E-5	-	1.69E-4	-	1.27E-3	-

Table 4.5: Rate of convergence computed on the pressure in the case of the Kidder isentropic compression, with the second-order DG scheme without limitation, at time $t = \frac{\tau}{2}$.

4.3 Third-order DG scheme

In this section, we present the third-order extension of the DG scheme described in Section 4.2. Here, we consider that the continuous mapping characterizing the flow motion of the fluid is quadratic with respect to the space variables. Thus, the edges of the mesh could curve through the flow deformation. We start this section by introducing the main features related to the curvilinear geometry.

4.3.1 Geometric consideration

We assume that the flow motion of the fluid is characterized by a quadratic mapping function. We remind that in the second-order case, the chosen basis functions used to develop the linear mapping were the P_1 barycentric coordinate functions, defined in (4.56). Thus, naturally we use the quadratic P_2 Finite Element basis functions. The notions and definitions inherent to this P_2 representation are taken from [39].

Now, characterizing the flow of the motion fluid by means of a quadratic mapping, we consider that the reference triangle $\mathcal{T}_i$ corresponds to the triangle in its initial position. We identify τ_i as the transformation of $\mathcal{T}_i$ at time t through the flow motion, refer to Figure 4.20.

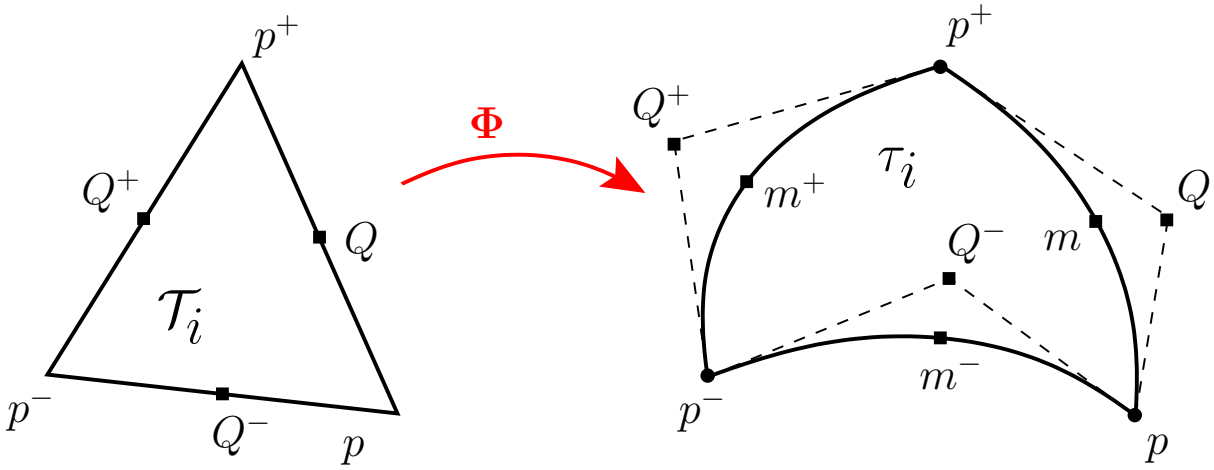


Figure 4.20: Transformation of $\mathcal{T}_i$ into τ_i.

We assume that the initial triangle is defined by the coordinate $\boldsymbol{X}$, while $\boldsymbol{x}$ characterizes the Bezier triangle τ_i at time t. In this frame, the quadratic mapping function writes

$$\boldsymbol{x} = \boldsymbol{\Phi}(\boldsymbol{X}, t) = \sum_p \boldsymbol{x}_p(t)\, \mu_p(\boldsymbol{X}), \tag{4.104}$$

where the points p are the triangular nodes and the control points Q of the Bezier edges. Let us recall the definition of the P_2 barycentric coordinate functions μ_p

$$\begin{aligned} \mu_p = (\lambda_p)^2, \quad \mu_{p^+} = (\lambda_{p^+})^2, \quad \mu_{p^-} = (\lambda_{p^-})^2, \\ \mu_Q = 2\lambda_p\lambda_{p^+}, \quad \mu_{Q^+} = 2\lambda_{p^+}\lambda_{p^-}, \quad \mu_{Q^-} = 2\lambda_{p^-}\lambda_p, \end{aligned}$$

where the functions λ_l, with $l \in \{p, p^+, p^-\}$, are the P_1 Finite Elements linear basis functions defined in (4.56). Finally, the quadratic mapping expresses as

$$\boldsymbol{\Phi}(\boldsymbol{X}, t) = \sum_{p \in \mathcal{P}(\mathcal{T}_i)} \left[\boldsymbol{x}_p(t)\, (\lambda_p(\boldsymbol{X}))^2 + 2\boldsymbol{x}_Q(t)\, \lambda_p(\boldsymbol{X})\lambda_p^+(\boldsymbol{X}) \right]. \tag{4.105}$$

Here, we emphasize that the trace of these P_2 Finite Elements functions over $\partial\mathcal{T}_i$ defines Bezier curves. Thus, we start by recalling the main features relative to the Bezier curves.

Bezier curves. Given the three control points p, Q and p^+, and a curvilinear parameter ζ in $[0, 1]$, we define the parametric representation of the curve $\mathcal{C}_{pp^+}$, depicted in Figure 4.21, as

$$\boldsymbol{x}(\zeta) = (1-\zeta)^2\boldsymbol{x}_p + 2\zeta(1-\zeta)\boldsymbol{x}_Q + \zeta^2\boldsymbol{x}_{p^+}. \tag{4.106}$$

This curve passes through the points p, m and p^+. The point $\boldsymbol{x}(\frac{1}{2}) = \boldsymbol{x}_m$ is called the midpoint of the curve and corresponds by definition to

$$\boldsymbol{x}_m = \boldsymbol{x}(\frac{1}{2}) = \frac{2\boldsymbol{x}_Q + \boldsymbol{x}_p + \boldsymbol{x}_{p^+}}{4}. \tag{4.107}$$

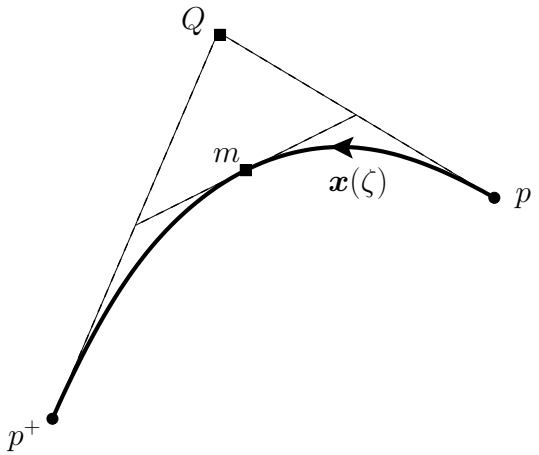


Figure 4.21: Bezier curve $\mathcal{C}_{pp^+}$ defined by the three control points p, Q and p^+, and its midpoint m.

This relation means that the m point is the middle of the respective middle points of segments pQ and Qp^+. Conversely, the control point Q expression is

$$\boldsymbol{x}_Q = \frac{4\boldsymbol{x}_m - \boldsymbol{x}_p - \boldsymbol{x}_{p^+}}{2}. \tag{4.108}$$

In the case where the control point Q is not aligned with p and p^+, the edge is necessarily a curve. On the contrary, if $\boldsymbol{x}_Q = \boldsymbol{x}_m$, the edge remains a straight line. Using the definition of the midpoint (4.107), the curve equation rewrites

$$\boldsymbol{x}(\zeta) = (1-\zeta)(1-2\zeta)\boldsymbol{x}_p + 4\zeta(1-\zeta)\boldsymbol{x}_m + \zeta(2\zeta-1)\boldsymbol{x}_{p^+}, \tag{4.109}$$

where $\zeta \in [0,1]$. Now, we aim at expressing the tangent of the curve $\mathcal{C}_{pp^+}$. Recalling that $\boldsymbol{t}\mathrm{d}l = \mathrm{d}\boldsymbol{x}$, the tangent $\boldsymbol{t}\mathrm{d}l$ of a Bezier curve writes as follows

$$\boldsymbol{t}\mathrm{d}l = \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\zeta}\,\mathrm{d}\zeta = 2\left((1-\zeta)(\boldsymbol{x}_Q - \boldsymbol{x}_p) + \zeta(\boldsymbol{x}_{p^+} - \boldsymbol{x}_Q)\right)\mathrm{d}\zeta. \tag{4.110}$$

Introducing the tangents $l_{pQ}\boldsymbol{t}_{pQ} = \boldsymbol{x}_Q - \boldsymbol{x}_p$ and $l_{Qp^+}\boldsymbol{t}_{Qp^+} = \boldsymbol{x}_{p^+} - \boldsymbol{x}_Q$ in the last formula, one gets

$$\boldsymbol{t}\mathrm{d}l = 2\left((1-\zeta)l_{pQ}\boldsymbol{t}_{pQ} + \zeta\, l_{Qp^+}\boldsymbol{t}_{Qp^+}\right)\mathrm{d}\zeta, \tag{4.111}$$

which means that the tangent vector is a convex combination of the vectors $l_{pQ}\boldsymbol{t}_{pQ}$ and $l_{Qp^+}\boldsymbol{t}_{Qp^+}$. Thus, the tangent of the curve evaluated at the nodes p and p^+ is respectively collinear with $\boldsymbol{t}_{pQ}$ and $\boldsymbol{t}_{Qp^+}$. We notice that the curve tangent at the midpoint is collinear with the vector $\boldsymbol{t}_{pp^+}$.

Here, we introduce the functions λ_p and λ_{p^+} in $[0,1]$ such as $\lambda_p(\zeta) + \lambda_{p^+}(\zeta) = 1$. These definitions lead to a new expression of equation (4.106)

$$\boldsymbol{x}(\zeta) = \lambda_p^2(\zeta)\boldsymbol{x}_p + 2\lambda_p(\zeta)\lambda_{p^+}(\zeta)\boldsymbol{x}_Q + \lambda_{p^+}^2(\zeta)\boldsymbol{x}_{p^+}. \tag{4.112}$$

This is the Bernstein (or Bezier) well-known representation. The linear functions λ_p and λ_{p^+} write $\lambda_p(\zeta) = (1-\zeta)$ and $\lambda_{p^+} = \zeta$, and identify with the restriction on the curve of the P_1 barycentric coordinate functions (4.56). We define the curve restriction of the P_2 functions, μ_p, μ_Q and μ_{p^+}, as follows

$$\mu_p = \lambda_p^2, \quad \mu_Q = 2\lambda_p\lambda_{p^+}, \quad \mu_{p^+} = \lambda_{p^+}^2. \tag{4.113}$$

Thus, the Bezier curve definition (4.112) rewrites

$$\boldsymbol{x}(\zeta) = \mu_p(\zeta)\boldsymbol{x}_p + \mu_Q(\zeta)\boldsymbol{x}_Q + \mu_{p^+}(\zeta)\boldsymbol{x}_{p^+}. \tag{4.114}$$

4.3.2 Deformation gradient tensor discretization

Thanks to the previous section, we define the mapping function $\boldsymbol{x} = \boldsymbol{\Phi}(\boldsymbol{X}, t)$ describing the flow motion, at time t in the triangle $\mathcal{T}_i$ as

$$\boldsymbol{\Phi}(\boldsymbol{X}, t) = \sum_{p\in\mathcal{P}(\mathcal{T}_i)} \left[\boldsymbol{x}_p(t)\left(\lambda_p(\boldsymbol{X})\right)^2 + 2\boldsymbol{x}_Q(t)\,\lambda_p(\boldsymbol{X})\lambda_p^+(\boldsymbol{X})\right], \tag{4.115}$$

where the position of the control point Q is constrained by the definition (4.108). In this frame, the deformation gradient tensor is piecewise linear over the triangles, and writes

$$\mathsf{F}_i(\boldsymbol{X}, t) = \frac{1}{|\mathcal{T}_i|}\sum_{p\in\mathcal{P}(\mathcal{T}_c)} \left[2\boldsymbol{x}_p(t)\otimes\lambda_p(\boldsymbol{X})L_{pc}\boldsymbol{N}_{pc} + 2\boldsymbol{x}_Q(t)\otimes\left(\lambda_p(\boldsymbol{X})L_{p^+c}\boldsymbol{N}_{p^+c} + 2\lambda_p^+(\boldsymbol{X})L_{pc}\boldsymbol{N}_{pc}\right)\right].$$

Performing an index permutation on this last equation, one gets

$$\mathsf{F}_i(\boldsymbol{X},t) = \frac{1}{|\mathcal{T}_c|}\sum_{p\in\mathcal{P}(\mathcal{T}_i)} 2\lambda_p(\boldsymbol{X})\left(\boldsymbol{x}_p(t)\otimes L_{pc}\boldsymbol{N}_{pc} + \boldsymbol{x}_Q(t)\otimes L_{p^+c}\boldsymbol{N}_{p^+c} + \boldsymbol{x}_{\overline{Q}}^-(t)\otimes L_{p^-c}\boldsymbol{N}_{p^-c}\right).$$

Finally, the deformation gradient tensor derived from a quadratic mapping writes

$$\mathsf{F}_i(\boldsymbol{X},t) = \sum_{p\in\mathcal{P}(\mathcal{T}_i)} \lambda_p(\boldsymbol{X})\ \mathsf{F}_p^i(t), \tag{4.116}$$

where the tensors F_p^i are defined as follows

$$\mathsf{F}_p^i(t) = \frac{2}{|\mathcal{T}_c|}\left(\boldsymbol{x}_p(t)\otimes L_{pc}\boldsymbol{N}_{pc} + \boldsymbol{x}_Q(t)\otimes L_{p^+c}\boldsymbol{N}_{p^+c} + \boldsymbol{x}_{\overline{Q}}^-(t)\otimes L_{p^-c}\boldsymbol{N}_{p^-c}\right). \tag{4.117}$$

In equation (4.116), the Jacobian tensor F is expressed by means of the continuous P_1 Finite Element basis functions. Nonetheless, considering the definition of the corner contributions F_p^i, the deformation gradient tensor is discontinuous between triangles. Concerning the initialization of the deformation gradient tensor, we recall that $\sum_{\mathcal{T}_i}\lambda_p(\boldsymbol{X}) = 1$. Thus, at the initial time $t = 0$, F_p^i is equal to the identity matrix.

Regarding the continuity of the tangent $\mathsf{F}_i\boldsymbol{T}\mathrm{d}L$, where $\boldsymbol{T}$ is the unit tangent of an edge of $\mathcal{T}_i$, we process identically to the second-order case. The computation of $\mathsf{F}_i\boldsymbol{T}\mathrm{d}L$ on the face f_{pp^+} leads to

$$\begin{aligned}\mathsf{F}_i\,\boldsymbol{T}\,\mathrm{d}L_{|_{pp^+}} &= \frac{2\,\mathrm{d}L}{L_{pp^+}}\left(\lambda_p(\boldsymbol{X})\,(\boldsymbol{x}_Q-\boldsymbol{x}_p) + \lambda_{p^+}(\boldsymbol{X})\,(\boldsymbol{x}_{p^+}-\boldsymbol{x}_Q)\right),\\ &= \frac{2\,\mathrm{d}L}{L_{pp^+}}\left(\lambda_p(\boldsymbol{X})\,l_{pQ}\boldsymbol{t}_{pQ} + \lambda_{p^+}(\boldsymbol{X})\,l_{Qp^+}\boldsymbol{t}_{Qp^+}\right),\end{aligned} \tag{4.118}$$

which is nothing but the Bezier curve tangent definition (4.111). The Eulerian normal is constructed with the only contribution of the points on the considered edge. This property ensures the continuity of the tangent and hence the continuity of the normals thanks to (4.28).

Finally, thanks to equation (3.10), the semi-discrete equation of tensor F reads

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathsf{F}_i(\boldsymbol{X},t) &= \frac{2}{|\mathcal{T}_i|}\sum_{p\in\mathcal{P}(\mathcal{T}_i)}\lambda_p(\boldsymbol{X})\left[\boldsymbol{U}_p(t)\otimes L_{pc}\boldsymbol{N}_{pc} + \boldsymbol{U}_Q(t)\otimes L_{p^+c}\boldsymbol{N}_{p^+c} + \boldsymbol{U}_{\overline{Q}}^-(t)\otimes L_{p^-c}\boldsymbol{N}_{p^-c}\right],\\ &= \sum_{p\in\mathcal{P}(\mathcal{T}_c)}\lambda_p(\boldsymbol{X})\ \frac{\mathrm{d}}{\mathrm{d}t}\mathsf{F}_p^i(t).\end{aligned} \tag{4.119}$$

In this expression, the velocity $\boldsymbol{U}_Q$ would not correspond to the velocity of a material point. The control point Q is a geometric entity required by the characterization of edges Bezier functions. Thus, assuming that through the flow transformation, a Bezier curve remains a Bezier curve, the control point velocity is constrained by the definition (4.108) and writes

$$\boldsymbol{U}_Q = \frac{4\boldsymbol{U}_m - \boldsymbol{U}_p - \boldsymbol{U}_{p^+}}{2}, \tag{4.120}$$

where $\boldsymbol{U}_p$, $\boldsymbol{U}_{p^+}$ and $\boldsymbol{U}_m$ are the nodal and midpoint velocities advecting the mesh.

4.3.3 Variational formulations

Here, we aim at developing a third-order discontinuous Galerkin discretization of the Lagrangian gas dynamics equations. In this Bezier cell frame, an accurate integration of the boundary terms in the cell is needed. This requirement is due to the geometric conservation law. In [10], B. Boutin and his co-workers have developed a first-order Finite Volume scheme on unstructured conical meshes satisfying the GCL. The Bezier curves being a particular case of the conical, the treatment of the GCL in the first-order case is done similarly than in [10]. One has to compute $\int_{\omega_c} \overline{\boldsymbol{U}} \cdot \boldsymbol{n} \mathrm{d}l$ in a compatible manner with the rate of change of the volume of cell ω_c. Nonetheless, as for the second-order DG scheme, we consider an high-order extension of this GCL consideration. It consists in computing the time rate of change of a curvilinear cell volume.

Geometric conservation law. Let us consider a polygonal cell ω_c deformed through the quadratic flow motion of the fluid. In this case, the edges of ω_c are nothing but Bezier curves presented above, refer to Figure 4.22. Through the use of the divergence theorem (A.34b), we can easily define the volume of the Bezier cell ω_c as

$$|\omega_c| = \int_{\omega_c} \mathrm{d}\omega = \frac{1}{2} \int_{\omega_c} \nabla_x \cdot \boldsymbol{x} \, \mathrm{d}V = \frac{1}{2} \int_{\partial\omega_c} \boldsymbol{x} \cdot \boldsymbol{n} \, \mathrm{d}l. \tag{4.121}$$

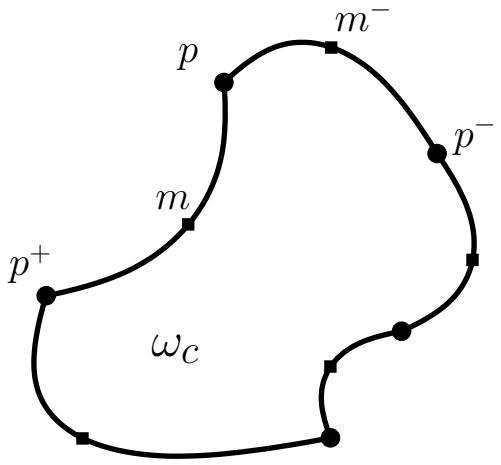


Figure 4.22: Image of a polygonal cell Ω_c in the actual configuration: $\omega_c = \boldsymbol{\Phi}(\Omega_c, t)$

The analytical integration of this expression using the definitions (4.109) and (4.111) gives

$$|\omega_c| = \frac{1}{2} \sum_{p \in \mathcal{P}(c)} \frac{1}{3} \boldsymbol{x}_p \cdot l_{Q^-Q} \boldsymbol{n}_{Q^-Q} + \frac{1}{2} \sum_{m \in \mathcal{M}(c)} \frac{2}{3} \boldsymbol{x}_m \cdot l_{pp^+} \boldsymbol{n}_{pp^+}, \tag{4.122}$$

where $\mathcal{P}(c)$ is the node set of cell ω_c and $\mathcal{M}(c)$ the set constituted of the midpoints of the Bezier edges. This formula computes the volume of a polygonal cell wherein the edges are characterized by Bezier curves. Now, to determine the time rate of change of the volume $|\omega_c|$, we compute the material derivative of (4.122). In the end, one gets

$$\frac{\mathrm{d}}{\mathrm{d}t}|\omega_c| = \sum_{p \in \mathcal{P}(c)} \frac{1}{3} \boldsymbol{U}_p \cdot l_{Q^-Q} \boldsymbol{n}_{Q^-Q} + \sum_{m \in \mathcal{M}(c)} \frac{2}{3} \boldsymbol{U}_m \cdot l_{pp^+} \boldsymbol{n}_{pp^+}. \tag{4.123}$$

In [10], in the particular case of Bezier curves, the volume of a cell and its time evolution reduce to the previous results (4.122) and (4.123). Furthermore, in the special case of straight edges, one knows that the geometric point Q and the midpoint m coincide. Considering a linear velocity field, we set the midpoint velocity as the average value of the edge nodes p and p^+, *i.e.*, $\boldsymbol{U}_m = \frac{1}{2}(\boldsymbol{U}_p + \boldsymbol{U}_{p^+})$. Under these assumptions, the results presented in the case of a Bezier cell reduce to the formulas (3.45) and (3.48) exhibited in Section 3.3.2.

Specific volume equation discretization. Let us recall the variational formulation over the cell Ω_c of the specific volume equation

$$\begin{aligned}\int_{\Omega_c} \rho^0 \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})\sigma_q^c \,\mathrm{d}V &= \sum_{k=0}^{K} \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})_k^c \int_{\Omega_c} \rho^0 \sigma_q^c \sigma_k^c \,\mathrm{d}V, \\ &= -\sum_{i=1}^{ntri} \int_{\mathcal{T}_i^c} \boldsymbol{U} \cdot \mathsf{G}\, \boldsymbol{\nabla}_X \sigma_q^c \,\mathrm{d}V + \int_{\partial\Omega_c} \overline{\boldsymbol{U}} \cdot \sigma_q^c \,\mathsf{G}\boldsymbol{N}\mathrm{d}L. \end{aligned} \tag{4.124}$$

Again, the main difficulty lies in the evaluation of the boundary term. The fundamental assumption considering that through the flow motion a Bezier curve remains a Bezier curve gives a definition of the edges velocity

$$\boldsymbol{U}_{|_{pp^+}}(\zeta) = (1-\zeta)(1-2\zeta)\boldsymbol{U}_p + 4\zeta(1-\zeta)\boldsymbol{U}_m + \zeta(2\zeta-1)\boldsymbol{U}_{p^+}, \tag{4.125}$$

where $\boldsymbol{U}_{|_{pp^+}}$ is the quadratic velocity of the face f_{pp^+}, and $\zeta \in [0,1]$ is the curvilinear abscissa. Similarly to the second-order discretization, we want to integrate analytically the boundary terms. Here, the polynomial basis function are piecewise quadratic. Thus, on the face f_{pp^+} the function σ_q^c can be defined by means of the curvilinear abscissa and of its extrapolated values at the nodes p, p^+ and at the midpoint m as

$$\sigma_{q|_{pp^+}}^c(\zeta) = (1-\zeta)(1-2\zeta)\,\sigma_q^c(\boldsymbol{X}_p) + 4\zeta(1-\zeta)\,\sigma_q^c(\boldsymbol{X}_m) + \zeta(2\zeta-1)\,\sigma_q^c(\boldsymbol{X}_{p^+}). \tag{4.126}$$

This time, the deformation gradient tensor F, and hence the G matrix, are piecewise linear over the triangle. But thanks to equations (4.110) and (4.118), we are able to express $\mathsf{G}\boldsymbol{N}\mathrm{d}L$ on face f_{pp^+} as

$$\mathsf{G}\,\boldsymbol{N}\,\mathrm{d}L_{|_{pp^+}} = 2\left((1-\zeta)\,l_{pQ}\boldsymbol{n}_{pQ} + \zeta\, l_{Qp^+}\boldsymbol{n}_{Qp^+}\right)\mathrm{d}\zeta. \tag{4.127}$$

Substituting (4.127) in the definition of the boundary term leads to

$$\begin{aligned}\int_{\partial\Omega_c} \overline{\boldsymbol{U}}\sigma_q^c \cdot \mathsf{G}\boldsymbol{N}\mathrm{d}L &= \sum_{p\in\mathcal{P}(c)} \int_p^{p^+} \overline{\boldsymbol{U}}\sigma_q^c \cdot \mathsf{G}\boldsymbol{N}\mathrm{d}L, \\ &= \sum_{p\in\mathcal{P}(c)} \Bigg[\left(\int_0^1 \overline{\boldsymbol{U}}\sigma_{q|_{pp^+}}^c(\zeta)\; 2(1-\zeta)\,\mathrm{d}\zeta\right)\cdot l_{pQ}\boldsymbol{n}_{pQ} \\ &\qquad + \left(\int_0^1 \overline{\boldsymbol{U}}\sigma_{q|_{pp^+}}^c(\zeta)\; 2\zeta\,\mathrm{d}\zeta\right)\cdot l_{Qp^+}\boldsymbol{n}_{Qp^+}\Bigg].\end{aligned} \tag{4.128}$$

Combining this last expression with (4.125) and (4.126), we are able to analytically evaluate (4.128). At the end, one gets

$$\int_{\partial\Omega_c} \overline{\boldsymbol{U}}\sigma_q^c \cdot \mathsf{G}\boldsymbol{N}\mathrm{d}L = \sum_{p\in\mathcal{P}(c)} \frac{1}{3}\boldsymbol{U}_p \cdot l_{pc}^q \boldsymbol{n}_{pc}^q + \sum_{m\in\mathcal{M}(c)} \frac{2}{3}\boldsymbol{U}_m \cdot l_{mc}^q \boldsymbol{n}_{mc}^q, \tag{4.129}$$

where the corner normals $l^q_{pc}\boldsymbol{n}^q_{pc}$, weighted by the basis function σ^c_q, write

$$l^q_{pc}\boldsymbol{n}^q_{pc} = l^{-,q}_{pc}\boldsymbol{n}^{-,q}_{pc} + l^{+,q}_{pc}\boldsymbol{n}^{+,q}_{pc}, \tag{4.130}$$

$$l^{-,q}_{pc}\boldsymbol{n}^{-,q}_{pc} = \frac{1}{10}\left[(6\,\sigma^c_q(\boldsymbol{X}_p) + 4\,\sigma^c_q(\boldsymbol{X}_{m^-}))l_{Q^-p}\boldsymbol{n}_{Q^-p} + (\sigma^c_q(\boldsymbol{X}_p) - \sigma^c_q(\boldsymbol{X}_{p^-}))l_{p^-p}\boldsymbol{n}_{p^-p}\right], \tag{4.131}$$

$$l^{+,q}_{pc}\boldsymbol{n}^{+,q}_{pc} = \frac{1}{10}\left[(6\,\sigma^c_q(\boldsymbol{X}_p) + 4\,\sigma^c_q(\boldsymbol{X}_m))l_{pQ}\boldsymbol{n}_{pQ} + (\sigma^c_q(\boldsymbol{X}_p) - \sigma^c_q(\boldsymbol{X}_{p^+}))l_{pp^+}\boldsymbol{n}_{pp^+}\right], \tag{4.132}$$

and the weighted midpoint normals $l^q_{mc}\boldsymbol{n}^q_{mc}$ express as follows

$$l^q_{mc}\boldsymbol{n}^q_{mc} = l^{-,q}_{mc}\boldsymbol{n}^{-}_{mc} + l^{+,q}_{mc}\boldsymbol{n}^{+}_{mc}, \tag{4.133}$$

$$l^{-,q}_{mc}\boldsymbol{n}^{-}_{mc} = \frac{1}{5}\left(4\,\sigma^c_q(\boldsymbol{X}_m) + \sigma^c_q(\boldsymbol{X}_p)\right) l_{pQ}\boldsymbol{n}_{pQ}, \tag{4.134}$$

$$l^{+,q}_{mc}\boldsymbol{n}^{+}_{mc} = \frac{1}{5}\left(4\,\sigma^c_q(\boldsymbol{X}_m) + \sigma^c_q(\boldsymbol{X}_{p^+})\right) l_{Qp^+}\boldsymbol{n}_{Qp^+}. \tag{4.135}$$

In the particular case of $q = 0$, these normals reduce to

$$l_{pc}\boldsymbol{n}_{pc} = l_{Q^-Q}\boldsymbol{n}_{Q^-Q}, \tag{4.136}$$

$$l_{mc}\boldsymbol{n}_{mc} = l_{pp^+}\boldsymbol{n}_{pp^+}. \tag{4.137}$$

Thus, the first moment of equation (4.129) reads

$$\int_{\partial\Omega_c} \overline{\boldsymbol{U}}\,.\,\mathsf{G}\boldsymbol{N}\mathrm{d}L = \int_{\partial\omega_c} \overline{\boldsymbol{U}}\,.\,\boldsymbol{n}\mathrm{d}l = \sum_{p\in\mathcal{P}(c)} \frac{1}{3}\boldsymbol{U}_p\,.\,l_{Q^-Q}\boldsymbol{n}_{Q^-Q} + \sum_{m\in\mathcal{M}(c)} \frac{2}{3}\boldsymbol{U}_m\,.\,l_{pp^+}\boldsymbol{n}_{pp^+}. \tag{4.138}$$

Considering (4.123), this result ensures the respect of the GCL condition. Finally, we have derived the semi-discrete equations on the specific volume successive moments, in respect with the GCL. The semi-discrete equations write

$$\int_{\Omega_c} \rho^0 \frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{\rho})\sigma^c_q\,\mathrm{d}V = -\sum_{i=1}^{ntri}\int_{\mathcal{T}^c_i} \boldsymbol{U}\,.\,\mathsf{G}\boldsymbol{\nabla}_X\sigma^c_q\mathrm{d}V + \sum_{p\in\mathcal{P}(c)} \frac{1}{3}\boldsymbol{U}_p\,.\,l_{Q^-Q}\boldsymbol{n}_{Q^-Q} + \sum_{m\in\mathcal{M}(c)} \frac{2}{3}\boldsymbol{U}_m\,.\,l_{pp^+}\boldsymbol{n}_{pp^+}. \tag{4.139}$$

For the momentum discretization, we use a similar procedure as the one presented in the second-order section, extended to the quadratic case.

Momentum discretization. For the momentum equation, the local variational formulation of (4.2c) on Ω_c leads to

$$\int_{\Omega_c} \rho^0 \frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t}\sigma^c_q\,\mathrm{d}V = \sum_{i=1}^{ntri}\int_{\mathcal{T}^c_i} P\,\mathsf{G}\boldsymbol{\nabla}_X\sigma^c_q\,\mathrm{d}V - \int_{\partial\Omega_c} \overline{P}\sigma^c_q\,\mathsf{G}\boldsymbol{N}\mathrm{d}L. \tag{4.140}$$

To evaluate the boundary term of (4.140), we set the edge pressure in the cell Ω_c as

$$\overline{P}^{\,c}_{|_{pp^+}}(\zeta) = (1-\zeta)(1-2\zeta)P^+_{pc} + 4\zeta(1-\zeta)P_{mc} + \zeta(2\zeta-1)P^-_{p^+c}, \tag{4.141}$$

where P^+_{pc} and $P^-_{p^+c}$ are respectively the right and left nodal pressures at the nodes p and p^+, and P_{mc} the midpoint pressure in the cell Ω_c. The analytical integration of the cell boundary term of (4.140) writes

$$\int_{\partial\Omega_c} \overline{P}\sigma^c_q\,\mathsf{G}\boldsymbol{N}\mathrm{d}L = \sum_{p\in\mathcal{P}(c)} \frac{1}{3}\left(P^-_{pc}\,l^{-,q}_{pc}\boldsymbol{n}^{-,q}_{pc} + P^+_{pc}\,l^{+,q}_{pc}\boldsymbol{n}^{+,q}_{pc}\right) + \sum_{m\in\mathcal{M}(c)} \frac{2}{3}P_{mc}\,l^q_{mc}\boldsymbol{n}^q_{pc}. \tag{4.142}$$

Finally, we have derived the semi-discrete equations on the successive moments of the velocity, with an analytical integration of the numerical flux. The semi-discrete equations write

$$\int_{\Omega_c} \rho^0 \frac{\mathrm{d}\,\boldsymbol{U}}{\mathrm{d}t} \sigma_q^c \,\mathrm{d}V = \sum_{i=1}^{ntri} \int_{\mathcal{T}_i^c} P \mathsf{G} \boldsymbol{\nabla}_X \sigma_q^c \mathrm{d}V - \sum_{p \in \mathcal{P}(c)} \frac{1}{3} \boldsymbol{F}_{pc}^q - \sum_{m \in \mathcal{M}(c)} \frac{2}{3} \boldsymbol{F}_{mc}^q, \tag{4.143}$$

where $\boldsymbol{F}_{pc}^q = P_{pc}^- \, l_{pc}^{-,q} \boldsymbol{n}_{pc}^- + P_{pc}^+ \, l_{pc}^{+,q} \boldsymbol{n}_{pc}^+$ and $\boldsymbol{F}_{mc}^q = P_{mc} \, l_{mc}^q \boldsymbol{n}_{pc}^q$ identify with the qth moment of the subcell forces.

Total energy discretization. The procedure to discretize the total energy equation (4.2d) is identical as the one used for the momentum equation. Let us recall the local variational formulation of (4.2d) on Ω_c

$$\int_{\Omega_c} \rho^0 \frac{\mathrm{d}\,E}{\mathrm{d}t} \sigma_q^c \,\mathrm{d}V = \sum_{i=1}^{ntri} \int_{\mathcal{T}_i^c} P \, \boldsymbol{U} \cdot \mathsf{G}_i^c \boldsymbol{\nabla}_X \sigma_q^c \mathrm{d}V - \int_{\partial\Omega_c} \overline{P\boldsymbol{U}} \cdot \sigma_q^c \mathsf{G} \boldsymbol{N} \mathrm{d}L. \tag{4.144}$$

Now, we make equivalent assumptions on the numerical flux $\overline{P\boldsymbol{U}}$ to the one used in the second-order section. In this similar frame, the semi-discrete equations of the total energy write

$$\int_{\Omega_c} \rho^0 \frac{\mathrm{d}\,E}{\mathrm{d}t} \sigma_q^c \,\mathrm{d}V = \sum_{i=1}^{ntri} \int_{\mathcal{T}_i^c} P \, \boldsymbol{U} \cdot \mathsf{G} \boldsymbol{\nabla}_X \sigma_q^c \mathrm{d}V - \sum_{p \in \mathcal{P}(c)} \frac{1}{3} \boldsymbol{U}_m \cdot \boldsymbol{F}_{pc}^q - \sum_{m \in \mathcal{M}(c)} \frac{2}{3} \boldsymbol{U}_m \cdot \boldsymbol{F}_{mc}^q. \tag{4.145}$$

At the end, we have $3 \times (K+1)$ semi-discrete equations for the same number of unknowns. To complete our scheme, we need to define our nodal solvers and midpoints solvers to compute $\boldsymbol{U}_p$, $\boldsymbol{U}_m$, $\boldsymbol{F}_{pc}^q$ and $\boldsymbol{F}_{mc}^q$. For that purpose, we make use of the entropy analysis done in the general framework Section 4.1.3, to ensure that the kinetic energy is correctly dissipated in internal energy through a shock. This will provide an expression of each subcell forces.

4.3.4 Nodal solvers

The entropic analysis done in the general framework section gives us a specific relation between the numerical fluxes $\overline{P}$ and $\boldsymbol{U}$ ensuring a local entropy inequality. Let us recall this fundamental identity on the cell Ω_c

$$\overline{P} = P_h^c - Z_c \, (\overline{\boldsymbol{U}} - \boldsymbol{U}_h^c) \cdot \boldsymbol{n}, \tag{4.146}$$

where $Z_c = \rho_c \, a_c$ is the acoustic impedance. The use of this expression to calculate $\boldsymbol{F}_{pc}^q$ yields

$$\begin{aligned} \boldsymbol{F}_{pc}^q &= P_{pc}^- \, l_{pc}^{-,q} \boldsymbol{n}_{pc}^- + P_{pc}^+ \, l_{pc}^{+,q} \boldsymbol{n}_{pc}^+, \\ &= P_h^c(\boldsymbol{X}_p, t) \, l_{pc}^q \boldsymbol{n}_{pc}^q \quad - Z_c \left[(\boldsymbol{U}_p - \boldsymbol{U}_h^c(\boldsymbol{X}_p, t)) \cdot \boldsymbol{n}_{pc}^- \right] l_{pc}^{-,q} \boldsymbol{n}_{pc}^- \\ &\qquad\qquad\qquad - Z_c \left[(\boldsymbol{U}_p - \boldsymbol{U}_h^c(\boldsymbol{X}_p, t)) \cdot \boldsymbol{n}_{pc}^+ \right] l_{pc}^{+,q} \boldsymbol{n}_{pc}^+. \end{aligned}$$

Finally, the qth moment of the nodal subcell force writes

$$\boldsymbol{F}_{pc}^q = P_h^c(\boldsymbol{X}_p, t) \, l_{pc}^q \boldsymbol{n}_{pc}^q - \mathsf{M}_{pc}^q \, (\boldsymbol{U}_p - \boldsymbol{U}_h^c(\boldsymbol{X}_p, t)), \tag{4.147}$$

where the M_{pc}^q matrices are defined as

$$\mathsf{M}_{pc}^q = Z_c \left(l_{pc}^{-,q} \boldsymbol{n}_{pc}^{q,-} \otimes \boldsymbol{n}_{pc}^- + l_{pc}^{+,q} \boldsymbol{n}_{pc}^{q,+} \otimes \boldsymbol{n}_{pc}^{q,+} \right). \tag{4.148}$$

Now, regarding the midpoint subcell forces, substituting (4.146) in the definition of $\boldsymbol{F}^q_{mc}$ leads to

$$\begin{aligned}\boldsymbol{F}^q_{mc} &= P_{mc}\, l^q_{mc}\boldsymbol{n}^q_{mc},\\ &= P^c_h(\boldsymbol{X}_m,t)\, l^q_{mc}\boldsymbol{n}^q_{mc} - Z_c\left((\boldsymbol{U}_m - \boldsymbol{U}^c_h(\boldsymbol{X}_m,t))\,\boldsymbol{.}\,\boldsymbol{n}_{mc}\right) l^q_{mc}\boldsymbol{n}^q_{mc}.\end{aligned}$$

At the end, the qth moment of the midpoint subcell force can write as follows

$$\boldsymbol{F}^q_{mc} = P^c_h(\boldsymbol{X}_m,t)\, l^q_{mc}\boldsymbol{n}^q_{mc} - \mathsf{M}^q_{mc}\,(\boldsymbol{U}_m - \boldsymbol{U}^c_h(\boldsymbol{X}_m,t)), \tag{4.149}$$

where the M^q_{mc} matrices are defined as

$$\mathsf{M}^q_{mc} = Z_c\, l^q_{mc}\boldsymbol{n}^q_{mc}\otimes\boldsymbol{n}_{mc}. \tag{4.150}$$

The total energy conservation impose, in the simple case with no boundary contribution

$$\begin{aligned}\sum_c\int_{\Omega_c}\rho^0\frac{\mathrm{d}\,E}{\mathrm{d}t}\,\mathrm{d}V &= -\sum_c\sum_{p\in\mathcal{P}(c)}\frac{1}{3}\boldsymbol{U}_p\,\boldsymbol{.}\,\boldsymbol{F}_{pc} - \sum_c\sum_{m\in\mathcal{M}(c)}\frac{2}{3}\boldsymbol{U}_m\,\boldsymbol{.}\,\boldsymbol{F}_{mc},\\ &= \sum_p\frac{1}{3}\boldsymbol{U}_p\,\boldsymbol{.}\sum_{c\in\mathcal{C}(p)}\boldsymbol{F}_{pc} + \sum_m\frac{2}{3}\boldsymbol{U}_m\,\boldsymbol{.}\,(\boldsymbol{F}_{mL}+\boldsymbol{F}_{mR}),\\ &= 0,\end{aligned}$$

where $\mathcal{C}(p)$ is the set of cells surrounding the p node, and Ω_L and Ω_R the two cells surrounding the midpoint m, refer to Figure 4.23.

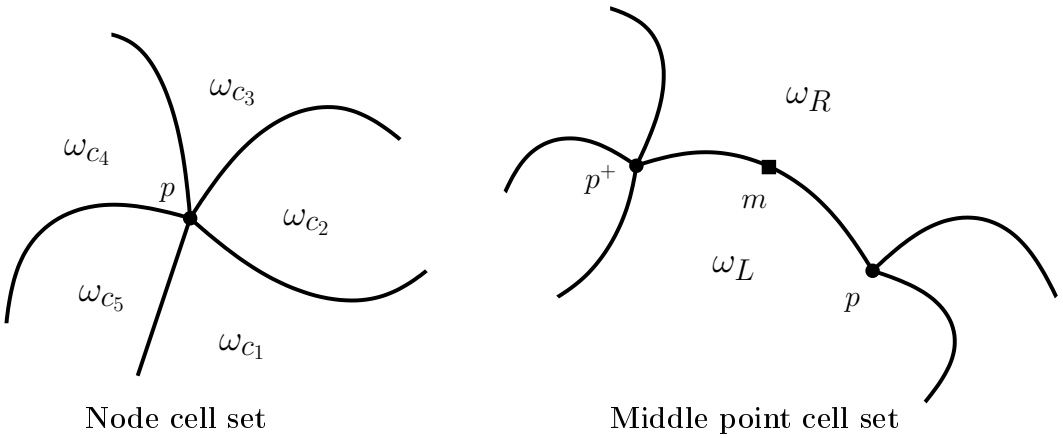


Figure 4.23: Node and midpoint cell set in the actual configuration.

To impose a total energy conservation, we set the following sufficient condition

$$\sum_{c\in\mathcal{C}(p)}\boldsymbol{F}_{pc} = \boldsymbol{0}, \tag{4.151}$$

$$\boldsymbol{F}_{mL} + \boldsymbol{F}_{mR} = \boldsymbol{0}. \tag{4.152}$$

This constraint also enforces the momentum conservation

$$\begin{aligned}\sum_c\int_{\Omega_c}\rho^0\frac{\mathrm{d}\,\boldsymbol{U}}{\mathrm{d}t}\,\mathrm{d}V &= -\sum_c\sum_{p\in\mathcal{P}(c)}\frac{1}{3}\boldsymbol{F}_{pc} - \sum_c\sum_{m\in\mathcal{M}(c)}\frac{2}{3}\boldsymbol{F}_{mc},\\ &= \sum_{p\in\mathcal{P}(\Omega)}\frac{1}{3}\sum_{c\in\mathcal{C}(p)}\boldsymbol{F}_{pc} + \sum_{m\in\mathcal{M}(\Omega)}\frac{2}{3}\,(\boldsymbol{F}_{mL}+\boldsymbol{F}_{mR}),\\ &= \boldsymbol{0}.\end{aligned}$$

Thanks to (4.151), we finally have an explicit expression of the nodal velocity $\boldsymbol{U}_p$

$$(\sum_{c\in\mathcal{C}(p)} M_{pc})\,\boldsymbol{U}_p = \sum_{c\in\mathcal{C}(p)} \left[P_h^c(\boldsymbol{X}_p,t)\,l_{pc}\boldsymbol{n}_{pc} + \mathsf{M}_{pc}\,\boldsymbol{U}_h^c(\boldsymbol{X}_p,t)\right], \tag{4.153}$$

where $\mathsf{M}_{pc} = Z_c\,(l_{pc}^+\boldsymbol{n}_{pc}^+\otimes\boldsymbol{n}_{pc}^+ + l_{pc}^-\boldsymbol{n}_{pc}^-\otimes\boldsymbol{n}_{pc}^-)$ are positive semi-definite matrices with a physical dimension of a density times a velocity.

Regarding the midpoint solver, the condition (4.152) imposes the midpoint pressure to be shared by the two surrounding cells Ω_L and Ω_R. This continuity property comes from the normal continuity. An identical procedure than the one used for the nodal solver leads the following definition of the midpoint velocity

$$\mathsf{M}_m\,\boldsymbol{U}_m = \mathsf{M}_m\left(\frac{Z_L\,\boldsymbol{U}_h^L(\boldsymbol{X}_m) + Z_R\,\boldsymbol{U}_h^R(\boldsymbol{X}_m)}{Z_L+Z_R}\right) - \frac{P_h^R(\boldsymbol{X}_m) - P_h^L(\boldsymbol{X}_m)}{Z_L+Z_R}\,l_{pp^+}\boldsymbol{n}_{pp^+}, \tag{4.154}$$

where the matrix $\mathsf{M}_m = \frac{1}{Z_L}\mathsf{M}_{mL} = \frac{1}{Z_R}\mathsf{M}_{mR}$, using the definition (4.150) in the case $q=0$, writes

$$\mathsf{M}_m = l_{pp^+}\boldsymbol{n}_{pp^+}\otimes\boldsymbol{n}_{pp^+}. \tag{4.155}$$

Obviously, the matrix is singular. The equation (4.154) gives the normal contribution of the midpoint velocity, such as

$$(\boldsymbol{U}_m\,.\,\boldsymbol{n}_{pp^+}) = \left(\frac{Z_L\,\boldsymbol{U}_h^L(\boldsymbol{X}_m) + Z_R\,\boldsymbol{U}_h^R(\boldsymbol{X}_m)}{Z_L+Z_R}\right).\boldsymbol{n}_{pp^+} - \frac{P_h^R(\boldsymbol{X}_m) - P_h^L(\boldsymbol{X}_m)}{Z_L+Z_R}. \tag{4.156}$$

This is nothing but the solution of the Riemann problem at the cell interface located at the midpoint m. Regarding the tangential contribution, we make the choice of

$$(\boldsymbol{U}_m\,.\,\boldsymbol{t}_{pp^+}) = \left(\frac{Z_L\,\boldsymbol{U}_h^L(\boldsymbol{X}_m) + Z_R\,\boldsymbol{U}_h^R(\boldsymbol{X}_m)}{Z_L+Z_R}\right).\boldsymbol{t}_{pp^+}, \tag{4.157}$$

where $\boldsymbol{t}_{pp^+}$ is the tangent vector located at the midpoint. This definition of the midpoint tangential velocity corresponds to the centered part of the midpoint normal definition (4.156). This definition ensures the continuity of the tangential velocity at the cell interfaces. Using (4.156) and (4.157), the midpoint velocity finally writes

$$\boldsymbol{U}_m = \frac{Z_L\,\boldsymbol{U}_h^L(\boldsymbol{X}_m) + Z_R\,\boldsymbol{U}_h^R(\boldsymbol{X}_m)}{Z_L+Z_R} - \frac{P_h^R(\boldsymbol{X}_m) - P_h^L(\boldsymbol{X}_m)}{Z_L+Z_R}\,\boldsymbol{n}_{pp^+}. \tag{4.158}$$

This is the midpoint velocity definition used for the numerical application. In [10], they make use of an upwind kind scheme on the tangential component of the velocity as follows

$$\begin{array}{ll}
\text{if } (\boldsymbol{U}_m\,.\,\boldsymbol{t}_{pp^+} > 0) & \boldsymbol{U}_m\,.\,\boldsymbol{t}_{pp^+} = \boldsymbol{U}_L(m)\,.\,\boldsymbol{t}_{pp^+},\\
\text{else if } (\boldsymbol{U}_m\,.\,\boldsymbol{t}_{pp^+} < 0) & \boldsymbol{U}_m\,.\,\boldsymbol{t}_{pp^+} = \boldsymbol{U}_R(m)\,.\,\boldsymbol{t}_{pp^+},\\
\text{else} & \boldsymbol{U}_m\,.\,\boldsymbol{t}_{pp^+} = \left(\frac{Z_L\,\boldsymbol{U}_h^L(\boldsymbol{X}_m)+Z_R\,\boldsymbol{U}_h^R(\boldsymbol{X}_m)}{Z_L+Z_R}\right).\boldsymbol{t}_{pp^+}.
\end{array}$$

This midpoint pressure continuity comes naturally from the parametrization of the pressure on the edge (4.141). An alternative procedure would be to consider the midpoint as a corner node. In this frame, we split the midpoint point force $\boldsymbol{F}_{mc}$ into two subcell forces $\boldsymbol{F}_{mc}^-$ and $\boldsymbol{F}_{mc}^+$ as follows

$$\boldsymbol{F}_{mc} = P_{mc}^- l_{pQ}^-\boldsymbol{n}_{pQ}^- + P_{mc}^+ l_{Qp^+}^+\boldsymbol{n}_{Qp^+}^+, \tag{4.159}$$

where P_{mc}^{-} and P_{mc}^{+} are the two midpoint pressures. This particular treatment of the pressure leads to a new definition of the matrix M_m writing

$$\mathsf{M}_m = l_{pQ}\boldsymbol{n}_{pQ} \otimes \boldsymbol{n}_{pQ} + l_{Qp^+}\boldsymbol{n}_{Qp^+} \otimes \boldsymbol{n}_{Qp^+}. \tag{4.160}$$

This matrix is invertible in the case where the control point Q is not aligned with the nodes p and p^+. Thus, a particular treatment is required on the straight edges of the mesh.

Stéphane Del Pino led to the same conclusion in his work on curvilinear Finite Volume method to solve compressible gas dynamics in semi-Lagrangian coordinates. In [87], he presents a Finite Volume method to solve compressible gas dynamics in semi- Lagrangian coordinates on curvilinear grids. His approach relies on a Finite Elements method based on the grid edges to compute the mesh velocity using an acoustic Riemann solver approximation. He concludes that this global continuous problem admits a unique solution if and only if the mesh does not present any straight-line edge.

Regarding the semi-discrete equation of the deformation gradient tensor in the triangles constituting the cells (4.119), we obviously need to define the velocity of the interior points, refer to Figure 4.24. Any linear assumption on the definition of these interior points velocity will lead to a second-order order scheme. Consequently, we set the velocity of an interior point q of cell Ω_c to the extrapolated value at this point of the polynomial approximation of the velocity inside the cell, *i.e.*, $\boldsymbol{U}_q = \boldsymbol{U}_h^c(\boldsymbol{X}_q)$ where $\boldsymbol{X}_q$ is the initial position of point q.

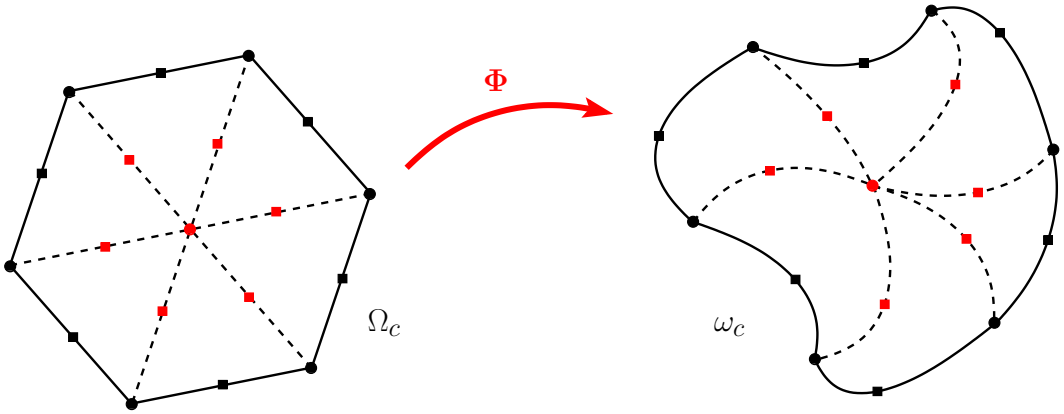


Figure 4.24: Transformation of Ω_c to ω_c through the flow. • boundary cell nodes, ■ boundary cell midpoints, • interior cell nodes, ■ interior cell midpoints.

The third-order scheme development is still under investigation. Some improvements would certainly lie in the definition of the node and midpoint solvers, or in the definition of the interior points velocity. A solution could be the use of the kinematic jump condition deriving from the Rankine-Hugoniot relation through a discontinuity of the deformation gradient tensor equation (4.2a). This jump relation writes

$$w[\![\mathsf{F}]\!] + [\![\boldsymbol{U}]\!] \otimes \boldsymbol{N}_w = 0, \tag{4.161}$$

where $[\![\mathsf{F}]\!]$ is the jump of the deformation gradient tensor through a discontinuity. $[\![\boldsymbol{U}]\!]$ corresponds to the velocity jump, $\boldsymbol{N}_w$ is the normal discontinuity and w its normal velocity. This is the so-called Hadamard compatibility condition [48]. This relation could possibly gives a new definition of the points velocity by means of the jump of the deformation gradient tensor.

In this whole scheme implementation, we have considered a null deformation of the initial mesh at time $t = 0$. Now, we present the procedure allowing the take into account of curve initial geometries.

4.3.5 Deformed initial mesh

Let Ω_c be a cell of the initial configuration mesh Ω at time $t = 0$, and ω_c its corresponding cell in the actual configuration domain ω at time t. We assume that the initial mesh presents curve geometries. In that case, the scheme discretization remain correct, one just has to initialize correctly the deformation gradient tensor and the mass matrix. Let the straight polygonal cell Ω_c^r be the referential cell of Ω_c, in the referential domain Ω^r, refer to Figure 4.25.

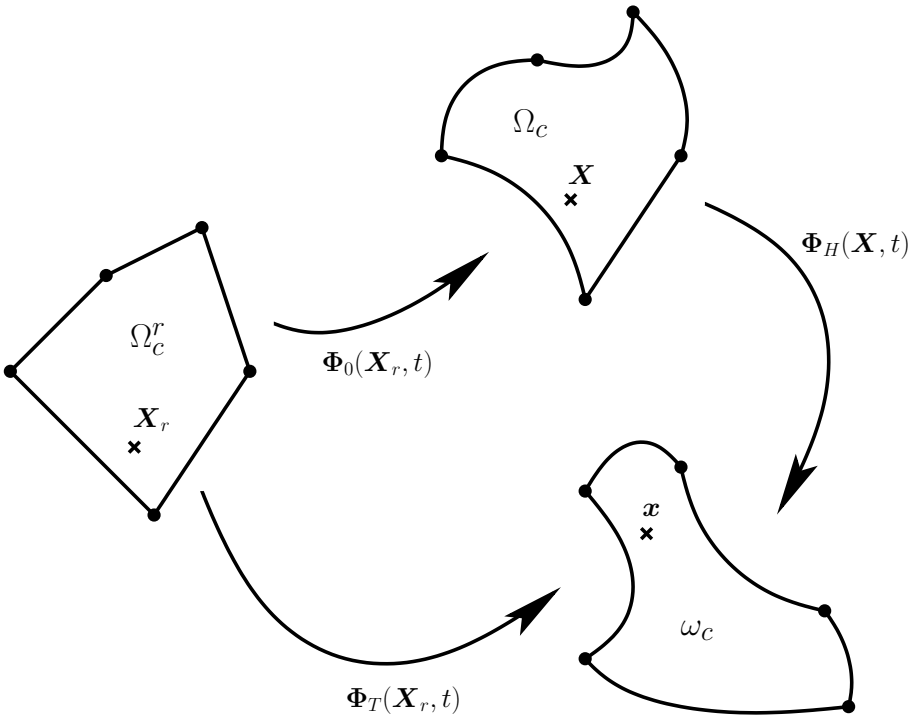


Figure 4.25: Reference, initial and actual configuration.

$\boldsymbol{X}$ denotes the initial position of the moving point located at time t at the position $\boldsymbol{x}$. The mathematical transformation $\boldsymbol{\Phi}_H(\boldsymbol{X},t)$ represents the displacement due to the flow motion of the fluid. $\boldsymbol{X}_r$ is the referential position of the point initially located at $\boldsymbol{X}$. $\boldsymbol{\Phi}_0(\boldsymbol{X}_r)$ represents the initial transformation of the domain and $\boldsymbol{\Phi}_T(\boldsymbol{X}_r,t)$ corresponds to the total deformation mapping, refer to Figure 4.25. We rewrite these definition with the use of the mathematical relation $\boldsymbol{\Phi}_T(\boldsymbol{X}_r,t) = \boldsymbol{\Phi}_H(\boldsymbol{X},t) \circ \boldsymbol{\Phi}_0(\boldsymbol{X}_r)$.

The use of the chain rule of composed derivatives and of the deformation gradient tensor definition gives the following relation

$$\begin{aligned} \mathsf{F}_T &= \nabla_{X_r} \boldsymbol{\Phi}_T(\boldsymbol{X}_r,t), \\ &= \nabla_X \boldsymbol{\Phi}_H(\boldsymbol{X},t) \circ \nabla_{X_r} \boldsymbol{\Phi}_0(\boldsymbol{X}_r), \\ &= \mathsf{F}_H \, \mathsf{F}_0, \end{aligned} \tag{4.162}$$

where $\mathsf{F}_0 = \nabla_{X_r} \boldsymbol{\Phi}_0(\boldsymbol{X}_r)$ and $\mathsf{F}_H = \nabla_X \boldsymbol{\Phi}_H(\boldsymbol{X}, t)$. Taking the determinant of the tensors on both sides of (4.162), this last formula yields

$$J_T(\boldsymbol{X}_r, t) = J_H(\boldsymbol{X}, t)\, J_0(\boldsymbol{X}_r), \tag{4.163}$$

where $J_T = \det \mathsf{F}_T$, $J_H = \det \mathsf{F}_H$ and $J_0 = \det \mathsf{F}_0$.
To recover the Lagrangian equation of continuity, we recall the mass conservation principle

$$\int_{\Omega_c} \rho^0(\boldsymbol{X})\, \mathrm{d}\Omega = \int_{\omega_c} \rho(\boldsymbol{x}, t)\, \mathrm{d}\omega.$$

Expressing the integrands in terms of the referential coordinate yields

$$\int_{\Omega_c^r} \rho^0(\boldsymbol{\Phi}_0(\boldsymbol{X}_r))\, J_0(\boldsymbol{X}_r)\, \mathrm{d}\Omega^r = \int_{\Omega_c^r} \rho(\boldsymbol{\Phi}_T(\boldsymbol{X}_r, t), t)\, J_T(\boldsymbol{X}_r, t)\, \mathrm{d}\Omega^r.$$

From this equation, it follows easily that

$$\rho^0\, J_0 = \rho\, J_T. \tag{4.164}$$

This new continuity equation only affects the construction of the mass matrix. Indeed, trying to calculate $\int_{\omega_c} \rho \frac{\mathrm{d}\,\psi}{\mathrm{d}t}\, \sigma_q\, \mathrm{d}\omega$, the time rate of change of the successive moments of the function ψ, one gets

$$\begin{aligned} \int_{\omega_c} \rho \frac{\mathrm{d}\,\psi}{\mathrm{d}t}\, \sigma_q\, \mathrm{d}\omega &= \int_{\Omega_c^r} \rho\, J_T \frac{\mathrm{d}\,\psi}{\mathrm{d}t}\, \sigma_q^c\, \mathrm{d}\Omega^r, \\ &= \int_{\Omega_c^r} \rho^0\, J_0 \frac{\mathrm{d}\,\psi}{\mathrm{d}t}\, \sigma_q^c\, \mathrm{d}\Omega^r. \end{aligned}$$

And interchanging the function ψ by its polynomial approximation, $\psi_h^c(\boldsymbol{X}_r, t) = \sum_{k=0}^K \psi_k^c(t) \sigma_k^c(\boldsymbol{X}_r)$, one gets

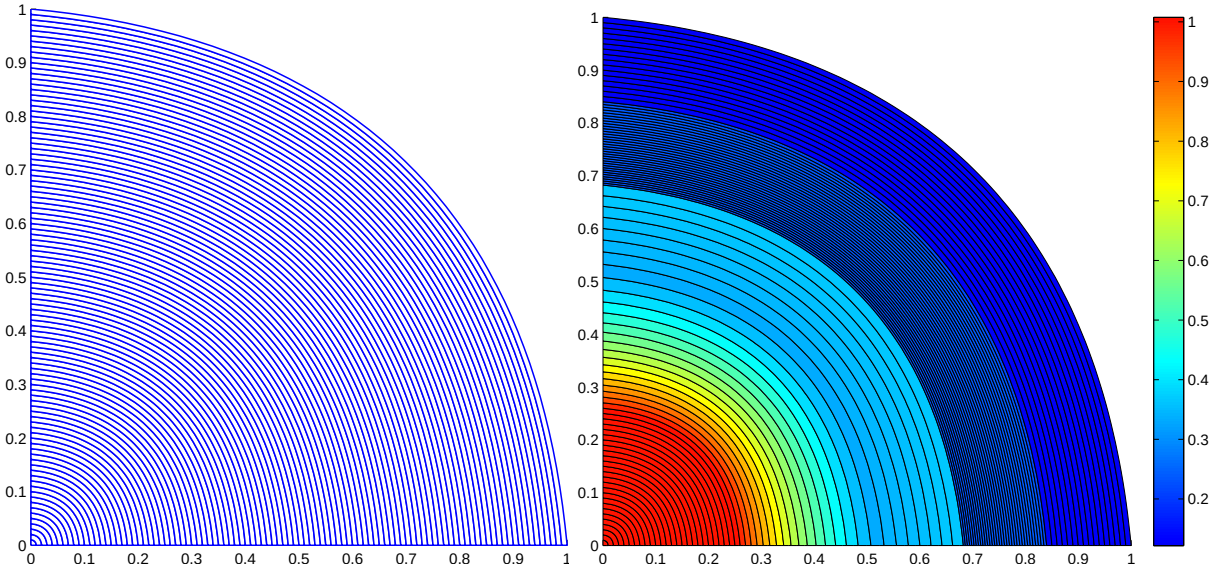
$$\int_{\omega_c} \rho \frac{\mathrm{d}\,\psi_h^c}{\mathrm{d}t}\, \sigma_q\, \mathrm{d}\omega = \sum_{k=0}^{K} \frac{\mathrm{d}\,\psi_k}{\mathrm{d}t} \int_{\Omega_c^r} \rho^0\, J_0\, \sigma_q\, \sigma_k\, \mathrm{d}\Omega^r. \tag{4.165}$$

Finally, in equation (4.165) we identify $\int_{\Omega_c^r} \rho^0\, J_0\, \sigma_q\, \sigma_k\, \mathrm{d}\Omega^r$ as the coefficients of the mass matrix. In the case of non-deformed initial geometry, $J_0(\boldsymbol{X}_r) = 1$. Thus, we recover the former discretization. Regarding the deformation gradient tensor related to the initial deformation, the piecewise linear matrix function F_0 is defined using (4.116). We give an example of initial curve meshes in the case of the Polar Sod shock tube problem, refer to Figure 4.26-(a), 4.27-(a) and 4.28-(a).

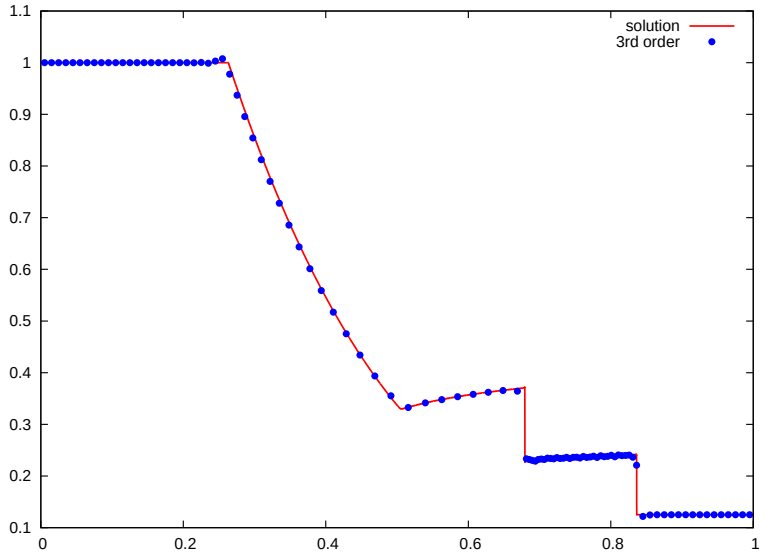
To complete the scheme development, we apply to all these semi-discretized equations a classical third-order TVD Runge-Kutta time discretization scheme [27]. The time step is evaluated following [27]. Thus, in this third-order frame, $C_e = \frac{1}{5}$. The slope limitation is performed using the Riemann invariants limitation procedure, refer to Section 4.1.4. Now, we present the preliminary results of validation.

4.3.6 Numerical results

Polar Sod shock tube problem. Here, we present the numerical solutions performed with our third-order DG scheme on the polar Sod shock tube problem described in 4.2.4. Thanks to Section 4.3.5, we start the computations with deformed initial meshes, refer to Figure 4.26-(a), 4.27-(a) and 4.28-(a).



(a) Initial grid. (b) Density map.

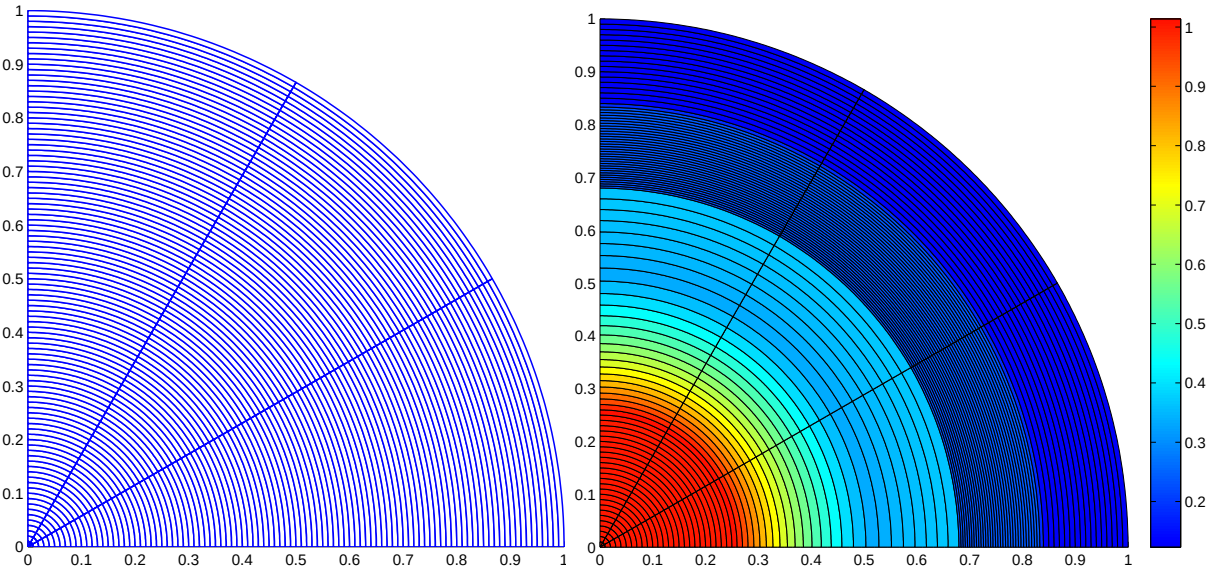


(c) Density profile.

Figure 4.26: Polar Sod shock tube problem. Grids and density map. Solution obtained with the third-order DG scheme without limitation on the domain defined in polar coordinates by $(r, \theta) \in [0, 1] \times [0, \frac{\pi}{2}]$ made of 100×1 cells.

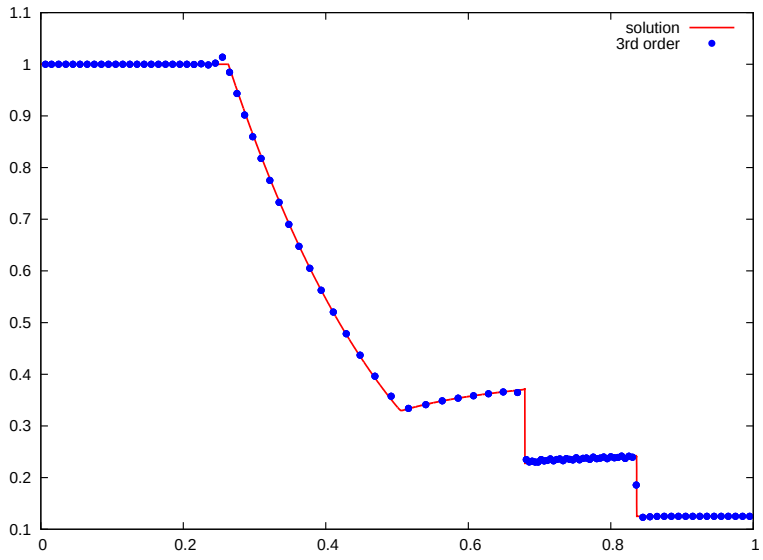
In the first case, the domain defined in polar coordinates by $(r, \theta) \in [0, 1] \times [0, \frac{\pi}{2}]$ is made of 100 cells in the radial direction but only 1 cell in the angular direction, refer to Figure 4.26-(a). Obviously, the mesh edges being parametrized by Bezier curves, the gird is not perfectly circular. Nevertheless, at the end of the computation, the numerical solution performed by the

third-order DG scheme exhibits a very good preservation of the polar criteria of the grid, refer to Figure 4.26-(b). We also note on Figure 4.26-(c) that the numerical solution is very similar to the one-dimensional cylindrical solution. This result proves the strong accuracy and robustness of the third-order scheme without limitation in this particular case of a grid made of only one cell in the radial direction.



(a) Initial grid.

(b) Density map.

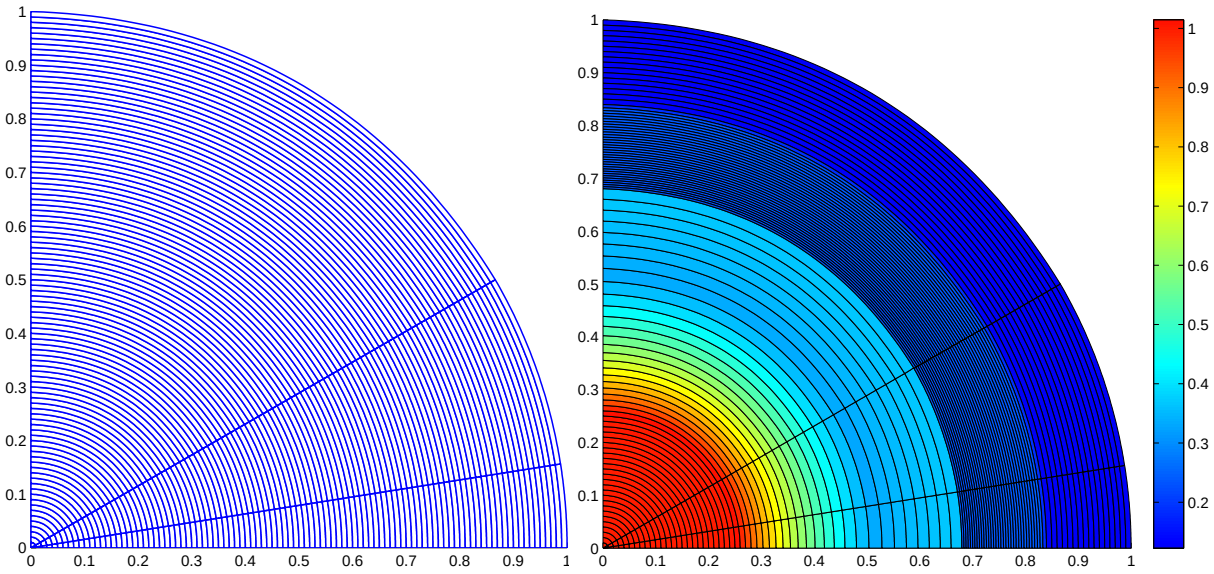


(c) Density profile.

Figure 4.27: Polar Sod shock tube problem. Grids and density map. Solution obtained with the third-order DG scheme without limitation on the domain defined in polar coordinates by $(r, \theta) \in [0, 1] \times [0, \frac{\pi}{2}]$ made of 100×3 cells.

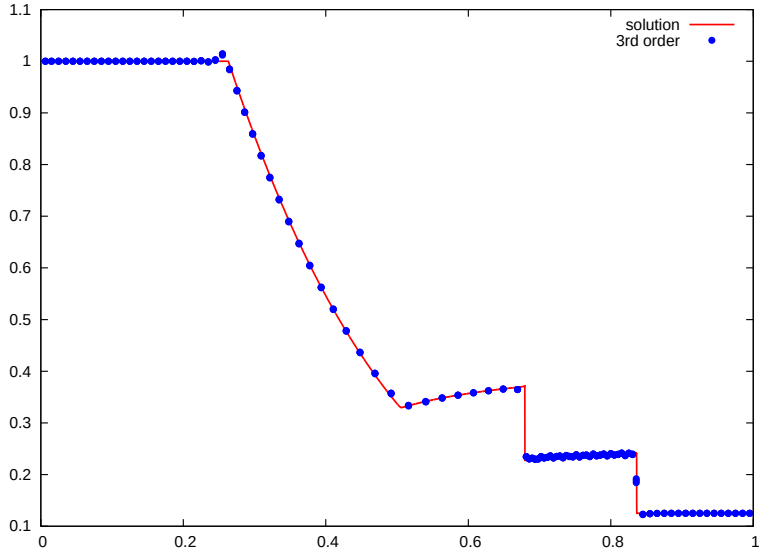
In the second case, the domain defined in polar coordinates by $(r, \theta) \in [0, 1] \times [0, \frac{\pi}{2}]$ is made of 100 cells in the radial direction and 3 in the angular direction, refer to Figure 4.27-(a). Refining the mesh in the angular direction allows a better representation of circular meshes. In this case, we observe the perfect preservation of the symmetry at the end of the computation,

refer Figure 4.27-(b). Moreover, the results displayed in Figure 4.27-(c) again assure the strong accuracy and robustness of the third-order scheme presented. Now, we present the same problem in the case of a non-uniform polar grid, refer to Figure 4.28-(a).



(a) Initial grid.

(b) Density map.

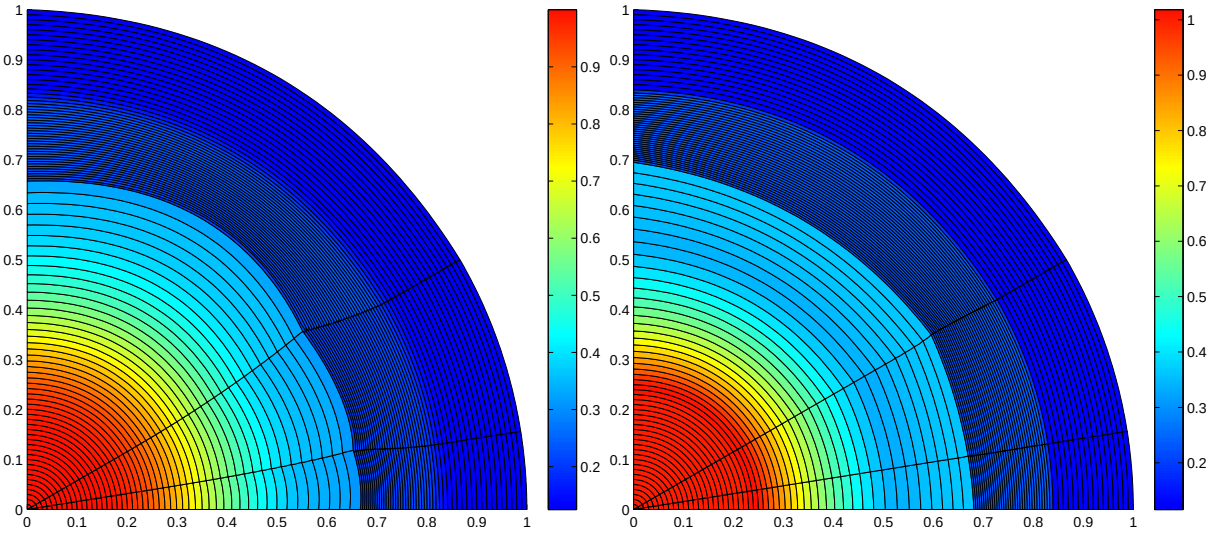


(c) Density profile.

Figure 4.28: Polar Sod shock tube problem. Grids and density map. Solution obtained with the third-order DG scheme without limitation on the domain defined in polar coordinates by $(r, \theta) \in [0, 1] \times [0, \frac{\pi}{2}]$ made of 100×3 non-uniform cells.

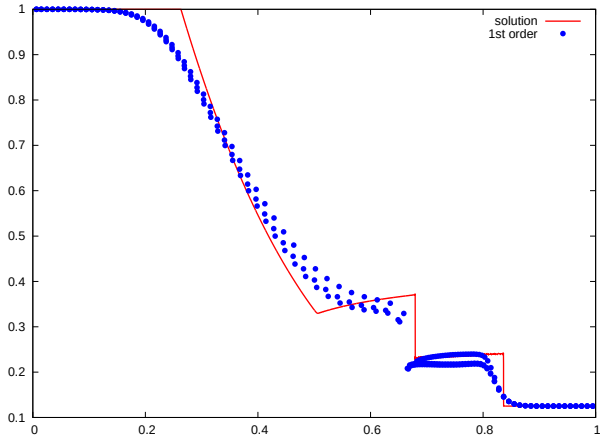
In this case, the domain defined in polar coordinates by $(r, \theta) \in [0, 1] \times [0, \frac{\pi}{2}]$ is made of 100 uniform cells of length $\frac{1}{100}$ in the radial direction and 3 non-uniform cells of length $\frac{\Pi}{20}$, $\frac{7\Pi}{60}$ and $\frac{\Pi}{3}$ in the angular direction, refer to Figure 4.28-(a). This non-uniformity in the mesh could cause serious problems in the preservation of the symmetry. However, the third-order scheme exhibits excellent results, refer to Figure 4.27-(b) and 4.28-(c). The symmetry preservation is due to the high accuracy of the third order scheme. Indeed, performing the same problem, on

the same initial mesh, with our first and second-order schemes, we clearly observe the loss of symmetry, refer to Figure 4.29-(a) and 4.29-(b).

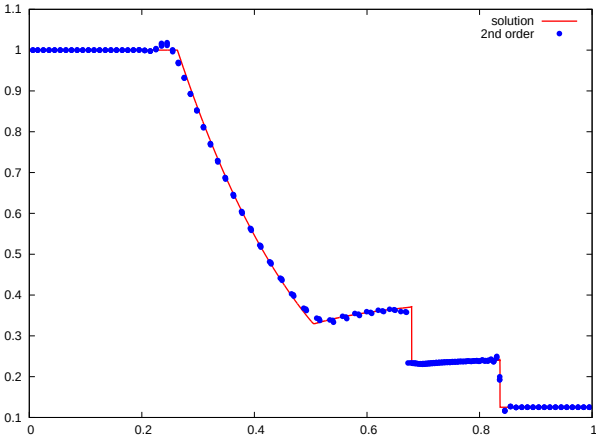


(a) First-order scheme.

(b) Second-order scheme.



(c) First-order scheme.

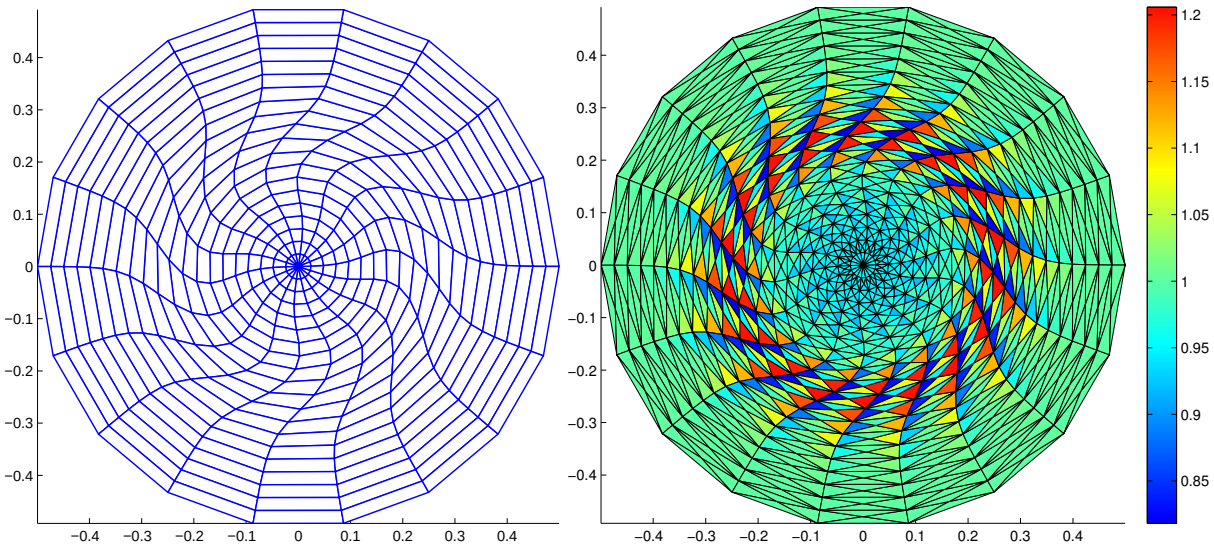


(d) Second-order scheme.

Figure 4.29: Polar Sod shock tube problem.Density maps and profiles. Solutions obtained with the first and second-order DG schemes on the domain defined in polar coordinates by $(r, \theta) \in [0, 1] \times [0, \frac{\pi}{2}]$ made of 100×3 non-uniform cells.

Variant of the Gresho vortex problem. Here, we display the numerical solutions computed for the alternative Gresho problem, refer to Section 4.2.4. The computational domain is defined in polar coordinates by $(r, \theta) \in [0, 1] \times [0, 2\pi]$. As we did in the second-order section, the solutions are displayed with a zoom in the zone $(r, \theta) \in [0, 0.5] \times [0, 2\pi]$. Here, we compare the efficiency of the first-order and second-order scheme presented in the Section 4.2 and the third-order scheme presented in this section. We know that if we run this Gresho problem far in time, some important difficulties will occur. On the one-hand, if the numerical scheme is too diffusive, the computation stops very early, and thus the final solution has nothing in common with the expected one. On the other hand, if the computation does not stop, grid deformation gets more and more important. At the end, the mesh is strongly deformed, and may present

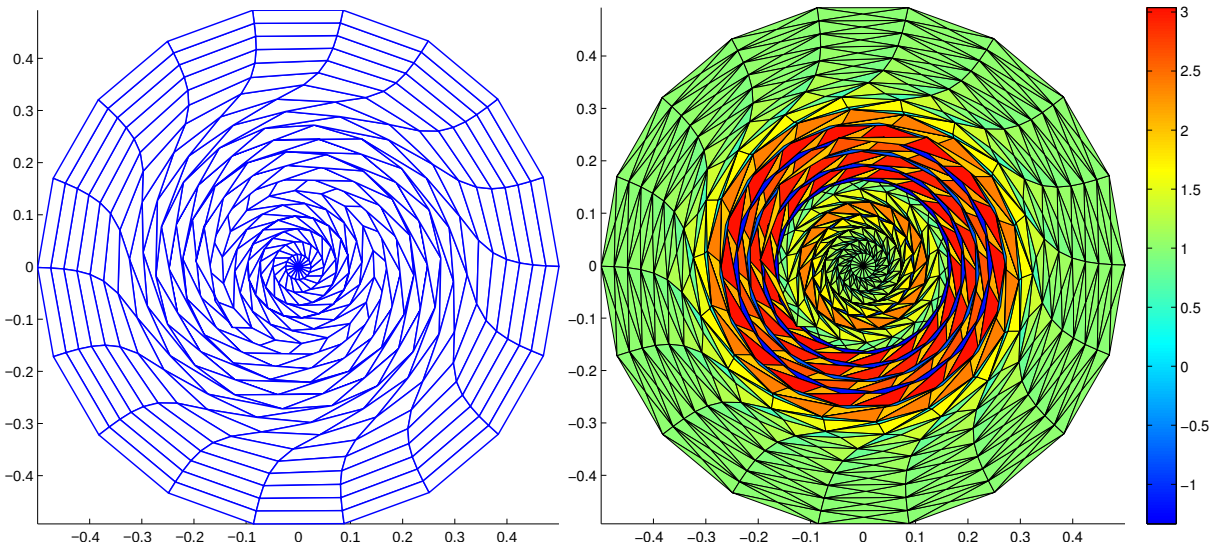
some cross points or negative Jacobians. To point out these phenomena, we run this Gresho problem until the final time $t = 1$, with our DG schemes with three different orders of accuracy.



(a) Grid deformation. (b) Jacobian.

Figure 4.30: Gresho problem variant on a polar grid made of 40×18 cells at the final time $t = 1$ with the first-order scheme.

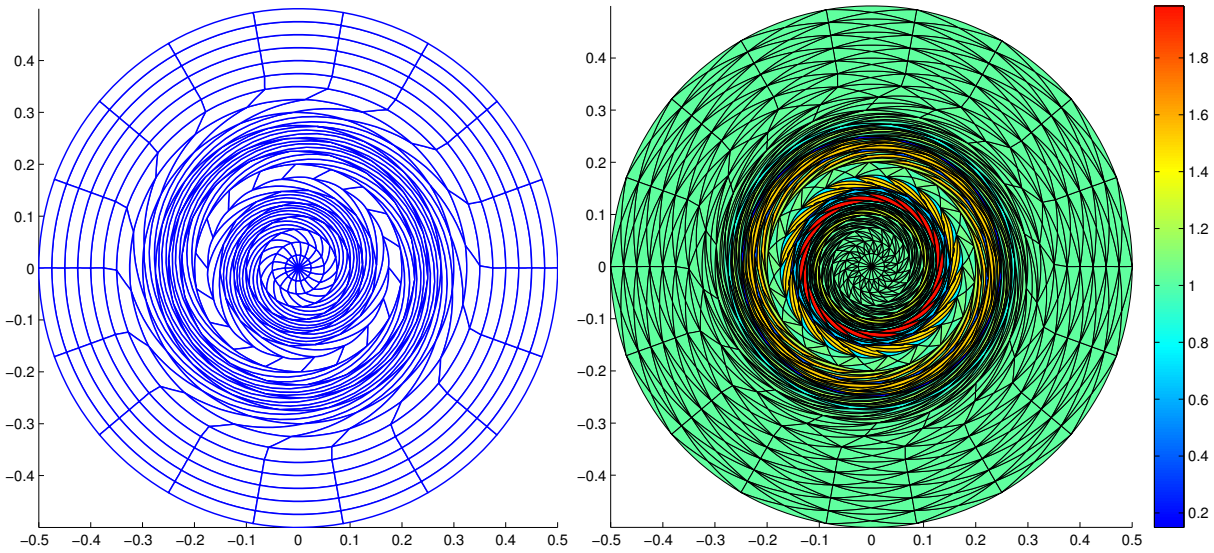
We start with the first-order scheme. On Figure 4.30-(a), we note that the grid is barely deformed. Comparing this result with the one obtained for a final time $t = 0.36$, see Figure 4.5-(a), we clearly see that these two results are identical. This means that at the time $t = 0.36$, the computation already stopped. This is due to the too important numerical diffusion. Obviously, the grid being slightly deformed, the mesh does not present any cross point, neither than negative Jacobian in the triangular cells, see Figure 4.30-(b).



(a) Grid deformation. (b) Jacobian.

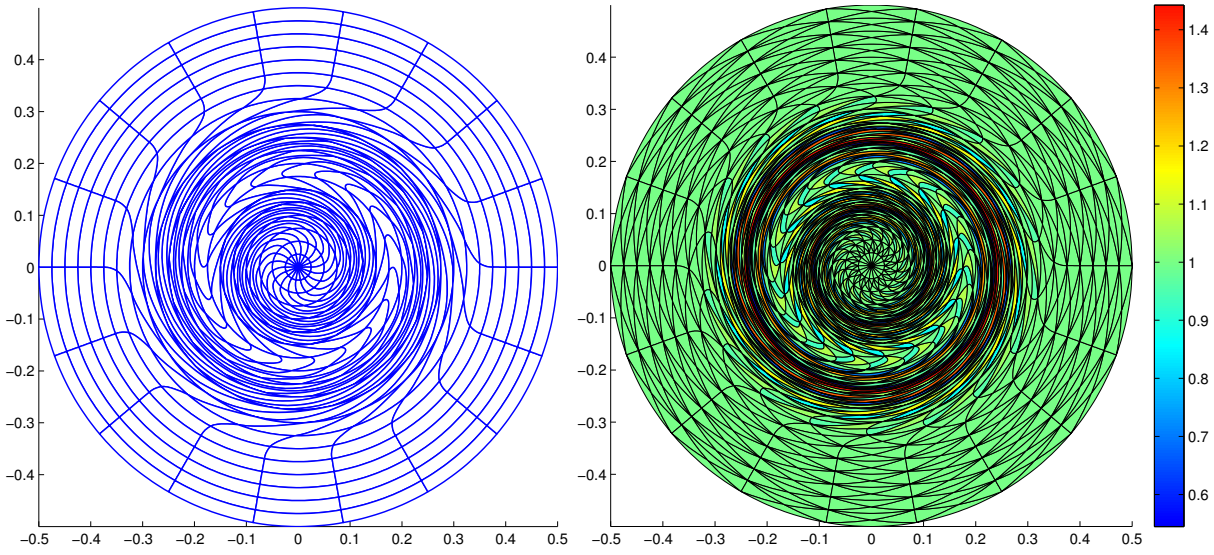
Figure 4.31: Gresho problem variant on a polar grid made of 40×18 cells at the final time $t = 1$ with the second-order scheme without limitation.

Now, with our second-order DG scheme, the solution presents another problem. This time, the computation does not stop before the final time. At the end of the computation, the grid is strongly deformed, and one can see the vortex structure inherent of this Gresho problem, refer to Figure 4.31-(b). However, the linear approximation as well as the straight-line representation do not allow us to follow properly the deformation. We note on Figure 4.31-(b) that the final solution presents cross points and negative Jacobians in some cells.



(a) Grid deformation. (b) Jacobian.

Figure 4.32: Gresho problem variant on a polar grid made of 40×18 cells at the final time $t = 1$ with the third-order scheme without limitation.



(a) Grid deformation. (b) Jacobian.

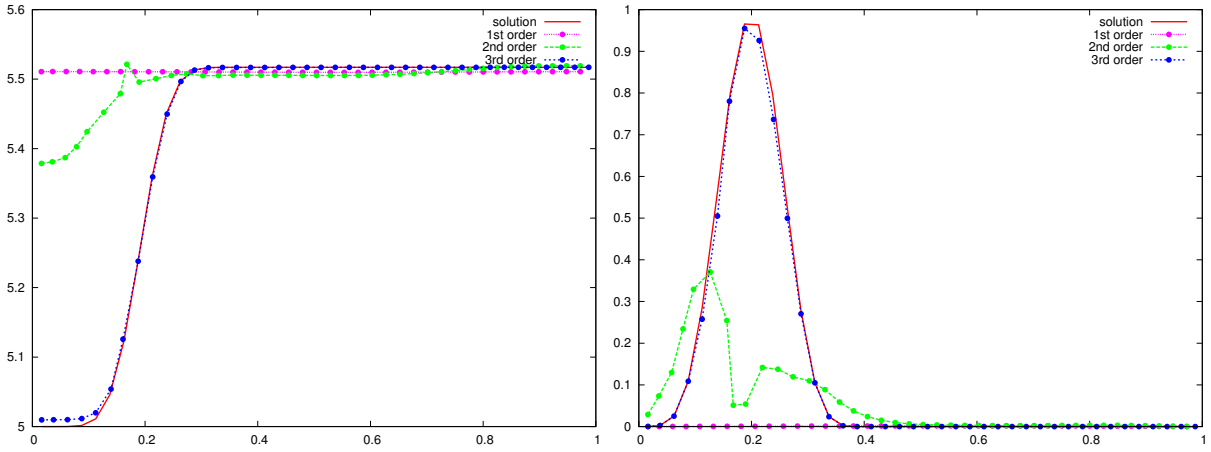
Figure 4.33: Exact solution of the Gresho problem variant on a polar grid made of 40×18 cells at the final time $t = 1$.

Finally, we perform this Gresho vortex problem with our third-order DG scheme on a curved

polar grid made of 40×18 as well. We also display in Figure 4.33 the exaction motion of this mesh to compare it with our numerical solutions.

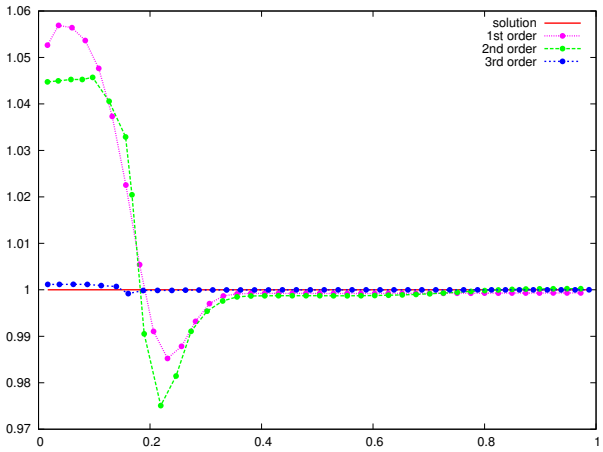
First, we note that as in the first-order and second-order cases, the third-order scheme preserves symmetry. This scheme being very few diffusive, the gird is extremely deformed at the end of the computation, more than in the second-order case, refer to Figure 4.32-(a). But thanks to the scheme properties and to the Bezier representation, the solution is very close to the expected one, see Figure 4.33-(a). Furthermore, the grid does not present any cross point, and the Jacobian of the triangular cells remain positive, see Figure 4.32-(b).

We have also displayed the plots corresponding to the pressure, the velocity and the density expressed at the centroid of the cells, in the three different cases presented, refer to Figure 4.34.



(a) Pressure profile.

(b) Velocity profile.



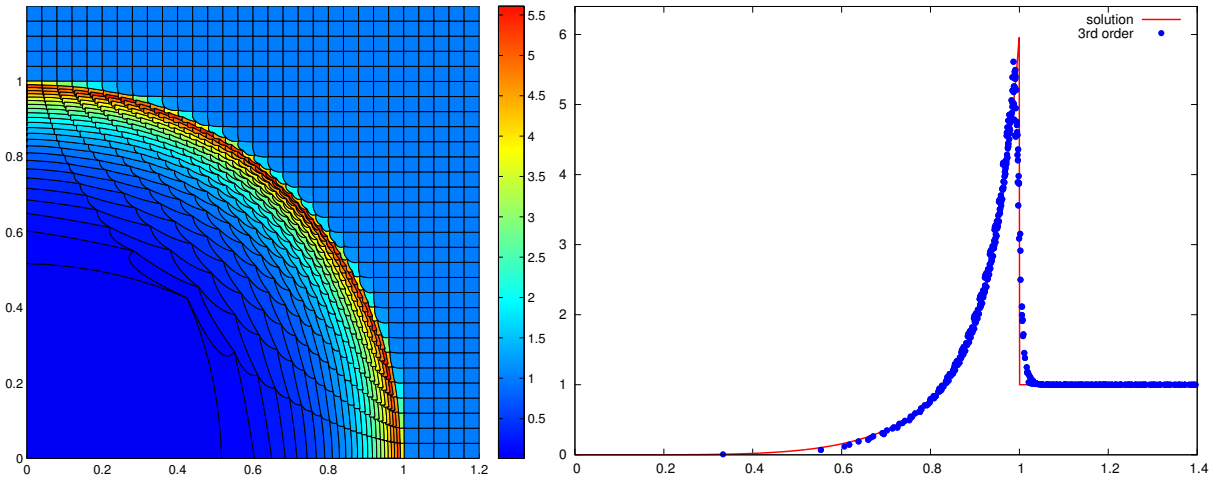
(c) Density profile.

Figure 4.34: Gresho problem variant on a polar grid made of 40×18 cells at the final time $t = 1$: comparison between the analytical solution and those obtained with the first, second and third-order numerical schemes, plotted at the centroid of the cells.

We note on Figure 4.34-(a) and 4.34-(b) that the numerical solutions corresponding to the first-order schemes are totally dissipated. We also observe the huge gain in accuracy between the second-order and the third-order DG scheme. Despite the extreme deformation of the grid, the numerical solutions obtained by the third-order scheme are very close to the exact solutions.

This confirm the high accuracy and robustness of the designed scheme. Now, regarding the incompressibility assumption, this test case deriving from a solution of the incompressible Euler equations, the density must remain equal to 1 during the calculation. We note on Figure 4.34-(c) that the result obtained with the third-order scheme is a lot better than the ones obtained with the first-order and second-order numerical schemes. At the end of the computation, even if the mesh is highly deformed, the incompressibility assumption is very well satisfied, the density being bounded in $\rho \in [0.9992, 1.0012]$.

Sedov point blast problem. We run the Sedov problem described in Section 4.2.4, with the third-order DG scheme with the limitation procedure based on the Riemann invariants polynomial approximation, described in the Section 4.1.4. The initial computational domain is defined by $(X, Y) = [0, 1.2] \times [0, 1.2]$ and paved by a 30×30 Cartesian grid.



(a) Density map.

(b) Density profile.

Figure 4.35: Point blast Sedov problem on a Cartesian grid made of 30×30 cells with the third-order scheme with limitation.

We note on Figure 4.35-(b) that the numerical solution is very close to the one-dimensional analytical solution. And despite the presence of cross edges, we see that the shock wave front is cylindrical and well located at the end of the computation, see Figure 4.35-(a). These results also demonstrate the robustness and the accuracy of this scheme. The overlapping of cells in the final mesh must derive from the midpoint velocity definition. Some improvements are certainly needed on this point.

We clearly see the mesh edges curving during the computation, refer to Figure 4.35-(a). This cells overlapping phenomenon may result from several reasons. The first one is the local criteria of our discretization. The scheme has been built such as the normals in the actual configuration are continuous on cells sharing a common face. But there is no continuity in the normals nor in the tangents between neighboring edges. The choice of Bezier curves to parametrize the mesh edges has been done to be consistent with the definition of the mapping using P_2 Finite Elements basis functions, these curves being the trace of the P_2 polynomials functions on the edges. It is clear that in some cases, this representation is not accurate enough, and thus bring geometric discontinuities. Considering the polar mesh depicted in Figure 4.27-(a), the Bezier representation does not allow us to obtain a circular mesh. The normals are discontinuous between edges in the angular direction. In [10], they use conics to parametrize their cells.

These functions allows a perfect representation of circular meshes, and the normals would be continuous in this case. However, these functions not corresponding to the trace of the P_2 Finite Elements basis functions of the edges, another discretization of the mapping and thus of the deformation gradient tensor is needed. This phenomenon could also derive from the limitation procedure. A better take into of curve geometries is probably required. Another possible explanation of this phenomenon may come from the midpoint velocity definition. The use of the Hadamard compatibility condition (4.161) and helped by the fact that the deformation gradient tensor is discontinuous between triangles inside the polygonal cells could bring more diffusion and thus more stability in the definition of the interior points velocity. At the end, an ALE approach may also be needed.

Taylor-Green vortex problem. We make use of the smooth Taylor-Green vortex test case described in Section 4.2.4, in order to perform a convergence analysis on our third-order DG scheme. But first, we compare the solutions obtained with our first-order, second-order and third-order discontinuous Galerkin schemes with the exact solution, on a Cartesian grid made of 10×10 cells. The results displayed in Figure 4.36 clearly show the huge gain in accuracy between the three numerical schemes, the numerical solution corresponding to the third-order scheme being very similar to the exact solution.

Now, knowing the analytical solution of this Taylor-Green problem, we compute the global truncation error corresponding to our second-order and third-order DG schemes and display it in Tables 4.6 and 4.7.

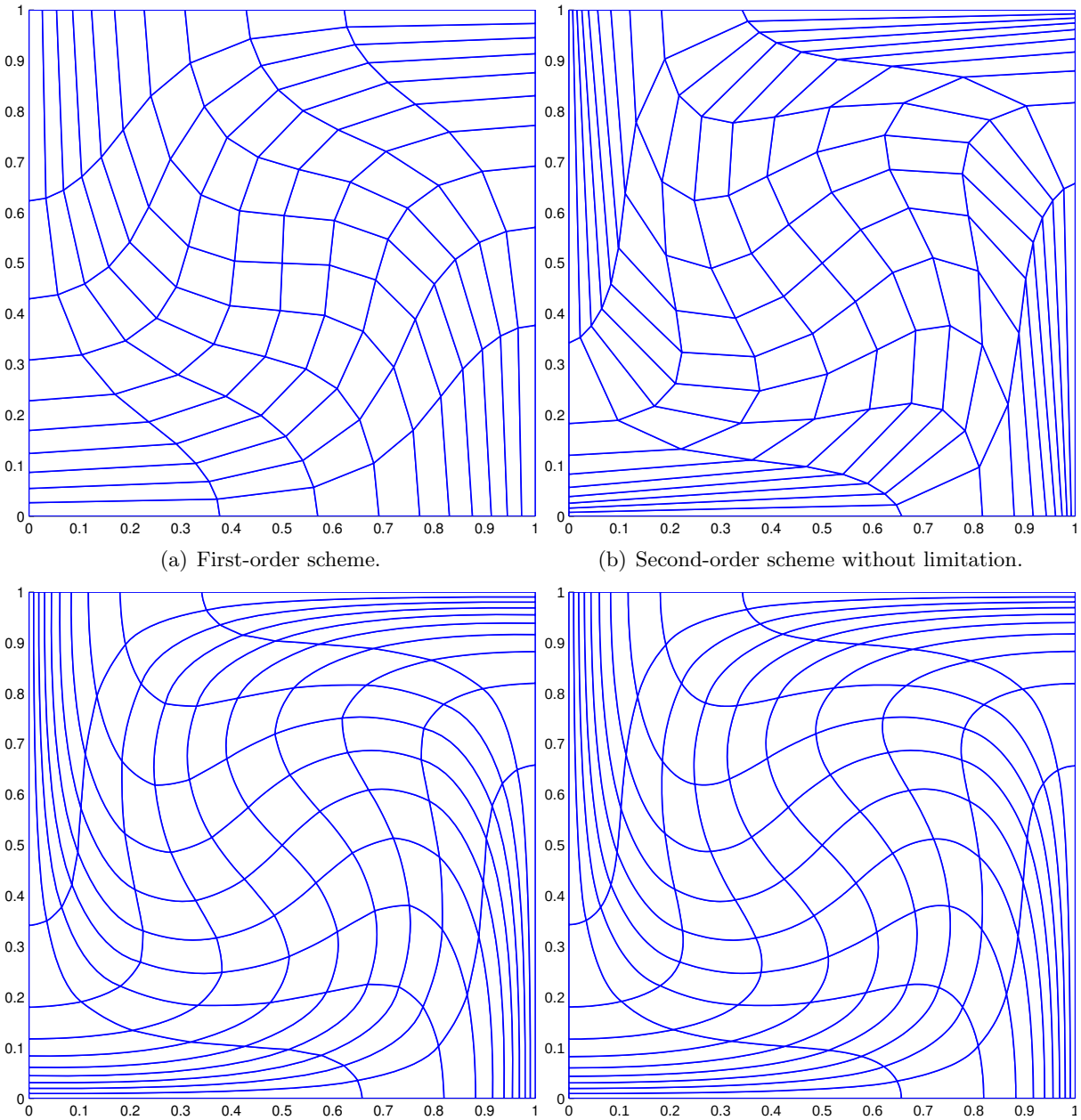
	L_1		L_2		L_∞	
h	$E^h_{L_1}$	$q^h_{L_1}$	$E^h_{L_2}$	$q^h_{L_2}$	$E^h_{L_\infty}$	$q^h_{L_\infty}$
$\frac{1}{10}$	2.50E-2	1.48	3.71E-2	1.30	1.72E-1	1.35
$\frac{1}{20}$	8.98E-3	1.88	1.51E-2	1.75	6.73E-2	1.27
$\frac{1}{40}$	2.44E-3	1.94	4.48E-3	1.95	2.79E-2	1.68
$\frac{1}{80}$	6.36E-4	2.00	1.16E-3	2.00	8.68E-3	1.95
$\frac{1}{160}$	1.59E-4	-	2.90E-4	-	2.24E-3	-

Table 4.6: Rate of convergence computed on the pressure in the case of the Taylor-Green vortex at time $\boldsymbol{t = 0.6}$, with the ***second-order*** DG scheme without limitation.

	L_1		L_2		L_∞	
h	$E^h_{L_1}$	$q^h_{L_1}$	$E^h_{L_2}$	$q^h_{L_2}$	$E^h_{L_\infty}$	$q^h_{L_\infty}$
$\frac{1}{10}$	4.39E-3	3.00	7.73E-3	2.68	3.90E-2	1.93
$\frac{1}{20}$	5.50E-4	3.04	1.21E-3	3.10	1.03E-2	2.98
$\frac{1}{40}$	6.68E-5	2.91	1.40E-4	2.87	1.30E-3	2.66
$\frac{1}{80}$	8.90E-6	2.89	1.92E-5	2.83	2.11E-4	2.74
$\frac{1}{160}$	1.20E-6	-	2.70E-6	-	3.16E-5	-

Table 4.7: Rate of convergence computed on the pressure in the case of the Taylor-Green vortex at time $\boldsymbol{t = 0.6}$, with the ***third-order*** DG scheme without limitation.

Comparing the results gathered in Tables 4.6 and 4.7, we conclude that as expected, the third-order scheme is a lot more accurate than the second-order scheme. We notice that in the case



(a) First-order scheme.

(b) Second-order scheme without limitation.

(c) Third-order scheme without limitation.

(d) Exact solution.

Figure 4.36: Taylor-Green vortex deformation of a Cartesian grid made of 10×10 cells, at time $t = 0.75$.

of the third-order scheme, the rate of convergences are close to three. These results confirm that our DG scheme reaches the correct order of accuracy. Now, performing the same rate of convergence analysis, for a lower final time corresponding to a less deformed final configuration, the results are even more relevant, the asymptotic regime being reached, refer to Table 4.8.

In this chapter, we have presented cell-centered DG discretizations using Taylor basis for solving two-dimensional gas dynamics equations on general unstructured grids, using the Lagrangian coordinates related to the initial configuration of the flow. In this frame, the mesh displace-

	L_1		L_2		L_∞	
h	$E^h_{L_1}$	$q^h_{L_1}$	$E^h_{L_2}$	$q^h_{L_2}$	$E^h_{L_\infty}$	$q^h_{L_\infty}$
$\frac{1}{10}$	2.67E-4	2.96	3.36E-7	2.94	1.21E-3	2.86
$\frac{1}{20}$	3.43E-5	2.97	4.36E-5	2.96	1.66E-4	2.93
$\frac{1}{40}$	4.37E-6	2.99	5.59E-6	2.98	2.18E-5	2.96
$\frac{1}{80}$	5.50E-7	2.99	7.06E-7	2.99	2.80E-6	2.99
$\frac{1}{160}$	6.91E-8	-	8.87E-8	-	3.53E-7	-

Table 4.8: Rate of convergence computed on the pressure in the case of the Taylor-Green vortex at time $\boldsymbol{t = 0.1}$, with the ***third-order*** DG scheme without limitation.

ment and deformation are followed by the deformation gradient tensor we chose to discretize on triangles with the use of finite element basis functions. The GCL and the Piola compatibility condition are satisfied by construction of the scheme. We build our nodal solvers, and midpoint solvers in the third-order case, imposing a correct entropy production and total energy conservation. Having developed properly and in a continuous manner the geometry, we can observe and display the solutions on the actual mesh. A robust and accurate limitation procedure on the polynomial approximation of the Riemann invariants has been constructed. The strong scheme robustness and accuracy have been assessed using several relevant test cases.

Now, in the case of very strong shock wave, some improvements on the limitation procedure could be needed to avoid the cross edges phenomenon observed in the Sedov test case, see Figure 4.35-(a). In future, we may be also interesting to investigate the extension of the present DG discretization to a higher geometric representation as the conics or P_3 Finite Elements basis functions, or to an ALE approach.

Chapter 5

Conclusion and perspectives

The aim of this work was to present a high-order discontinuous Galerkin scheme for solving the gas dynamics equations written under total Lagrangian form on two-dimensional unstructured grids. To this end, we have developed a progressive approach to study the numerical difficulties inherent step by step.

First, we have investigated the development of a third-order DG method applied to scalar conservation laws in one and two dimensions. The main feature of the presented DG scheme lies on the use of a Taylor basis [71]. This particular choice allows in the two-dimensional case to take into account in a unified framework general unstructured grids. In this frame, a vertex-based hierarchical limitation which preserves smooth extrema has been implemented [60, 109]. A generic form of numerical fluxes ensuring the global stability of our semi-discrete discretization in the L_2 norm has also been designed, following [52, 24, 98, 49].

Second, we applied this approach to the case of one-dimensional systems. Noticing that the application of the limiting procedure, developed for scalar equations, to the physical variables leads to spurious oscillations, we have described a limiting procedure based on the characteristic variables following the approach given by B. Cockburn and C.-W. Shu [29]. In the case of the one-dimensional gas dynamics case, numerical fluxes have been designed so that our semi-discrete DG scheme satisfies a global entropy inequality. The work done on the development of DG scheme for one and two-dimensional scalar conservations laws and for one-dimensional systems led to the publication of an article [104].

Then, after having briefly described the important concepts related to kinematics of fluid motion such as the Lagrangian and Eulerian formalisms, we have recall the classical Finite Volume EUCCLHYD scheme [78, 79, 74, 73, 75, 76]. This particular scheme in its first-order version is the corner stone of our DG formulation presented. We emphasize that in its first-order version, our DG method naturally recovers the first-order EUCCLHYD scheme.

Afterwards, gathering this knowledge we apply this methodology to the case of the two-dimensional gas dynamics equation written under total Lagrangian form. This work can be seen as the natural extension of [70, 1]. In this particular framework, the computational grid is fixed, however one has to follow the time evolution of the Jacobian matrix associated to the Lagrange-Euler flow map namely the gradient deformation tensor [56, 55]. In a first time, we present the general framework of the discontinuous Galerkin discretization of the gas dynamics equations in the Lagrangian formalism, written using the Lagrangian coordinates. Here, the flow map is discretized by means of continuous mapping, using a finite element basis. This

provides an approximation of the deformation gradient tensor which satisfies the Piola identity. The discretization of the physical conservation laws for the momentum and the total energy relies on a discontinuous Galerkin method. Then, the particular cases of the second-order and third-order schemes are detailed.

In the former, the mapping deriving from the flow motion of the fluid is discretized by means of P_1 barycentric coordinate basis functions. This particular description of the flow mapping leads to have the Piola compatibility condition and the geometry continuity ensured by construction of the scheme. The physical conservation laws are discretized through the use of a DG formulation wherein as specific treatment of the boundary terms is performed ensuring the respect of the GCL condition. The second-order DG scheme robustness and accuracy have been assessed using several relevant test cases taken from the literature. This particular scheme has proved to be highly accurate, exhibiting good results on a large number of test cases. It also appears to be more accurate than the second-order Finite Volume scheme. This second-order cell-centered DG scheme was the purpose of a second article [103].

Finally, we extend the second-order DG scheme presented to the third-order of accuracy. In this particular description, the flow mapping is discretized by means of P_2 Finite Elements basis functions. Thus, the geometry in the actual configuration is parametrized by Bezier curves. This time, the DG discretization of the physical variables requires a midpoint solver in addition to the nodal solver [10, 87]. This scheme is constructed ensuring the respect of the GCL. This third-order development being still under investigation, only early results are presented. Nevertheless, the first results confirm the very high accuracy of the implemented scheme. And the rate of convergence analysis done in the smooth case of a Taylor-Green vortex assures that the correct order of accuracy is reached.

In future, it would be interesting to investigate further a proper limitation procedure on generic Bezier cells. Alternative velocity solvers could also be designed using the jump of deformation gradient tensor between triangles, through the use of the Hadamard compatibility condition (4.161). And finally, the next step may certainly be an ALE approach, not regarding the shape of the moving mesh but its deformation gradient tensor F.

Appendix A

Notations and reminders related to vectors and tensors

In this appendix, we aim at recalling notation and formulas related to vectors and tensors which are required to derive the mathematical modeling of fluid mechanics. We do not intend to present the complete rigorous tensor theory, which is beyond the scope of the present chapter, but rather introduce useful formulas. The present description has been constructed collecting materials from [9, 96, 11, 91, 40, 64].
In all what follows, we consider $\mathbb{R}^d$ the d-dimensional Euclidean space, where d is a nonnegative integer ranging from 1 to 3. Namely, $\mathbb{R}^d$ is a d-dimensional vector space equipped with an inner product and an orthonormal basis $(\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_d)$. An arbitrary vector in $\mathbb{R}^d$, $\boldsymbol{x}$, is defined by its coordinates, $(x_1, x_2, \ldots, x_d)$, as $\boldsymbol{x} = \sum_{i=1}^{d} x_i \boldsymbol{e}_i$. The inner product of two vectors $\boldsymbol{x}$ and $\boldsymbol{x'}$ is defined by $\boldsymbol{x} \cdot \boldsymbol{x'} = \sum_{i=1}^{d} x_i x'_i$. The inner product of $\boldsymbol{x}$ with itself is always nonnegative and allows us to define the Euclidean norm on $\mathbb{R}^d$ as

$$\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x} \cdot \boldsymbol{x}} = \sqrt{\sum_{i=1}^{d} x_i^2}.$$

A.1 Introduction to second-order tensors

A.1.1 Definitions

Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be two vectors of $\mathbb{R}^d$. The tensor product, otherwise named dyadic product, of $\boldsymbol{a}$ and $\boldsymbol{b}$ is denoted by $\boldsymbol{a} \otimes \boldsymbol{b}$ and defined as

$$\forall \boldsymbol{x} \in \mathbb{R}^d \, , (\boldsymbol{a} \otimes \boldsymbol{b})\boldsymbol{x} = (\boldsymbol{b} \cdot \boldsymbol{x})\boldsymbol{a}. \tag{A.1}$$

This relation defines a linear transformation of $\mathbb{R}^d$ which is characterized by a matrix whose components are written

$$(\boldsymbol{a} \otimes \boldsymbol{b})_{ij} = a_i b_j, \ (i, j = 1, 2 \ldots, d). \tag{A.2}$$

Having in mind the notion of tensor product of two vectors, we define a second-order tensor on $\mathbb{R}^d$ as the sum of tensor products of vectors in $\mathbb{R}^d$. This definition involves that the set of second-order tensors defined on $\mathbb{R}^d$ is a d^2-dimensional vector space equipped with the basis $\boldsymbol{e}_i \otimes \boldsymbol{e}_j$ $(i, j = 1, 2 \ldots, d)$. This definition involves that an arbitrary tensor, T, can be written

$$\mathsf{T} = \sum_{i=1}^{d} \sum_{j=1}^{d} T_{ij} \boldsymbol{e}_i \otimes \boldsymbol{e}_j, \tag{A.3}$$

where T_{ij} are components of the tensor T. Let us remark that to any second-order tensor T corresponds the linear transformation of $\mathbb{R}^d$ defined by

$$\boldsymbol{x'} = \mathsf{T}\boldsymbol{x}, \quad x'_i = \sum_{j=1}^{d} T_{ij} x_j.$$

Note that the second-order tensor T corresponds either to the previous linear transformation or to the matrix defined by its components T_{ij} in an orthonormal basis. We conclude this paragraph recalling the following basic definitions which coincide with the classical definition of linear algebra for matrices.

- Zero tensor: this tensor, which is denoted by O, maps every vector $\boldsymbol{x}$ into the zero vector

$$\mathsf{O}\boldsymbol{x} = \mathbf{0}. \tag{A.4}$$

- Identity or unit tensor: this tensor, which is denoted by I_d, maps every vector into itself

$$\mathsf{I}_d\boldsymbol{x} = \boldsymbol{x}. \tag{A.5}$$

  The components of the matrix associated to the unit tensor are given by $I_{ij} = \delta_{ij}$, where δ_{ij} stands for the Kronecker symbol which takes the values $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$.

- Transpose of a tensor: the transpose of the tensor T is the unique tensor T^t defined by

$$\mathsf{T}\boldsymbol{x} \cdot \boldsymbol{x'} = \boldsymbol{x} \cdot \mathsf{T}^t\boldsymbol{x'}. \tag{A.6}$$

  Components of transpose of T are given by $T^t_{ij} = T_{ji}$.

- Symmetric and skew tensors: a tensor T is symmetric if $\mathsf{T}^t = \mathsf{T}$, it is skew if $\mathsf{T}^t = -\mathsf{T}$. Every tensor can expressed uniquely as the sum of a symmetric tensor and a skew tensor as

$$\mathsf{T} = \frac{1}{2}(\mathsf{T} + \mathsf{T}^t) + \frac{1}{2}(\mathsf{T} - \mathsf{T}^t). \tag{A.7}$$

- Trace of a tensor: the trace of a tensor T is the scalar denoted by $\mathrm{tr}(\mathsf{T})$ and defined as

$$\mathrm{tr}(\mathsf{T}) = \sum_{i=1}^{d} T_{ii}. \tag{A.8}$$

  If S and T are two second-order tensors, it is straightforward to check that

$$\mathrm{tr}(\mathsf{ST}) = \mathrm{tr}(\mathsf{TS}). \tag{A.9}$$

- Determinant of a tensor: the determinant of a tensor T is the scalar denoted by $\det \mathsf{T}$ and defined as the determinant of its corresponding matrix. It is computed by means of the following formula

$$\det \mathsf{T} = \sum_{i=1}^{d} T_{ij}(-1)^{i+j} M_{ij} = \sum_{j=1}^{d} T_{ij}(-1)^{i+j} M_{ij}, \tag{A.10}$$

  where M_{ij} denote the minors of matrix T, that is the determinant formed by omitting the ith row and jth column of T.

- Cofactor of a tensor: the cofactor matrix of a tensor T is denoted by $\mathrm{cof}(\mathsf{T})$ and defined as

$$\mathrm{cof}(\mathsf{T}) = J\mathsf{T}^{-\mathrm{t}}, \tag{A.11}$$

  where j is the determinent of T, *i.e.*, $J = \det \mathsf{T}$.

- Inverse of a tensor: the inverse of a tensor T is the tensor denoted by T^{-1}. It is defined provided that $\det \mathsf{T} \neq O$. It satisfies $\mathsf{T}\mathsf{T}^{-1} = \mathsf{I}_d$ and its components are given by

$$T^{-1}_{ij} = \frac{1}{\det \mathsf{T}}(-1)^{i+j} M_{ji}. \tag{A.12}$$

- Positive definite tensor: a tensor T is positive definite if for an arbitrary vector $\boldsymbol{x}$ the following property holds

$$\mathsf{T}\boldsymbol{x} \centerdot \boldsymbol{x} > 0. \tag{A.13}$$

A.1.2 Properties of the dyadic product of two vectors

The dyadic product of two vectors satisfies the following important properties

$$\mathrm{tr}(\boldsymbol{a} \otimes \boldsymbol{b}) = \boldsymbol{a} \centerdot \boldsymbol{b}, \tag{A.14a}$$
$$(\boldsymbol{a} \otimes \boldsymbol{b})^t = \boldsymbol{b} \otimes \boldsymbol{a}, \tag{A.14b}$$
$$\mathsf{T}(\boldsymbol{a} \otimes \boldsymbol{b}) = (\mathsf{T}\boldsymbol{a}) \otimes \boldsymbol{b}, \tag{A.14c}$$
$$(\boldsymbol{a} \otimes \boldsymbol{b})\mathsf{T} = \boldsymbol{a} \otimes (\mathsf{T}^t\boldsymbol{b}), \tag{A.14d}$$
$$(\boldsymbol{x} \otimes \boldsymbol{x'})(\boldsymbol{a} \otimes \boldsymbol{b}) = (\boldsymbol{x'} \centerdot \boldsymbol{a})\boldsymbol{x} \otimes \boldsymbol{b}, \tag{A.14e}$$

The demonstration of these properties can be found in [48].

A.1.3 Inner product of second-order tensors

Let S and T be two second-order tensors. The inner product of S and T is a scalar denoted by $\mathsf{S} : \mathsf{T}$ and defined as

$$\mathsf{S} : \mathsf{T} = \mathrm{tr}(\mathsf{S}^t\mathsf{T}). \tag{A.15}$$

The inner product can also be expressed in terms of the tensor components as

$$\mathsf{S} : \mathsf{T} = \sum_i \sum_j S_{ij} T_{ij}. \tag{A.16}$$

We note that $\mathsf{S} : \mathsf{T} = \mathsf{T} : \mathsf{S}$ and $\mathsf{S} : \mathsf{I}_d = \mathrm{tr}(\mathsf{S})$. This inner product defines a scalar product over the space of second-order tensor. The corresponding norm is the Frobenius norm defined as

$$\|\mathsf{T}\| = \sqrt{\mathrm{tr}(\mathsf{T}^t\mathsf{T})} = \sqrt{\sum_i \sum_j T^2_{ij}}. \tag{A.17}$$

Using the previous definition, one can show easily the following identities

$$\mathsf{L} : (\mathsf{S}\mathsf{T}) = (\mathsf{S}^t\mathsf{L}) : \mathsf{T} = (\mathsf{L}\mathsf{T}^t) : \mathsf{S}, \tag{A.18a}$$
$$(\boldsymbol{x} \otimes \boldsymbol{x'}) : (\boldsymbol{a} \otimes \boldsymbol{b}) = (\boldsymbol{x} \centerdot \boldsymbol{a})(\boldsymbol{x'} \centerdot \boldsymbol{b}), \tag{A.18b}$$
$$\mathsf{L} : (\boldsymbol{a} \otimes \boldsymbol{b}) = \boldsymbol{a} \centerdot (\mathsf{L}\boldsymbol{b}) = (\boldsymbol{a} \otimes \boldsymbol{b}) : \mathsf{L}. \tag{A.18c}$$

Let T be a symmetric second-order tensor and S be an arbitrary second-order tensor, then their inner product satisfies the following identity

$$\mathsf{T} : \mathsf{S} = \mathsf{T} : \frac{1}{2}(\mathsf{S} + \mathsf{S}^t). \tag{A.19}$$

This results from the symmetry of T and property of the trace operator (A.9).

A.2 Eigenvalues and eigenvectors of a tensor

Let T be a second-order tensor. Let λ be a scalar and $\boldsymbol{U}$ a vector such that $\mathsf{T}\boldsymbol{U} = \lambda\boldsymbol{U}$. Then λ is called an eigenvalue and $\boldsymbol{U}$ and eigenvector. Note that the eigenvalues and the eigenvectors are independent of the choice of the basis, that is they are invariant under change of basis. Eigenvalues are computed by finding the roots of the characteristic polynomial equation $P_{\mathsf{T}}(\lambda) = 0$, where $P_{\mathsf{T}}(\lambda) = \det(\mathsf{T} - \lambda\mathsf{I}_d)$. A second-order tensor can have at most d eigenvectors and eigenvalues since the space is d-dimensional. We note that if a tensor is positive definite, then its eigenvalues are positive, since by definition $\mathsf{T}\boldsymbol{U}\cdot\boldsymbol{U} = \lambda\|\boldsymbol{U}\|^2 > 0$. If a tensor T admits d real eigenvalues, $\lambda_i,\ i = 1\dots d$, then its trace and its determinant can be expressed in terms of these eigenvalues as

$$\operatorname{tr}(\mathsf{T}) = \sum_{i=1}^{d} \lambda_i, \quad \det \mathsf{T} = \Pi_{i=1}^{d}\lambda_i. \tag{A.20}$$

Therefore, the trace and the determinant are scalar invariants of a tensor.
We conclude this section by recalling the fundamental property satisfies by symmetric tensors. If T is symmetric, it admits d real eigenvalues, $\lambda_i,\ i = 1\dots d$, whose corresponding eigenvectors are mutually orthogonal. Therefore, the normalized eigenvectors, $\boldsymbol{U}_i,\ i = 1\dots d$, form the orthonormal basis of the spectral decomposition which writes as

$$\mathsf{T} = \sum_{i=1}^{d} \lambda_i \boldsymbol{U}_i \otimes \boldsymbol{U}_i. \tag{A.21}$$

This expression is the spectral decomposition of the symmetric tensor T.

A.3 Tensor analysis

In this section, we recall the definition of the divergence and the gradient of a vector and a second-order tensor. It is assumed that the functions employed here are sufficiently smooth so that the limits and derivatives taken are always defined.

A.3.1 Derivative of tensor with respect to a scalar

Let $\mathsf{L} = \mathsf{L}(t)$ be a tensor function of scalar t. The derivative of L with respect to t is the second-order tensor defined by

$$\frac{d}{dt}\mathsf{L} = \lim_{h\to 0} \frac{\mathsf{L}(t+h) - \mathsf{L}(t)}{h}. \tag{A.22}$$

Using this definition, the derivative of the dyadic product of the two vectors $\boldsymbol{a} = \boldsymbol{a}(t)$ and $\boldsymbol{b} = \boldsymbol{b}(t)$ is given by

$$\frac{d}{dt}(\boldsymbol{a}\otimes\boldsymbol{b}) = \frac{d\boldsymbol{a}}{dt}\otimes\boldsymbol{b} + \boldsymbol{a}\otimes\frac{d\boldsymbol{b}}{dt}. \tag{A.23}$$

A.3.2 Derivative of scalar valued functions of vectors

Let $f = f(\boldsymbol{x})$ be a real valued function of the vector $\boldsymbol{x}$. Then, the derivative of $f(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ in the direction $\boldsymbol{a}$ is the vector $\boldsymbol{\nabla} f$ defined as

$$\boldsymbol{\nabla} f \cdot \boldsymbol{a} = \lim_{h\to 0} \frac{f(\boldsymbol{x}+h\boldsymbol{a}) - f(\boldsymbol{x})}{h}. \tag{A.24}$$

The vector $\boldsymbol{\nabla} f$ is the gradient of f, which writes with respect to the orthonormal basis

$$\boldsymbol{\nabla} f = \sum_{i=1}^{d} \frac{\partial f}{\partial x_i} \boldsymbol{e}_i.$$

A.3.3 Derivative of vector valued functions of vectors

Let $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{x})$ be a vector valued function of the vector $\boldsymbol{x}$. Then, the derivative of $\boldsymbol{f}(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ in the direction $\boldsymbol{a}$ is the second-order tensor $\nabla \boldsymbol{f}$ defined as

$$(\nabla \boldsymbol{f})\boldsymbol{a} = \lim_{h \to 0} \frac{\boldsymbol{f}(\boldsymbol{x} + h\boldsymbol{a}) - \boldsymbol{f}(\boldsymbol{x})}{h}. \tag{A.25}$$

On the orthonormal basis, the gradient tensor $\nabla \boldsymbol{f}$ read as

$$\nabla \boldsymbol{f} = \sum_{i=1}^{d} \sum_{j=1}^{d} \frac{\partial f_i}{\partial x_j} (\boldsymbol{e}_i \otimes \boldsymbol{e}_j).$$

A.3.4 Derivative of scalar valued functions of second-order tensors

Let $f = f(\mathsf{S})$ be a scalar valued function of the second-order tensor S. Then, the derivative of $f(\mathsf{S})$ with respect to S in the direction of the second-order tensor T is the second-order tensor $\frac{\partial f}{\partial \mathsf{S}}$ defined as

$$\frac{\partial f}{\partial \mathsf{S}} : \mathsf{T} = \lim_{h \to 0} \frac{f(\mathsf{S} + h\mathsf{T}) - f(\mathsf{S})}{h}. \tag{A.26}$$

On the orthonormal basis, this second-order tensor writes as

$$\frac{\partial f}{\partial \mathsf{S}} = \sum_{i=1}^{d} \sum_{j=1}^{d} \frac{\partial f}{\partial S_{ij}} (\boldsymbol{e}_i \otimes \boldsymbol{e}_j).$$

Let us consider the composed scalar valued function $f = f(\mathsf{S}(t))$, then its derivative with respect to the scalar variable t is given by

$$\frac{df}{dt} = \frac{\partial f}{\partial \mathsf{S}} : \frac{d\mathsf{S}}{dt}. \tag{A.27}$$

A.3.5 Derivative of the determinant of a tensor

The derivative of the determinant of a second-order tensor T with respect to itself is the second-order tensor

$$\frac{\partial}{\partial \mathsf{T}} (\det \mathsf{T}) = \mathsf{T}^{-t} \det \mathsf{T}, \tag{A.28}$$

where T^{-t} denotes the transpose of the inverse of T. The proof of this result consists in computing the partial derivative of the determinant with respect to its component through the use of (A.10)

$$\frac{\partial}{\partial T_{ij}} (\det \mathsf{T}) = \sum_{i=1}^{d} (-1)^{i+j} M_{ij}.$$

Then, comparing this result with the definition of the inverse (A.12) it is clear that

$$\frac{\partial}{\partial T_{ij}} (\det \mathsf{T}) = T_{ji}^{-1} \det \mathsf{T},$$

which ends the proof.

A.3.6 Divergence of a vector and a tensor

Let $\boldsymbol{U} = \boldsymbol{U}(\boldsymbol{x})$ be a vector function of $\boldsymbol{x}$, its divergence is a scalar denoted by $\nabla \boldsymbol{.} \boldsymbol{U}$ and defined as

$$\nabla \boldsymbol{.} \boldsymbol{U} = \sum_{i=1}^{d} \frac{\partial U_i}{\partial x_i}. \tag{A.29}$$

Let us remark that the divergence of a vector can also be defined as

$$\nabla \boldsymbol{.} \boldsymbol{U} = \mathrm{tr}(\nabla \boldsymbol{U}). \tag{A.30}$$

This alternative definition is more intrinsic since it does not require the use of the components of $\boldsymbol{U}$ provided that the tensor gradient of $\boldsymbol{U}$ is well defined.
Let T be a second-order tensor, its divergence is a vector denoted by $\nabla \boldsymbol{.} \mathsf{T}$ whose components are expressed in terms of the tensor components as follows

$$(\nabla \boldsymbol{.} \mathsf{T})_i = \sum_{j} \frac{\partial T_{ij}}{\partial x_j}. \tag{A.31}$$

A.3.7 Tensor identities

Having in mind the previous notation and definitions it is straightforward to demonstrate the following important identities.

$$\nabla \boldsymbol{.} (\mathsf{T}\boldsymbol{U}) = \nabla \boldsymbol{.} (\mathsf{T}^t) \boldsymbol{.} \boldsymbol{U} + \mathsf{T}^t : \nabla \boldsymbol{U}, \tag{A.32a}$$
$$\nabla \boldsymbol{.} (\boldsymbol{U} \otimes \boldsymbol{V}) = (\nabla \boldsymbol{U})\boldsymbol{V} + \boldsymbol{U} \nabla \boldsymbol{.} \boldsymbol{V}, \tag{A.32b}$$
$$\nabla(\rho \boldsymbol{U}) = \rho \nabla \boldsymbol{U} + \boldsymbol{U} \otimes \boldsymbol{\nabla} \rho, \tag{A.32c}$$
$$\nabla \boldsymbol{.} (\rho \mathsf{T}) = \rho \nabla \boldsymbol{.} \mathsf{T} + \mathsf{T} \boldsymbol{\nabla} \rho, \tag{A.32d}$$

where ρ, $\boldsymbol{U}$, $\boldsymbol{V}$ and T are respectively arbitrary scalar, vectors and second-order tensor.

Comment 5 *Let us remark that tensor identity* (A.32a) *allows to propose the following intrinsic definition of the divergence of a tensor. Let* T *be a second-order tensor and* $\boldsymbol{U}$ *a constant arbitrary vector. The divergence of* T *is the vector denoted* $\nabla \boldsymbol{.} \mathsf{T}$ *such as*

$$(\nabla \boldsymbol{.} \mathsf{T}) \boldsymbol{.} \boldsymbol{U} = \nabla \boldsymbol{.} (\mathsf{T}^t \boldsymbol{U}), \quad \forall \boldsymbol{U}. \tag{A.33}$$

A.4 Integral transformation formulas

The Green formula, otherwise named divergence formula, states that the volume integral of the divergence of a function is equal to the total flux of this function through the surface enclosing the volume. We recall its formulations for vectors and second-order tensors. All the real, vector and tensor valued functions are assumed to be continuously differentiable with respect to the spatial variables. Let ω be a domain of the d-dimensional space enclosed by a surface $\partial\omega$, then for arbitrary second-order tensor T and vector $\boldsymbol{U}$

$$\int_{\omega} \nabla \boldsymbol{.} \mathsf{T} \, \mathrm{d}v = \int_{\partial\omega} \mathsf{T}\boldsymbol{n} \, \mathrm{d}s, \tag{A.34a}$$
$$\int_{\omega} \nabla \boldsymbol{.} \boldsymbol{U} \, \mathrm{d}v = \int_{\partial\omega} \boldsymbol{U} \boldsymbol{.} \boldsymbol{n} \, \mathrm{d}s, \tag{A.34b}$$

where $\boldsymbol{n}$ denotes the unit outward normal to the enclose surface $\partial\omega$. Let f denotes a scalar valued function, then using the previous results it is straightforward to demonstrate the following useful formulas

$$\int_{\omega} \nabla \boldsymbol{U} \, \mathrm{d}v = \int_{\partial\omega} \boldsymbol{U} \otimes \boldsymbol{n} \, \mathrm{d}s, \tag{A.35a}$$

$$\int_{\omega} \boldsymbol{\nabla} f \, \mathrm{d}v = \int_{\partial\omega} f \boldsymbol{n} \, \mathrm{d}s. \tag{A.35b}$$

We conclude this section by extending the divergence theorem to the case of piecewise continuously differentiable functions. To this end, we assume that ω is divided into two non-overlapping volumes ω_1 and ω_2 over which $\boldsymbol{U}$ and $\boldsymbol{T}$ are continuously differentiable. Let σ be the intersection surface of ω_1 and ω_2, i.e. $\sigma = \omega_1 \cap \omega_2$, refer to Figure A.1. This is the surface of discontinuity for functions $\boldsymbol{U}$ and T. Let $\boldsymbol{x}_\sigma$ be the position vector of a point located on this surface and $\boldsymbol{n}_\sigma$ the unit normal to σ located at $\boldsymbol{x}_\sigma$ and pointing in the direction of ω_2.

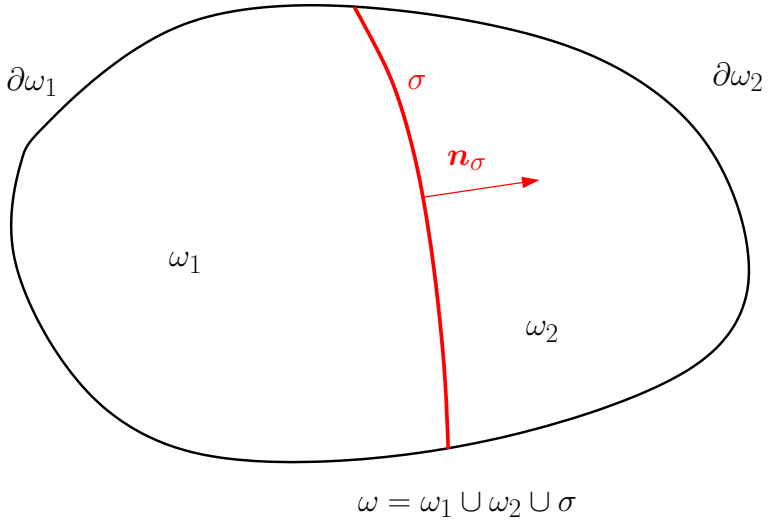


Figure A.1: Volume ω is split into two sub-volumes ω_1 and ω_2 by the discontinuity surface σ.

The jumps of the functions $\boldsymbol{U}$ and T through the discontinuity surface σ are defined as

$$\begin{aligned} [\![\boldsymbol{U}]\!] &= \boldsymbol{U}_2 - \boldsymbol{U}_1, \\ [\![\mathsf{T}]\!] &= \mathsf{T}_2 - \mathsf{T}_1, \end{aligned}$$

where for an arbitrary $\boldsymbol{x}_\sigma \in \sigma$

$$\begin{aligned} \boldsymbol{U}_1 &= \lim_{h \to 0^+} \boldsymbol{U}(\boldsymbol{x}_\sigma - h\boldsymbol{n}_\sigma), \quad \boldsymbol{U}_2 = \lim_{h \to 0^+} \boldsymbol{U}(\boldsymbol{x}_\sigma + h\boldsymbol{n}_\sigma), \\ \mathsf{T}_1 &= \lim_{h \to 0^+} \mathsf{T}(\boldsymbol{x}_\sigma - h\boldsymbol{n}_\sigma), \quad \mathsf{T}_2 = \lim_{h \to 0^+} \mathsf{T}(\boldsymbol{x}_\sigma + h\boldsymbol{n}_\sigma). \end{aligned}$$

$\boldsymbol{U}$ being continuously differentiable separately over ω_1 and ω_2, application of divergence formula (A.34b) yields

$$\begin{aligned} \int_{\omega_1} \nabla \boldsymbol{\cdot} \boldsymbol{U} \, \mathrm{d}v &= \int_{\partial\omega_1 \setminus \sigma} \boldsymbol{U} \boldsymbol{\cdot} \boldsymbol{n} \, \mathrm{d}s + \int_{\sigma} \boldsymbol{U}_1 \boldsymbol{\cdot} \boldsymbol{n}_\sigma \, \mathrm{d}s, \\ \int_{\omega_2} \nabla \boldsymbol{\cdot} \boldsymbol{U} \, \mathrm{d}v &= \int_{\partial\omega_2 \setminus \sigma} \boldsymbol{U} \boldsymbol{\cdot} \boldsymbol{n} \, \mathrm{d}s - \int_{\sigma} \boldsymbol{U}_2 \boldsymbol{\cdot} \boldsymbol{n}_\sigma \, \mathrm{d}s. \end{aligned}$$

The sum of the two previous equations leads to

$$\int_{\omega_1 \cup \omega_2} \nabla \cdot \boldsymbol{U} \, \mathrm{d}v + \int_{\sigma} [\![\boldsymbol{U}]\!] \cdot \boldsymbol{n}_{\sigma} \, \mathrm{d}s = \int_{\partial\omega} \boldsymbol{U} \cdot \boldsymbol{n} \, \mathrm{d}s. \tag{A.36}$$

This equation consists of a generalization of divergence formula (A.34b) to piecewise continuously differentiable vector function. The second term in the left-hand side corresponds to the flux through σ resulting from the jump across this surface. We notice that this term cancels when $[\![\boldsymbol{U}]\!] \to 0$. Proceeding with the tensor valued function as before yields

$$\int_{\omega_1 \cup \omega_2} \nabla \cdot \mathsf{T} \, \mathrm{d}v + \int_{\sigma} [\![\mathsf{T}]\!] \boldsymbol{n}_{\sigma} \, \mathrm{d}s = \int_{\partial\omega} \mathsf{T} \boldsymbol{n} \, \mathrm{d}s. \tag{A.37}$$

Applying the same reasoning formulas (A.35a) and (A.35b) transform into

$$\int_{\omega_1 \cup \omega_2} \nabla \boldsymbol{U} \, \mathrm{d}v + \int_{\sigma} [\![\boldsymbol{U}]\!] \otimes \boldsymbol{n}_{\sigma} \, \mathrm{d}s = \int_{\partial\omega} \boldsymbol{U} \otimes \boldsymbol{n} \, \mathrm{d}s, \tag{A.38a}$$

$$\int_{\omega_1 \cup \omega_2} \boldsymbol{\nabla} f \, \mathrm{d}v + \int_{\sigma} [\![f]\!] \boldsymbol{n}_{\sigma} \, \mathrm{d}s = \int_{\partial\omega} f \boldsymbol{n} \, \mathrm{d}s. \tag{A.38b}$$

Bibliography

[1] R. Abgrall, R. Loubère, and J. Ovadia. A Lagrangian Discontinuous Galerkin-type method on unstructured meshes to solve hydrodynamics problems. *Int. J. Numer. Meth. Fluids*, 44:645–663, 2004.

[2] F. L. Adessio, J R Baumgardner, J. K. Dukowicz, N. L. Johnson, B. A. Kashiwa, R. M. Rauenzahn, and C. Zemach. CAVEAT: a computer code for fluid dynamics problems with large distortion and internal slip. Technical Report LA-10613-MS, Rev. 1, UC-905, Los Alamos National Laboratory, 1992.

[3] A.J. Barlow. A compatible finite element multi-material ALE hydrodynamics algorithm. *Int. J. Numer. Meth. Fluids*, 56:953–964, 2008.

[4] A.J. Barlow and P.L. Roe. A cell centred Lagrangian Godunov scheme for shock hydrodynamics. *Computers and Fluids*, 46(1):133–136, 2011.

[5] T. J. Barth and D. C. Jespersen. The design and application of upwind schemes on unstructured meshes. In *AIAA paper 89-0366*, 27th Aerospace Sciences Meeting, Reno, Nevada, 1989.

[6] M. Ben-Artzi and J. Falcovitz. *Generalized Riemann problems in Computational Fluids Dynamics*. Cambridge University press, 2003.

[7] D. J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Comput. Methods Appl. Mech. and Engrg.*, 99:235–394, 1992.

[8] R. Biswas, K. Devine, and J.E. Flaherty. Parallel adaptive finite element methods for conservation laws. *Applied Numerical Mathematics*, 14:255–284, 1994.

[9] J. Botsis and M. Deville. *Mécanique des milieux continus*. Presses Polytechniques et Universitaires Romandes, 2006.

[10] B. Boutin, E. Deriaz, P. Hoch, and P. Navaro. Extension of ALE methodology to unstructured conical meshes. *ESAIM: Proceedings*, 7:1–10, 2011.

[11] A. F. Bower. *Applied Mechanics of Solids*. CRC Press, 2010.

[12] J. Breil, L. Hallo, P.-H. Maire, and M. Olazabal-Loumé. Hydrodynamic instabilities in axisymmetric geometry self-similar models and numerical simulations. *Laser and Particle Beams*, 23:155–160, 2005.

[13] A. Burbeau-Augoula. A Node-Centered Artificial Viscosity Method for Two-Dimensional Lagrangian Hydrodynamics Calculations on a Staggered Grid. *Commun. Comput. Phys.*, 8:877–900, 2009.

[14] D.E. Burton. Multidimensional Discretization of Conservation Laws for Unstructured Polyhedral Grids. Technical Report UCRL-JC-118306, Lawrence Livermore National Laboratory, 1994.

[15] J.C. Campbell and M.J. Shashkov. A tensor artificial viscosity using a mimetic finite difference algorithm. *J. Comp. Phys.*, 172:739–765, 2001.

[16] J.C. Campbell and M.J. Shashkov. A compatible Lagrangian hydrodynamics algorithm for unstructured grids. *Selçuk J. Appl. Math.*, 4:53–70, 2003.

[17] E. J. Caramana, M. J. Shashkov, and P. P. Whalen. Formulations of artificial viscosity for multidimensional shock wave computations. *J. Comp. Phys.*, 144:70–97, 1998.

[18] E.J. Caramana, D.E. Burton, M.J. Shashkov, and P.P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *J. Comput. Phys.*, 146:227–262, 1998.

[19] E.J. Caramana and M.J. Shashkov. Elimination of artificial grid distorsion and hourglass–type motions by means of Lagrangian subzonal masses and pressures. *J. Comp. Phys.*, 142:521–561, 1998.

[20] G. Carré, S. Delpino, B. Després, and E. Labourasse. A cell-centered Lagrangian hydrodynamics scheme in arbitrary dimension. *J. Comp. Phys.*, 228:5160–5183, 2009.

[21] J. Cheng and C.-W. Shu. A high order eno conservative lagrangian type scheme for the compressible euler equations. *J. Comp. Phys.*, 227(2):1567–1596, 2007.

[22] J. Cheng and C.-W. Shu. A third-order conservative lagrangian type scheme on curvilinear meshes for the compressible euler equations. *Commun. Comput. Phys.*, 4:1008–1024, 2008.

[23] J. Cheng and C.-W. Shu. A cell-centered lagrangian scheme with the preservation of symmetry and conservation properties for compressible fluid flows in two-dimensional cylindrical geometry. *J. Comp. Phys.*, 229(19):7191–7206, 2010.

[24] B. Cockburn. Discontinuous Galerkin methods. School of Mathematics, University of Minnesota, 2003. Available at `http://compmath.files.wordpress.com/2008/10/discontinuous_galerkin_methods_cockburn.pdf`.

[25] B. Cockburn, S. Hou, and C.-W. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Math. Comp.*, 54:545–581, 1990.

[26] B. Cockburn, S.-Y. Lin, and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. *J. Comp. Phys.*, 84:90–113, 1989.

[27] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Math. Comp.*, 52:411–435, 1989.

[28] B. Cockburn and C.-W. Shu. The Runge-Kutta local projection P1-discontinuous Galerkin finite element method for conservation laws. *RAIRO Modél. Math. Anal. Numér.*, 25:337–361, 1991.

[29] B. Cockburn and C.-W. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *J. Comp. Phys.*, 141:199–224, 1998.

[30] P. Collela and P. Woodward. The piecewise-parabolic method (PPM) for gas dynamics simulations. *J. Comp. Phys.*, 54:174–201, 1984.

[31] B. Després. *Lois de Conservation Euleriennes, Lagrangiennes et méthodes numériques.* Mathématiques et Applications. Springer, 2010.

[32] B. Després. Weak consistency of the cell-centered Lagrangian GLACE scheme on general meshes in any dimension. *Comput. Methods Appl. Mech. Eng.*, 199:2669–2679, 2010.

[33] B. Després and C. Mazeran. Lagrangian Gas Dynamics in Two Dimensions and Lagrangian systems. *Arch. Rational Mech. Anal.*, 178:327–372, 2005.

[34] V.A. Dobrev, T.E. Ellis, Tz. V. Kolev, and R. N. Rieben. Curvilinear finite elements for Lagrangian hydrodynamics . *Int. J. Numer. Meth. Fluids*, 65(11-12):1295–1310, 2011.

[35] V.A. Dobrev, T.E. Ellis, Tz. V. Kolev, and R. N. Rieben. High Order Curvilinear Finite Elements for Lagrangian Hydrodynamics. *submitted to SIAM Journal of Scientific Computing*, 2011. Reference LLNL-JRNL-516394.

[36] J. K. Dukowicz and B. Meltz. Vorticity errors in multidimensional Lagrangian codes. *J. Comp. Phys.*, 99:115–134, 1992.

[37] A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements*, volume 159. Springer, 2004.

[38] S. Galera, P.-H. Maire, and J. Breil. A two-dimensional unstructured cell-centered multimaterial ALE scheme using VOF interface reconstruction. *J. Comp. Phys.*, 229:5755–5787, 2010.

[39] P.-L. George, H. Borouchaki, and P. Laug. Construction de maillage de degré 2 - Partie 1 :triangle P2. Technical Report 7519, INRIA, 2011.

[40] P. Germain. *Mécanique*, volume I. Ellipses, 1986.

[41] W.B. Goad. WAT: A Numerical Method for Two-Dimensional Unsteady Fluid Flow. Technical Report LAMS 2365, Los Alamos National Laboratory, 1960.

[42] E. Godlewski and P.-A. Raviart. *Hyperbolic Systems of Conservation Laws.* Ellipses, 1991.

[43] E. Godlewski and P.-A. Raviart. *Hyperbolic Systems of Conservation Laws.* Springer Verlag, 2000.

[44] S.K. Godunov, A. Zabrodine, M. Ivanov, A. Kraiko, and G. Prokopov. *Résolution numérique des problèmes multidimensionnels de la dynamique des gaz.* Editions Mir, 1979.

[45] P. Gresho. On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via finite-element method that also introduces a nearly consistent mass matrix. Part 2: Applications. *Int. J. Numer. Meth. Fluids*, 1990.

[46] J. L. Guermond and R. Pasquetti. Entropy viscosity method for high-order approximations of conservation laws. *Lecture Notes in computational Science and Engineering*, 2009.

[47] J.-L. Guermond, R. Pasquetti, and B. Popov. Entropy viscosity method for non-linear conservation laws. *J. Comp. Phys.*, 230:4248–4267, 2011.

[48] M.E. Gurtin, E. Fried, and L. Anand. *The mechanics and thermodynamics of continua*. Cambridge University Press, 2009.

[49] S. Hou and X.-D. Liu. Solutions of Multi-dimensional Hyperbolic Systems of Conservation Laws by Square Entropy Condition Satisfying Discontinuous Galerkin Method. *J. Sci. Comput.*, 31:127–151, 2007.

[50] W. H. Hui, P. Y. Li, and Z. W. Li. A Unified Coordinate System for Solving the Two-Dimensional Euler Equations. *J. Comp. Phys.*, 153:596–637, 1999.

[51] Z. Jia and S. Zhang. A new high-order discontinuous galerkin spectral finite element for lagrangian gas dynamics in two-dimensions. *J. Comp. Phys.*, 230(7):2496–2522, 2011.

[52] G.-S. Jiang and C.-W. Shu. On cell entropy inequality for discontinuous galerkin method for a scalar hyperbolic equation. *Mathematics of Computation*, 62:531–538, 1994.

[53] J.R. Kamm and F.X. Timmes. On efficient generation of numerically robust Sedov solutions. Technical Report LA-UR-07-2849, Los Alamos National Laboratory, 2007.

[54] R. E. Kidder. Laser-driven compression of hollow shells : power requirements and stability limitations. *Nucl. Fusion*, 1:3–14, 1976.

[55] G. Kluth. *Analyse mathématique et numérique de systèmes hyperélastiques et introduction de la plasticité*. PhD thesis, Université Pierre et Marie Curie (Paris), 2008.

[56] G. Kluth and B. Després. Discretization of hyperelasticity on unstructured mesh with a cell-centered Lagrangian scheme. *J. Comp. Phys.*, 229(24):9092–9118, 2010.

[57] Tz.V. Kolev and R.N. Rieben. A tensor artificial viscosity using a finite element approach. *J. Comp. Phys.*, 228:8336–8366, 2010.

[58] V. P. Kolgan. Application of the principle of minimizing the derivative to the construction of finite-difference schemes for computing discontinuous solution of gas dynamics. *J. Comp. Phys.*, 230:2384–2390, 2011.

[59] A. Kurganov, G. Petrova, and B. Popov. Adaptive semi-discrete central-upwind schemes for non convex hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 29:2381–2401, 2007.

[60] D. Kuzmin. A vertex-based hierarchical slope limiter for p-adaptative discontinuous Galerkin methods. *J. Comp. Appl. Math.*, 233:3077–3085, 2009.

[61] P. Lascaux. Application de la méthode des éléments finis en hydrodynamique bi-dimensionnelle utilisant les variables de Lagrange. Technical Report DO 058, CEA-Centre d'Etudes de Limeil, 1972.

[62] P. Lascaux. Application of the Finite Element Method to 2D Lagrangian hydrodynamics. In *Finite element methods in flow problems*, pages 139–152. Proceedings of the International Symposium, Swansea, Wales; 7-11 Jan. 1974., 1974.

[63] P. Lax and B. Wendroff. Systems of conservation laws. *Commun. Pur. Appl. Math.*, 13:217–237, 1960.

[64] J. Lemaitre, J.-L. Chaboche, A. Benallal, and R. Desmorat. *Mécanique des matériaux solides*. Dunod, 2009.

[65] R. J. LeVeque. *Numerical Methods for Conservations Laws*. Lectures in Mathematics ETH Zürich. Birkhaüser, 1992.

[66] R. J. LeVeque. High-resolution conservative algorithms for advection in compressible flow. *SIAM J. Numer. Anal.*, 33:627–665, 1996.

[67] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*, volume 31. Cambridge Texts in Applied Mathematics, 2002.

[68] K. Lipnikov and M. Shashkov. A framework for developing a mimetic tensor artificial viscosity for Lagrangian hydrocodes on arbitrary polygonal meshes. *J. Comp. Phys.*, 229:7911–7941, 2010.

[69] W.L. Liu, J. Cheng, and C.-W. Shu. High order conservative lagrangian scheme with lax-wendroff type time discretization for the compressible euler equations. *J. Comp. Phys.*, 228(23):8872–8891, 2009.

[70] R. Loubère. *Une Méthode Particulaire Lagrangienne de type Galerkin Discontinu. Application à la Mécanique des Fluides et l'Interaction Laser/Plasma*. PhD thesis, Université Bordeaux I, 2002.

[71] H. Luo, J.D. Baum, and R. Löhner. Fast p-multigrid discontinuous Galerkin method for compressible flows at all speeds. *AIAA J.*, 46:635–652, 2008.

[72] S. M. Murman M. Berger, M.J. Aftosmis. Analysis of slope limiters on irregular grids. Technical Report NAS-05-007, NAS Technical Report, 2005.

[73] P.-H. Maire. A high-order cell-centered Lagrangian scheme for compressible fluid flows in two-dimensional cylindrical geometry. *J. Comp. Phys.*, 228:6882–6915, 2009.

[74] P.-H. Maire. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *J. Comp. Phys.*, 228:2391–2425, 2009.

[75] P.-H. Maire. A unified sub-cell force-based discretization for cell-centered Lagrangian hydrodynamics on polygonal grids. *Int. J. Numer. Meth. Fluids*, 65(11-12):1281–1294, 2010.

[76] P.-H. Maire. A high-order one-step sub-cell force-based discretization for cell-centered Lagrangian hydrodynamics on polygonal grids . *Computers and Fluids*, 46(1):479–485, 2011.

[77] P.-H. Maire. Contribution to the numerical modeling of Inertial Confinement Fusion, 2011. Habilitation à Diriger des Recherches, Bordeaux University, available at `http://tel.archives-ouvertes.fr/docs/00/58/97/58/PDF/hdr_main.pdf`.

[78] P.-H. Maire, R. Abgrall, J. Breil, and J. Ovadia. A cell-centered Lagrangian scheme for two-dimensional compressible flow problems. *SIAM J. Sci. Comput.*, 29:1781–1824, 2007.

[79] P.-H. Maire and J. Breil. A second-order cell-centered Lagrangian scheme for two-dimensional compressible flow problems. *Int. J. Numer. Meth. Fluids*, 56:1417–1423, 2008.

[80] P.-H. Maire, J. Breil, and S. Galera. A cell-centred arbitrary Lagrangian-Eulerian (ALE) method. *Int. J. Numer. Meth. Fluids*, 56:1161–1166, 2008.

[81] P.-H. Maire, R. Loubère, and P. Vàchal. Staggered Lagrangian discretization based on cell-centered Riemann solver and associated hydrodynamics scheme. *Commun. Comput. Phys.*, 10:940–978, 2011.

[82] P.-H. Maire and B. Nkonga. Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics. *J. Comp. Phys.*, 228:799–821, 2009.

[83] C. Mazeran. *Sur la structure mathématique et l'approximation numérique de l'hydrodynamique Lagrangienne bidimensionelle*. PhD thesis, Université Bordeaux I, 2007.

[84] R. Menikoff. Notes on Elastic-Plastic Flow. Technical Report LA-UR-03-0047, Los Alamos National Laboratory, 2003.

[85] W. F. Noh. Errors for calculations of strong shocks using artificial viscosity and an artificial heat flux. *J. Comp. Phys.*, 72:78–120, 1987.

[86] W.F. Noh. *Methods in Computational Physics*, volume 3, chapter CEL: A Time-dependent Two-Space-Dimensional Coupled Eulerian-Lagrangian Code, pages 117–179. Academic Press, 1964.

[87] S. Del Pino. A curvilinear finite-volume method to solve compressible gas dynamics in semi-Lagrangian coordinates. *Comptes Rendus Mathématique*, 348:1027–1032, 2010.

[88] J. Qiu, B. C. Khoo, and C.-W. Shu. A numerical study of the performance of the Runge-Kutta discontinuous Galerkin method based on different numerical fluxes. *J. Comp. Phys.*, 212:540–565, 2006.

[89] B. Rebourcet. Comments on the filtering of numerical instabilities in Lagrangian hydrocodes. Conference on Numerical methods for multi-material fluid flows; Czech Technical University in Prague on September 10 - 14, 2007. Available at `http://www-troja.fjfi.cvut.cz/~multimat07/presentations/tuesday/Rebourcet_filtering.pdf`.

[90] W. H. Reed and T. R. Hill. Triangular Mesh Methods for the Neutron Transport Equation. Technical Report LA-UR-73-479, Los Alamos National Laboratory, 1973.

[91] J. Salençon. *Mécanique des milieux continus*, volume I, Concepts généraux. Editions de l'Ecole Polytechnique, 2005.

[92] G. Scovazzi. Stabilized shock hydrodynamics: II. Design and physical interpretation of the SUPG operator for Lagrangian computations. *Comput. Methods Appl. Mech. and Engrg.*, 196:966–978, 2007.

[93] G. Scovazzi, M.A. Christon, T.J.R Hughes, and J.N. Shadid. Stabilized shock hydrodynamics: I. A Lagrangian method. *Comput. Methods Appl. Mech. and Engrg.*, 196:923–966, 2007.

[94] G. Scovazzi and T.J. Hugues. Lecture Notes on Continuum Mechanics on Arbitrary Moving Domains. Technical Report 2007-6312P, Sandia National Laboratories, 2007.

[95] G. Scovazzi, E. Love, and M.J. Shashkov. Multi-scale Lagrangian shock hydrodynamics on Q1/P0 finite elements: Theoretical framework and two-dimensional computations. *Comput. Methods Appl. Mech. and Engrg.*, 197:1056–1079, 2008.

[96] J. Serrin. Mathematical Principles of Classical Fluid Mechanics. In *Handbuch der Physik*, volume VIII, pages 125–263. Springer Verlag, 1959.

[97] C.-W. Shu. TVB uniformly high-order schemes for conservations laws. *Math. Comp.*, 49:105–121, 1987.

[98] C.-W. Shu. Discontinuous Galerkin methods: General approach and stability. Advanced Courses in Mathematics CRM Barcelona, 2009. Available at `http://www.dam.brown.edu/scicomp/media/report_files/BrownSC-2008-08.pdf`.

[99] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comp. Phys.*, 77:439–471, 1988.

[100] G. A. Sod. A survey of several finite difference methods for systems of non-linear hyperbolic conservation laws. *J. Comp. Phys.*, 27:1–31, 1978.

[101] B. van Leer. Towards the ultimate conservative difference scheme. V-A second-order sequel to Godunov's method. *J. Comput. Phys.*, 32:101–136, 1979.

[102] B. van Leer. A historical oversight: Vladimir P. Kolgan and his high-resolution scheme. *J. Comp. Phys.*, 230:2378–2383, 2011.

[103] F. Vilar. Cell-centered discontinuous galerkin discretization for two-dimensional lagrangian hydrodynamics. *Computers and Fluids*, 64:64–73, 2012.

[104] F. Vilar, P.-H. Maire, and R. Abgrall. Cell-centered discontinuous Galerkin discretizations for two-dimensional scalar conservation laws on unstructured grids and for one-dimensional Lagrangian hydrodynamics. *Computers and Fluids*, 46(1):498–604, 2010.

[105] J. von Neumann and R. D. Richtmyer. A method for the numerical calculations of hydrodynamical shocks. *J. Appl. Phys.*, 21:232–238, 1950.

[106] P. Whalen. Algebraic limitations on two dimensionnal hydrodynamics simulations. *J. Comput. Phys.*, 124:46–54, 1996.

[107] M.L. Wilkins. *Methods in Computational Physics*, volume 3, chapter Calculation of elastic-plastic flows, pages 211–263. Academic Press, 1964.

[108] Y. Xu and C.-W. Shu. Local Discontinuous Galerkin Methods for High-Order Time-Dependent Partial Differential Equations. *Commun. Comput. Phys.*, 7:1–46, 2010.

[109] M. Yang and Z.J. Wang. A parameter-free generalized moment limiter for high-order methods on unstrucured grids. *Adv. Appl. Math. Mech.*, 4:451–480, 2009.

[110] X. Zhang and C.-W. Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *J. Comp. Phys.*, 229:3091–3120, 2010.