Contributions to the high order approximation of compressible flows: multiphase flows, low Mach number flows

Vincent Perrier



Mémoire d'habilitation à diriger des recherches

présenté par

Vincent Perrier

Université de Pau et des Pays de l'Adour
Laboratoire de Mathématiques et de leurs applications de Pau UMR CNRS 5142
Équipe-projet inria Cagire

Contributions to the high order approximation of compressible flows: multiphase flows, low Mach number flows

---

Defended on June, 29th 2021

Defense committee

Rémi ABGRALL	Professor, Universität Zürich	Examiner
Hélène BARUCQ	Research director, INRIA	President
Francesco BASSI	Professor (retired), Università degli studi di Bergamo	Reviewer
Sergey GAVRILYUK	Professor, Aix-Marseille Université	Examiner
Philippe HELLUY	Professor, Université de Strasbourg	Reviewer
Claus-Dieter MUNZ	Professor, Universität Stuttgart	Reviewer

Contents

Introduction

The aim of this document is to summarize the scientific work I have been doing over the last years since my PhD defense in 2007. The topic of my PhD was the modeling and approximation of multiphase compressible flows with or without phase transition [2, 112, 113]. I also had the opportunity to briefly work on stochastic homogenization [10].

I was hired at Inria in 2008 in order to develop high order numerical methods, with a focus on discontinuous Galerkin methods for the approximation of compressible flows. This is a very active research topic and instead of focusing on the development of general high order schemes for hyperbolic systems, I focused on some specific types of flows which are known to combine several difficulties: multiphase flows, following my PhD thesis, and low Mach number flows, which was driven by applications in effusion cooling, one of the target flows of the team Cagire, the Inria team I have been working in. Over the last years, I have also been a main contributor of the AeroSol library, a high order finite element library developed at INRIA since 2010. Based on the development of this library, I was invited at NIA in 2017 for doing research on first order formulation for dissipative systems and for the development of compact WENO limiters for discontinuous Galerkin methods. The organization of this document follows these different research directions

Multiphase flows is dealt with in chapter 1. This chapter deals with diffuse interface models for compressible multiphase flows, a topic I have been interested since my Master thesis, published in [2]. During my PhD, I had also worked on stochastic homogenization[10]. I had remarked in my PhD[111, p.58] the following: when linearizing Kapila's model for mixture of two phase flows, the sound velocity is equal to the harmonic average of the sound velocities of the two fluids. This is exactly what is found when performing a linearization of the Euler system, and then the stochastic homogenization of the linearized system in two heterogeneous media [48]. This convinced me that multiphase flows could be addressed with stochastic models.

Using stochastic modeling for deriving models and numerical schemes for multiphase compressible flows is a a guiding principle of chapter 1. The numerical methods developed in this chapter are extensions of [3], a numerical scheme for multiphase flows derived with probabilistic ideas but without clear stochastic modeling. It is worth to remark that the continuous limit of [3] was left unknown, but was required for the cell integral of the discontinuous Galerkin formulation. In [51], a clear stochastic model was derived, which enables both to define a closure for a diffuse interface model, and to recover the finite volume numerical scheme of [3]. From this, a discontinuous Galerkin

was deduced. Some limiters have been tested with this numerical method, both for ensuring stability for flows with shocks, and for ensuring positivity of the volume fraction [50]. Also, the numerical method was tested with reactive Riemann solvers for modeling permeable fronts [52], as an extension of [113] and [21]. All this work was conducted in collaboration with Erwin Franquet, assistant professor at Pau University. In 2017, a project funded by Région Nouvelle Aquitaine, SEIGLE[1] gathering Inria, Poitiers University (PPrime) and CEA-CESTA was funded. The aim of the project is to study flows involving shock-droplets interactions from a numerical and experimental point of view. At the beginning of this project, the stochastic modeling was improved, allowing to find again some previously proposed closures for diffuse interface models, and allowing also to propose new closures [117]. This work was done in collaboration with Enrique Gutiérrez, whose postdoc was funded by the SEIGLE project.

Low Mach number flows. In 2011, the Inria CAGIRE team-project was created. This project initially aimed at exploring numerically and experimentally a special type of flow that is encountered in effusion cooling: the jets in cross flow. The Mach number of these flows is low, and acoustic waves induced by combustion instabilities may interact with the jets. Last, the simulations are intrinsically unstationary, justifying to resort to high order methods. The first contribution on low Mach number flows was the development of a discontinuous Galerkin method accurate a low Mach number, and able to deal with acoustics. This was done in collaboration with Simon Delmas, who was preparing his PhD under my co-supervision, and Jonathan Jung, assistant professor at Pau University[16]. Then jointly with Jonathan, we have developed a filtering method, which allows to recover accuracy for low Mach number steady flow by doing a simple postprocessing, see [74]. Last, and still in collaboration with Jonathan, a result known for finite volume methods [59], namely the accuracy of some numerical fluxes on triangles and tetrahedra was extended to the discontinuous Galerkin method. The work on low Mach number flows is summarized in chapter 2.

The AEROSOL library. As already stated, the Inria CAGIRE team aimed initially at exploring numerically and experimentally jets in cross flow. We did not have any numerical tool, and this led to the development of AEROSOL, a new high order finite element library on hybrid and possibly curvilinear meshes. This development began in collaboration with the team CARDAMOM (formerly BACCHUS), which was rather interested in continuous finite element approximations. One FP7 European project, IMPACT-AE[2], which I managed for Pau University, was accepted in 2011. This project gathered all major aero-engine manufactures in Europe, seven universities and three research establishments. Pau University was in charge of generating some Direct Numerical Simulations databases of jets in cross flow. This required to address large scale computations. The development of the parallel aspects of AEROSOL was done in collaboration with mainly Damien Genet (PhD student in the BACCHUS team), Benjamin Lux and Maxime Mogé, two young engineers funded by Inria I advised. The architecture, features, and results obtained with the AEROSOL library are summarized in chapter 3. Also, a prospective work on task based programming is summarized in this chapter.

[1] Simulation et Expérimentation de l'Interaction de Gouttes Liquides avec un Écoulement fortement compressible

[2] https://cordis.europa.eu/project/id/265586

Work done during my leave at NIA. In 2017, I was invited to work at National Institute of Aerospace (Hampton, VA), by Alireza Mazaheri (NASA Langley). I was hired there for working on two topics: the use of compact, positivity preserving WENO limiters for stabilizing discontinuous Galerkin methods and the hyperbolic formulation of Navier-Stokes equations. These are published in [93] and submitted in [118], and summarized in chapter 4.

All the chapters have their own introduction and conclusion, and can be read nearly independently. Each chapter ends with some research directions for the topic dealt with in. This document finishes with a conclusion, in which some broader and more transverse possible research directions are mentioned.

Chapter 1

Compressible multiphase flows

1.1 Introduction

When compressible single-phase inviscid flows are considered, the system to solve is the Euler system, except in some extreme cases, for example in rarefied gas dynamics. When it comes to deal with multiphase flows, the problem becomes astonishingly much more complicated. This is due to at least two reasons: first, the dynamic of interfaces is driven by the coupling of the two phases, which may be very stiff, due to strong disparities in the thermodynamic properties of the two phases. Second, multiphase flows quickly becomes a multiscale problem: a flow with well defined interfaces may quickly becomes a flow with very small inclusions of one phase in the other one. These small scales can hardly be neglected: for example, in a liquid water/gaseous air flow, a small droplet of water may still be important for the flow description, because of strong disparities in the densities of the two phases.

A direct consequence of the multiscale nature of multiphase flows is the lack of consensus on which model should be used. Depending on assumptions on the flow, several models have been proposed. For example, if one of the phase is dispersed, the Lagrangian/Eulerian approach can be used [140, 49]. If interfaces are well defined, then numerical methods including *Level-set* methods [106, 45], *Volume-Of-Fluid methods* [58], *Lagrangian* [144] or *Arbitrary-Lagrangian-Eulerian* method [68] can be considered. All these models or numerical methods rely more or less on the coupling of the Euler systems describing each phase. These methods, however, require to have a discretization resolution smaller than the smallest inclusions and interface wavelength. Depending on the methods, conservation may be hard to ensure: this is a well known weakness of level-set methods. Last, some models, are in theory able to blend not only with interface flows, but also with averaged flows. These models are usually reductions or extensions of the Baer-and-Nunziato model [9]. An example of reduction of this model is Kapila's model [76, 98]. Another question that may arise with reduced models is hyperbolicity; when the model is not hyperbolic, it may be approximated by an augmented model with stiff relaxation terms, see e.g. [54]. In these models, the presence of each fluid is described by their volume fraction. We will be interested in this latter type of model, often referred as *diffuse interface models*, see [132] for a review on this type of model.

Up to our knowledge, the most general diphasic model was proposed in [130], and reads

$$\begin{aligned}
&\partial_t \alpha_k + \mathbf{u}_I \cdot \nabla \alpha_k = \mu(P_k - P_{\bar{k}}) \\
&\partial_t\big(\alpha_k \rho_k\big) + \mathrm{div}_{\mathbf{x}}\big(\alpha_k \rho_k \mathbf{u}_k\big) = 0 \\
&\partial_t\big(\alpha_k \rho_k \mathbf{u}_k\big) + \mathrm{div}_{\mathbf{x}}\big(\alpha_k \rho_k \left(\mathbf{u}_k \otimes \mathbf{u}_k + P_k\right)\big) = P_I \nabla \alpha_k + \lambda(\mathbf{u}_{\bar{k}} - \mathbf{u}_k) \\
&\partial_t\big(\alpha_k \rho_k E_k\big) + \mathrm{div}_{\mathbf{x}}\Big(\alpha_k \left(\rho_k E_k + P_k\right) \mathbf{u}_k\Big) = P_I \mathbf{u}_I \cdot \nabla \alpha_k - \mu P_I (P_k - P_{\bar{k}}) + \lambda \mathbf{u}_I \cdot (\mathbf{u}_{\bar{k}} - \mathbf{u}_k)
\end{aligned} \tag{1.1}$$

where k is the index of the phase ($k = 1$ or 2). In (1.1), $\bar{k}$ denotes the conjugate index of k, i.e. $\bar{k} = 2$ if $k = 1$, and $\bar{k} = 1$ if $k = 2$. α_k denotes the volume fraction of the phase k, ρ_k its density, P_k its pressure, $\mathbf{u}_k$ its velocity, and E_k its total energy, which is defined as

$$E_k = \varepsilon_k + \frac{|\mathbf{u}_k|^2}{2},$$

where ε_k is the internal specific energy. It is linked with the other thermodynamic parameters by an equation of state

$$\varepsilon_k = \varepsilon_k(P_k, \rho_k).$$

System (1.1) depends on interfacial terms: the interfacial velocity $\mathbf{u}_I$ and the interfacial pressure P_I, which must be defined. Last, system (1.1) depends on relaxation parameters λ and μ, which shall be defined too.

This whole chapter will be driven by giving a stochastic background in the derivation of (1.1), and by explaining how a numerical scheme can be derived from these ideas. Even if the work I did was first focused on developing a numerical scheme, and then in the development of models, I have chosen to expose it in a more usual order, namely by exposing first the derivation of the model, and then, the associated numerical scheme. In section 1.2, the stochastic modeling and its physical meaning will be explained. Depending on some modeling parameters, several models will then be proposed in section 1.3. Properties of these models will be also discussed. These sections are submitted in [117]. Then in section 1.4, the derivation of the numerical scheme, based on the same ideas, will be exposed. In section 1.5, some of the numerical results will be shown. The numerical scheme and numerical results were published in [50, 51, 52]. The work on numerical scheme was done in collaboration with Erwin Franquet, and the work on modeling was done in collaboration with Enrique Gutiérrez.

1.2 Stochastic modeling of multiphase flows

In this section, we aim at exploring the stochastic modeling of two non miscible media. Even if the two media are not miscible, the modeling scale is supposed to be larger than the possible inclusions of one medium inside the other. As a consequence, on a point $\mathbf{x}$, it may be not certainly known whether the medium k or $\bar{k}$ lies in $\mathbf{x}$. The distribution of the medium k or $\bar{k}$ is modeled by a probabilistic approach. The probability of having the medium k or $\bar{k}$ in $\mathbf{x}$ is characterized by the volume fractions $\alpha_k(\mathbf{x})$ and $\alpha_{\bar{k}}(\mathbf{x})$, with $\alpha_k(\mathbf{x}) + \alpha_{\bar{k}}(\mathbf{x}) = 1$. For one realization, the presence of the fluid k is determined by an indicator function χ_k, where $\chi_k(\mathbf{x}) = 1$ if the fluid k is in $\mathbf{x}$, and $\chi_k(\mathbf{x}) = 0$ otherwise. Of course, we have

$$\chi_k + \chi_{\bar{k}} = 1.$$

Given an averaging operator $\mathbb{E}[\cdot]$, the consistency between the macroscopic model and the stochastic modeling is ensured if

$$\mathbb{E}[\chi_k(\mathbf{x})] = \alpha_k(\mathbf{x}). \tag{1.2}$$

Given a stochastic process $g_{\mathbf{x}}$ with mean $m(\mathbf{x})$ and variance 1, χ_k is supposed to be the image of $g_{\mathbf{x}}$ by the Heaviside function

$$\chi_k(\mathbf{x}) = \frac{1 + \mathrm{sgn}(g_{\mathbf{x}})}{2},$$

in which case we have

$$\chi_{\bar{k}}(\mathbf{x}) = \frac{1 - \mathrm{sgn}(g_{\mathbf{x}})}{2} = \frac{1 + \mathrm{sgn}(-g_{\mathbf{x}})}{2}.$$

1.2.1 Consistency with the volume fraction

We denote by m_k the mean of $g_{\mathbf{x}}$. For consistency with (1.2), we must have

$$\alpha_k(\mathbf{x}) = \int_{-\infty}^{\infty} \frac{1 + \mathrm{sgn}(u)}{2} \, \mathrm{e}^{-(u - m_k(\mathbf{x}))^2/2} \, \mathrm{d}u,$$

which can be rewritten

$$\alpha_k(\mathbf{x}) = \frac{1}{\sqrt{2\pi}} \int_{-m_k(\mathbf{x})}^{\infty} \exp(-u^2/2) \, \mathrm{d}u = \frac{1}{2} \left(1 + \mathrm{erf} \left(\frac{m_k(\mathbf{x})}{\sqrt{2}} \right) \right). \tag{1.3}$$

We thus see that in general the stochastic process $g_{\mathbf{x}}$ is *not stationary*. Also, m_k has the same regularity as α_k, and we have $m_{\bar{k}} = -m_k$.

1.2.2 Measurability of $\nabla \chi_k$

In the derivation of the multiphase model, integrations involving $\nabla \chi_k$ will be needed. For studying the measurability of this process, another parameter of the Gaussian process is introduced: the autocorrelation function. We denote it by R, and it is defined as

$$R \, : \, (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^d \times \mathbb{R}^d \longmapsto \mathbb{E}\left[(g_{\mathbf{x}} - m_k(\mathbf{x}))(g_{\mathbf{y}} - m_k(\mathbf{y})) \right].$$

We then can prove that

Proposition 1 (Prop. 4.2 p. 13 of [117])**.** *$\nabla \chi_k$ is measurable if and only if $\partial_{\mathbf{y}} R(\mathbf{x}, \mathbf{x}) = \partial_{\mathbf{x}} R(\mathbf{x}, \mathbf{x}) = 0$ and $\partial^2_{\mathbf{xy}} R(\mathbf{x}, \mathbf{x})$ is non-negative.*

Note that under hypothesis of Proposition 1, we have $\mathbb{E}\left[\nabla \chi_k(\mathbf{x}) \right] = \nabla \alpha_k(\mathbf{x})$.

1.2.3 Link between the flow topology and the autocorrelation function

Roughly speaking, the autocorrelation function can be understood as follows for our multiphase model: if the correlation of two close points $\mathbf{x}$ and $\mathbf{y}$ is high, then the values on $\mathbf{x}$ and $\mathbf{y}$ are strongly correlated, and so very likely to be in the same phase, so that $[\mathbf{x}, \mathbf{y}]$ is likely to be fully in the same phase. If the correlation of the same $\mathbf{x}$ and $\mathbf{y}$ is low, then it is likely that the two phases in $\mathbf{x}$ and $\mathbf{y}$ are different, so that the inclusions are smaller that the length of $[\mathbf{x}, \mathbf{y}]$. The aim of this section is to quantify the link between the autocorrelation function and the flow topology. The study is performed with a stationary process, in order to separate the effect of the autocorrelation function from the effect of the gradient of the volume fraction.

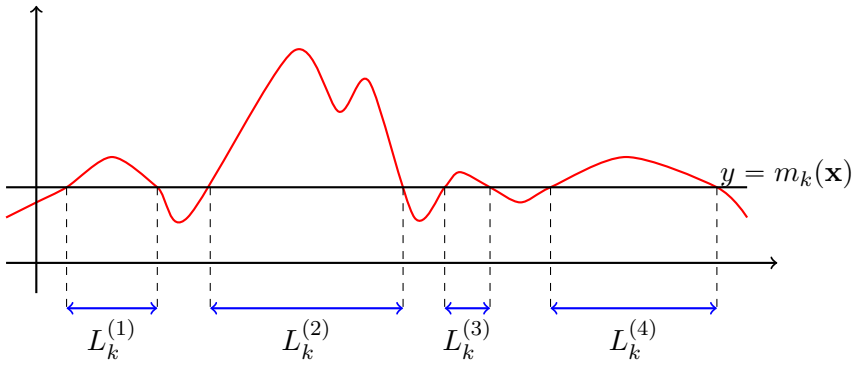


Figure 1.1: Example of one realization of the Gaussian process (in red). The values of x such that the Gaussian process is above $m_k(\mathbf{x})$ belong to the phase k, and the values of x for which $g_{\mathbf{x}} \leqslant m_k(\mathbf{x})$ belong to the phase $\bar{k}$. The double arrows in blue link an upcrossing of the value $m_k(\mathbf{x})$ with the first next downcrossing of $m_k(\mathbf{x})$. The length $L_k^{(i)}$ can be understood as the length of bubbles (or droplets) of the phase k inside the phase $\bar{k}$.

In section 1.3 we will see that an important parameter of the models is the second order derivative of the autocorrelation function $\partial^2_{\mathbf{xy}} R(\mathbf{x}, \mathbf{x})$. Given a normalized direction $\mathbf{n}$, we can consider $(g_{\mathbf{x}+t\mathbf{n}})_{t\in\mathbb{R}}$, which is a one dimensional Gaussian process, for which the auto-correlation function is

$$R_{\mathbf{n}}(t_x, t_y) = R(\mathbf{x} + t_x\mathbf{n}, \mathbf{x} + t_y\mathbf{n}),$$

so that

$$\partial^2_{t_x t_y} R_{\mathbf{n}}(0,0) = \mathbf{n}^T \partial^2_{\mathbf{xy}} R(\mathbf{x}, \mathbf{x})\mathbf{n}. \tag{1.4}$$

Thus this study is performed in one dimension. What we want to simulate is depicted in Figure 1.1. We aim at characterizing how the bubbles/droplets size L_k in one phase k depend on the volume fraction and the autocorrelation function. A numerical simulation is performed with a *squared exponential* auto-correlation function as in [124]:

$$R_{\mathrm{SE}}(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{(\mathbf{y} - \mathbf{x})^2}{2\zeta^2}\right), \tag{1.5}$$

where ζ is a free parameter, set to 3.5. Results shown in Figure 1.2 highlight the typical behavior of the size distribution f_{L_1} when using the model presented above: a single maximum value is obtained for a particular bubble length, with a constant decrease to the right and left of that value. Note therefore that the model is not suitable for flows involving more than one typical bubble size.

We would like to obtain an approximation of the average length of the bubble $\mathbb{E}[L_1]$. Following [4], the number of upcrossings N_u of the value u of a one dimensional stationary Gaussian stochastic process in $[0, 1]$ follows the Rice formula [126]

$$\mathbb{E}[N_u] = \frac{\lambda}{2\pi} \exp\left(-\frac{u^2}{2}\right),$$

where λ is the square root of the second spectral moment, matching with the square root of the second order derivative of the auto-correlation function. In our case, we are interested in

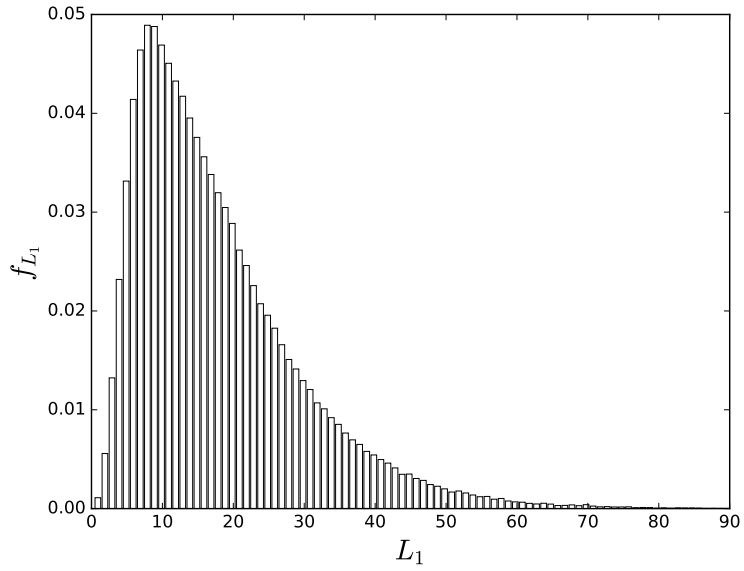


Figure 1.2: Typical bubble size distribution obtained by using the sign of a Gaussian process with a *squared exponential* auto-correlation (1.5) with a parameter ζ equal to 3.5. f_{L_1} is the normalized probability density function, and L_1 is the bubble characteristic length.

the upcrossings with the value $m_k(\mathbf{x})$. Referring to Figure 1.1, the total length of fluid k is

$$\sum_{i=1}^{N_{m_k(\mathbf{x})}} L_k^{(i)},$$

and we expect the mean of this to be $\alpha_k(\mathbf{x})$. If we suppose that the number of upcrossings and the length are independent, then

$$\mathbb{E}\left[L_k\right]\mathbb{E}\left[N_{m_k(\mathbf{x})}\right] = \mathbb{E}\left[\sum_{i=1}^{N_{m_k(\mathbf{x})}} L_k^{(i)}\right] = \alpha_k(\mathbf{x}),$$

which gives the following heuristic formula for $\mathbb{E}\left[L_k\right]$

$$\mathbb{E}\left[L_k\right] \approx \hat{L}_k := \frac{\alpha_k(\mathbf{x})}{\mathbb{E}\left[N_{m(\mathbf{x})}\right]} = \frac{2\pi}{\exp\left(-\dfrac{m_k(\mathbf{x})^2}{2}\right)}\,\alpha_k(\mathbf{x})\,\frac{1}{\lambda}. \tag{1.6}$$

Numerical simulations have been conducted in order evaluate the deviations of $\hat{L}_k$ defined in (1.6) with respect to the experimental values $\mathbb{E}\left[L_k\right]$. By using (1.5), several realizations of the Gaussian stochastic process have been computed, resulting in different sets of $\mathbb{E}\left[L_1\right]$.

Figure 1.3 shows a comparison between the average bubble lengths obtained numerically and those obtained by using (1.6), with respect to the volume fraction α. Several ζ parameters have been considered, ranging from 2.6 to 4.6 with increments of 0.4. Correspondingly,

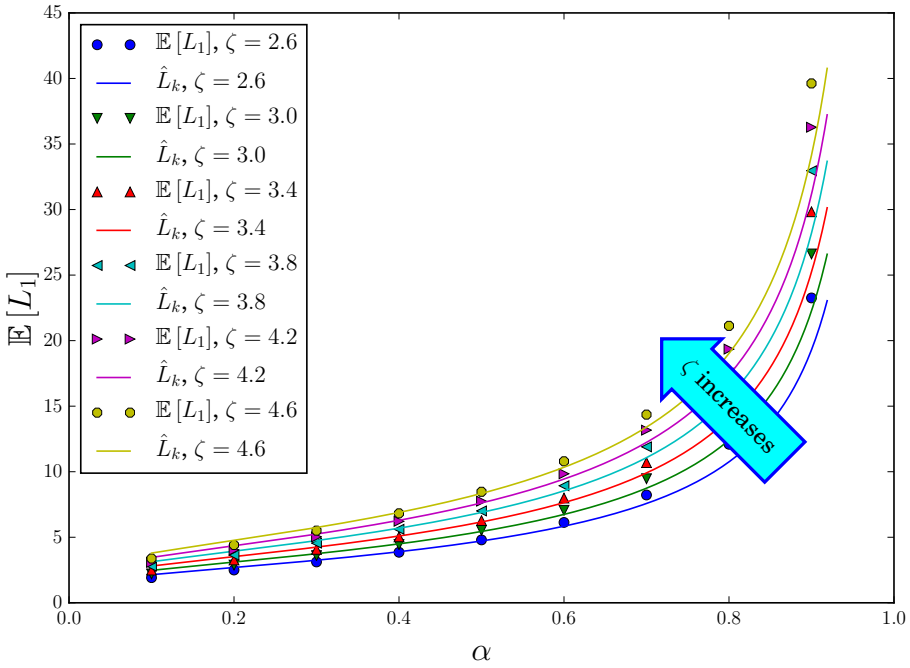


Figure 1.3: Results comparison of the average characteristic size of the bubble $\mathbb{E}[L_1]$ with respect to the volume fraction α. Numerical results are represented by points and those obtained by using (1.6) by lines. Several ζ parameters in (1.5) have been considered, ranging from 2.6 to 4.6 in 0.4 increments.

Figure 1.4 presents the relative error among results, showing the suitability of (1.6) around $\alpha = 0.4$ for the chosen conditions. The deviation of (1.6) with respect to the numerical value is less than 20% for the volume fraction ranging from 0.1 to 0.9.

Dependency of $\mathbb{E}[L_1]$ on λ In any case, it seems that the average bubble size remains proportional to λ^{-1} for any value of α, as suggests (1.6). For example, using the same auto-correlation function as above for a volume fraction ranging from 0.1 to 0.9, the mean bubble sizes $\mathbb{E}[L_1]$ obtained for different λ values have been computed. Figure 1.5 shows the behavior of the computed $\mathbb{E}[L_1]$ with respect to λ, in log scale, showing a well defined slope of -1.

This behavior of $\mathbb{E}[L_1]$ with respect to λ is very interesting in multiple dimensions. Indeed, following (1.4),

$$\lambda_{R,\mathbf{n}} = \sqrt{\mathbf{n}^T \partial^2_{\mathbf{xy}} R(\mathbf{x},\mathbf{x})\mathbf{n}}.$$

As a consequence, in multiple dimensions, even if it is not possible to get the average radius of the bubbles, $\partial^2_{\mathbf{xy}} R(\mathbf{x},\mathbf{x})$ fully defines the anisotropy of the average bubbles: its eigendirections define the directions of anisotropy, whereas the ratio of the square roots of its eigenvalues give the anisotropy intensity.

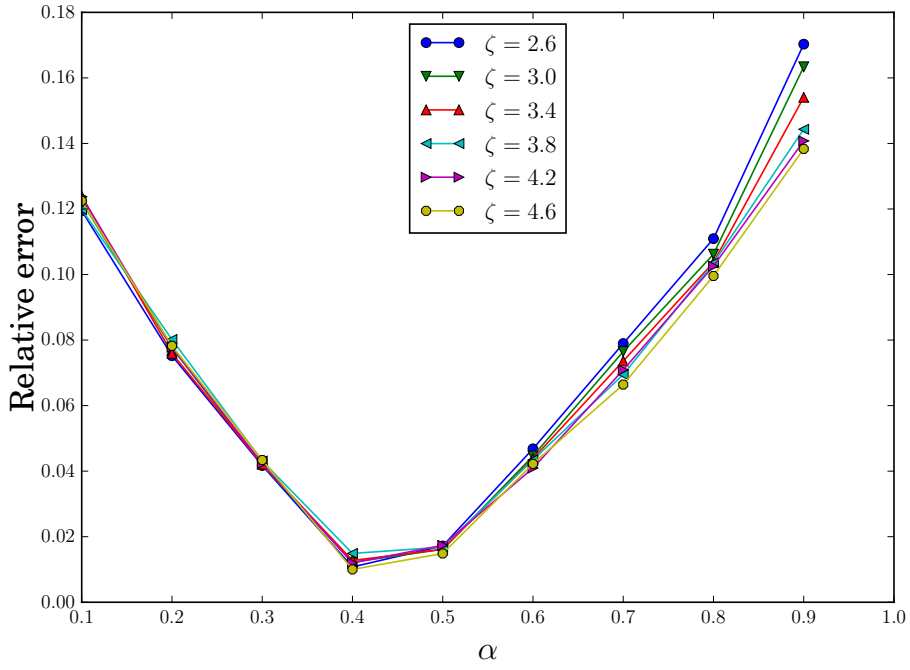


Figure 1.4: Error plot of the results comparison presented in Figure 1.4. The relative error E_1 has been computed as $E_1 = \left|\mathbb{E}[L_1] - \hat{L}_1\right| / \mathbb{E}[L_1]$, where $\mathbb{E}[L_1]$ denotes the average bubble length computed numerically, and $\hat{L}_1$ given by (1.6).

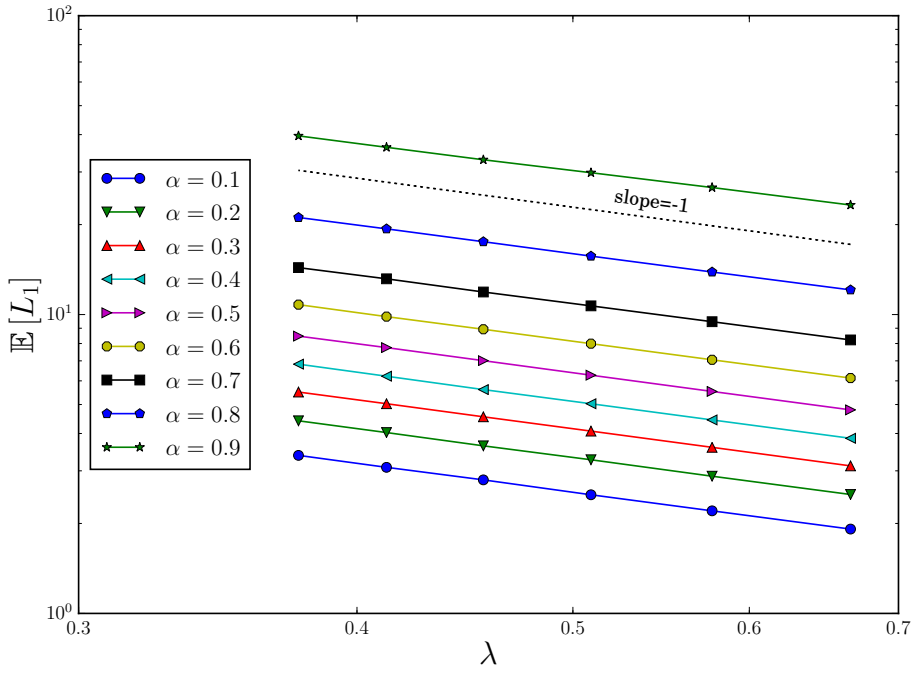


Figure 1.5: Average bubble length $\mathbb{E}[L_1]$ varying with respect to the square of the second order derivative of the auto-correlation function λ, for several values of the volume fraction α. A well-defined slope of -1 can be observed, for all values of α.

1.3 Models obtained for some multiphase topologies

In this section, continuous models for multiphase flows are derived from the stochastic distribution of the phases exposed in the previous section.

1.3.1 Derivation of the deterministic model

For the moment, we suppose that a sufficiently regular *level set* function f exists such that the fluid k is in $\mathbf{x}$, if and only if $f(\mathbf{x}) \geqslant 0$. The indicator function of the fluid k is therefore defined as

$$\chi_k(\mathbf{x}) = \frac{1 + \operatorname{sgn}(f(\mathbf{x}))}{2}.$$

The dynamic of the system is driven by two phenomena: what happens *inside each phase*, and what happens *at interfaces*. Inside each phase, the Euler system of equation, abbreviated as

$$\partial_t \mathbf{U}_k + \operatorname{div}_{\mathbf{x}} \mathbf{F}_k(\mathbf{U}_k) = 0 \tag{1.7}$$

holds. Moreover, The indicator function χ_k ensures the following equation weakly

$$\partial_t \chi_k + \mathbf{v}_i \cdot \nabla \chi_k = 0. \tag{1.8}$$

Indeed, equation (1.8) is clearly ensured when $\chi_k = 0$ or $\chi_k = 1$ because both gradient and time derivative vanish, and holds also on a jump of χ_k in a weak sense provided $\mathbf{v}_i$ is the local interface velocity. Following [37], the following equation holds for each phase in a weak sense

$$\chi_k \left(\partial_t \mathbf{U}_k + \operatorname{div}_{\mathbf{x}} \mathbf{F}_k(\mathbf{U}_k)\right) = 0. \tag{1.9}$$

Indeed, (1.9) holds when the fluid k is not present, because $\chi_k = 0$, but also when the fluid is present, because then the Euler system holds.

Based on equations (1.8) and (1.9), the following system can be formally derived

$$\partial_t \left(\chi_k \mathbf{U}_k\right) + \operatorname{div}_{\mathbf{x}}(\chi_k \mathbf{F}_k(\mathbf{U}_k)) = \left(\mathbf{F}_k(\mathbf{U}_k) - \mathbf{v}_i \mathbf{U}_k\right) \nabla \chi_k. \tag{1.10}$$

Note that by defining $\hat{\mathbf{F}} = (0, \mathbf{F})$ and $\hat{\mathbf{U}} = (1, \mathbf{U})$, equations (1.8) and (1.10) can be rewritten as

$$\partial_t \left(\chi_k \hat{\mathbf{U}}_k\right) + \operatorname{div}_{\mathbf{x}} \left(\chi_k \hat{\mathbf{F}}_k(\hat{\mathbf{U}}_k)\right) = \left(\hat{\mathbf{F}}_k(\hat{\mathbf{U}}_k) - \mathbf{v}_i \hat{\mathbf{U}}_k\right) \nabla \chi_k. \tag{1.11}$$

One idea developed in [3, 51] is that provided the local topology is known and sufficiently regular, $\mathbf{v}_i$ can be computed by solving a one dimensional Riemann problem. Indeed, at one interface (so, one point in which $f(\mathbf{x}) = 0$), and if we suppose that $\nabla f(\mathbf{x})$ does not vanish, the unit normal to the interface can be defined as

$$\mathbf{n} = \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|},$$

and $\mathbf{n}$ is inward with respect to the set $\left\{\mathbf{x} \in \mathbb{R}^d \quad f(\mathbf{x}) \geqslant 0\right\}$. The Riemann problem in the direction $\mathbf{n}$ with phase $\bar{k}$ on the left and phase k on the right can be solved, and denoting by $u^\star_{\bar{k}k}(\mathbf{n})$ and $P^\star_{\bar{k}k}(\mathbf{n})$ the velocity and pressure of the contact discontinuity, the interface velocity is $\mathbf{v}_i = \mathbf{u}^\star_{\bar{k}k}(\mathbf{n}) = u^\star_{\bar{k}k}(\mathbf{n})\mathbf{n}$, and the right hand side of (1.11) can be computed as

$$\left(\hat{\mathbf{F}}_k(\hat{\mathbf{U}}_k) - \mathbf{v}_i \hat{\mathbf{U}}_k\right) \nabla \chi_k = \hat{\mathbf{F}}^{lag}_{\bar{k}k} \left(\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}\right) \|\nabla f(\mathbf{x})\|,$$

where the Lagrangian flux $\hat{\mathbf{F}}^{lag}_{\bar{k}k}(\mathbf{n})$ is defined as

$$\hat{\mathbf{F}}^{lag}_{\bar{k}k}(\mathbf{n}) := \begin{pmatrix} -u^{\star}_{\bar{k}k}(\mathbf{n}) \\ 0 \\ P^{\star}_{\bar{k}k}(\mathbf{n})\mathbf{n} \\ P^{\star}_{\bar{k}k}(\mathbf{n})u^{\star}_{\bar{k}k}(\mathbf{n}) \end{pmatrix}.$$

This gives the final expression for (1.11)

$$\partial_t \left(\chi_k \hat{\mathbf{U}}_k\right) + \mathrm{div}_{\mathbf{x}} \left(\chi_k \hat{\mathbf{F}}_k(\hat{\mathbf{U}}_k)\right) = \hat{\mathbf{F}}^{lag}_{\bar{k}k}\left(\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}\right) \|\nabla f(\mathbf{x})\|\,. \tag{1.12}$$

1.3.2 Averaging

For obtaining the set of equations at the macroscopic scale, it remains to average (1.12). We suppose that f is stochastic, and we consider an averaging operator $\mathbb{E}[\cdot]$. This operator is supposed to commute with the spatial and time derivation (referred as Gauss and Leibniz rules in [37, p.102]). Then

$$\partial_t \left(\mathbb{E}\left[\chi_k \hat{\mathbf{U}}_k\right]\right) + \mathrm{div}_{\mathbf{x}} \left(\mathbb{E}\left[\chi_k \hat{\mathbf{F}}_k(\hat{\mathbf{U}}_k)\right]\right) = \mathbb{E}\left[\hat{\mathbf{F}}^{lag}_{\bar{k}k}\left(\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}\right) \|\nabla f(\mathbf{x})\|\right]. \tag{1.13}$$

Provided $\hat{\mathbf{U}}$ is supposed to not depend on the flow topology, the left hand side of (1.13) is equal to the time derivative and the conservative part of (1.1)

$$\partial_t \left(\alpha_k \hat{\mathbf{U}}\right) + \mathrm{div}_{\mathbf{x}} \left(\alpha_k \hat{\mathbf{F}}_k(\hat{\mathbf{U}}_k)\right) = \mathbb{E}\left[\hat{\mathbf{F}}^{lag}_{\bar{k}k}\left(\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}\right) \|\nabla f(\mathbf{x})\|\right], \tag{1.14}$$

whereas the right hand side

$$\mathbb{E}\left[\hat{\mathbf{F}}^{lag}_{\bar{k}k}\left(\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}\right) \|\nabla f(\mathbf{x})\|\right] \tag{1.15}$$

includes the non-conservative and relaxation terms of (1.13), and cannot be further made explicit without defining an averaging operator. This averaging is defined through the definition of an explicit stochastic model for χ_k as the image of a Gaussian process as described in section 1.2. With this averaging operator, (1.15) appears explicitly as an integral, which, denoting by $\check{\Sigma} = \partial^2_{\mathbf{xy}} R(\mathbf{x},\mathbf{x})^{-1}$ is

$$\begin{aligned} &\mathbb{E}\left[\hat{\mathbf{F}}^{lag}_{\bar{k}k}\left(\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}\right) \|\nabla f(\mathbf{x})\|\right] \\ &= \frac{\exp(-m_k(\mathbf{x})^2/2)}{((2\pi)^{d+1}\det\check{\Sigma})^{1/2}} \times \int_{\mathbf{x}_d\in\mathbb{R}^d} \mathrm{e}^{\frac{(\mathbf{x}_d-\nabla m_k(\mathbf{x}))^T\check{\Sigma}^{-1}(\mathbf{x}_d-\nabla m_k(\mathbf{x}))}{2}} \hat{\mathbf{F}}^{lag}_{\bar{k}k}\left(\frac{\mathbf{x}_d}{\|\mathbf{x}_d\|}\right) \|\mathbf{x}_d\|\,\mathrm{d}\mathbf{x}_d. \end{aligned} \tag{1.16}$$

This integral cannot be made further explicit in general. However, the argument of the exponential in the integral makes clearly appear a competition between two length scales: on one hand, the square root of the eigenvalues of $\check{\Sigma}$, and on the other hand, the invert of $\nabla m_k(\mathbf{x})$.

1.3.3 Models obtained

In this section, (1.16) is made explicit or approximated in the following cases: the one dimensional case, the "low memory" case and the "large memory case". In general, the system is an extension of (1.1) and reads

$$\begin{array}{l}\partial_t \alpha_k + \mathbf{u}_I \cdot \nabla \alpha_k = \mathcal{R}_k^{(\alpha)} \\ \partial_t\big(\alpha_k \rho_k\big) + \mathrm{div}_{\mathbf{x}}\big(\alpha_k \rho_k \mathbf{u}_k\big) = 0 \\ \partial_t\big(\alpha_k \rho_k \mathbf{u}_k\big) + \mathrm{div}_{\mathbf{x}}\big(\alpha_k \rho_k \left(\mathbf{u}_k \otimes \mathbf{u}_k + P_k\right)\big) = \mathbf{P}_I \nabla \alpha_k + \mathcal{R}_k^{(\rho \mathbf{u})} \\ \partial_t\big(\alpha_k \rho_k E_k\big) + \mathrm{div}_{\mathbf{x}}\Big(\alpha_k \left(\rho_k E_k + P_k\right) \mathbf{u}_k\Big) = (P\mathbf{u})_I \cdot \nabla \alpha_k + \mathcal{R}_k^{(\rho E)}\end{array} , \tag{1.17}$$

where $\mathbf{P}_I$ is a tensor, $(P\mathbf{u})_I$ is not directly linked with P_I and $\mathbf{u}_I$, and the $\mathcal{R}_k^\star$ are nonlinear relaxation terms.

1.3.3.1 One dimensional case

In the one dimensional case, the integral of (1.16) can be split into two integrals that can be computed explicitly, for giving the following model.

Proposition 2 (Proposition 5.3 of [117])**.** *In one dimension, setting*

$$w := \mathrm{erf}\left(\frac{|\partial_x m_k(x)|}{\lambda_1 \sqrt{2}}\right)$$

and λ_1 the square root of the single coefficient of $\check{\Sigma}^{-1}$, the system is closed by the following interfacial velocities

$$\left\{\begin{array}{r} \mathbf{u}_I = \mathrm{sgn}(\partial_x \alpha_k(x)) \left(\frac{1+w}{2}\, u_{\bar{k}k}^\star(\partial_x \alpha_k(x)) + \frac{w-1}{2}\, u_{\bar{k}k}^\star(-\partial_x \alpha_k(x))\right) \\ P_I = \mathrm{sgn}(\partial_x \alpha_k(x)) \left(\frac{1+w}{2}\, P_{\bar{k}k}^\star(\partial_x \alpha_k(x)) + \frac{w-1}{2}\, P_{\bar{k}k}^\star(-\partial_x \alpha_k(x))\right) \\ (P\mathbf{u})_I = \mathrm{sgn}(\partial_x \alpha_k(x)) \left(\frac{1+w}{2}\, P_{\bar{k}k}^\star(\partial_x \alpha_k(x)) u_{\bar{k}k}^\star(\partial_x \alpha_k(x)) \right. \\ \left. + \frac{w-1}{2}\, P_{\bar{k}k}^\star(-\partial_x \alpha_k(x)) u_{\bar{k}k}^\star(-\partial_x \alpha_k(x))\right) \end{array}\right.$$

and includes the following nonlinear relaxation term

$$\lambda_1 \mathrm{e}^{-\frac{\partial_x m_k(x)^2}{2\lambda_1^2}} \frac{\mathrm{e}^{-\frac{m_k(x)^2}{2}}}{2\pi} \begin{pmatrix} u_{\bar{k}k}^\star(-\partial_x \alpha(x)) - u_{\bar{k}k}^\star(\partial_x \alpha(x)) \\ 0 \\ P_{\bar{k}k}^\star(\partial_x \alpha(x)) - P_{\bar{k}k}^\star(-\partial_x \alpha(x)) \\ P_{\bar{k}k}^\star(\partial_x \alpha(x)) u_{\bar{k}k}^\star(\partial_x \alpha(x)) - P_{\bar{k}k}^\star(-\partial_x \alpha(x)) u_{\bar{k}k}^\star(-\partial_x \alpha(x)) \end{pmatrix}.$$

1.3.3.2 Long memory case

The long memory case matches with the case in which the auto-correlation function R is flat, more precisely, when $\partial_{\mathbf{xy}}^2 R(\mathbf{x}, \mathbf{x}) = 0$. In this case,

Proposition 3 (Long memory case)**.** *When $\partial_{\mathbf{xy}}^2 R(\mathbf{x}, \mathbf{x}) \to 0$, (1.16) gives the system (1.1) with $\lambda = \mu = 0$, and the following interfacial velocity and pressure*

$$\mathbf{u}_I = u^\star\left(\frac{\nabla \alpha(\mathbf{x})}{\|\nabla \alpha(\mathbf{x})\|}\right) \frac{\nabla \alpha(\mathbf{x})}{\|\nabla \alpha(\mathbf{x})\|} \quad \textit{and} \quad P_I = P^\star\left(\frac{\nabla \alpha(\mathbf{x})}{\|\nabla \alpha(\mathbf{x})\|}\right).$$

Note that this case was previously derived in [51]. It also matches with the limit model of the numerical method described in [3], and matches in one dimension with the closure derived in [22].

1.3.3.3 Short memory case

$\check{\Sigma}$ being symmetric, it can be factorized as $\check{\Sigma} = Q^T\Lambda^2 Q$ where Q is orthogonal and Λ is diagonal, then the argument of the exponential of (1.16) is $\left\|\Lambda^{-1}Q(\mathbf{x}_d - \nabla m_k)\right\|^2$. As $\mathbf{x}_d$ takes all the values of $\mathbb{R}^d$, the following variable change may be performed $\mathbf{y}_d = \Lambda^{-1}Q\mathbf{x}_d$. The integral still depends on $\Lambda^{-1}Q\nabla m_k$, and may be asymptotically developed when $\left\|\Lambda^{-1}Q\nabla m_k\right\| \ll 1$. From a physical point of view, this matches with the case in which the size of the bubbles or droplets is much smaller than the lengthscale of ∇m_k.

Proposition 4 (Proposition 5.4 of [117])**.** *We denote by Λ^2 a diagonalization of $\check{\Sigma}$ and Q an orthogonal matrix such that $\check{\Sigma} = Q^T\Lambda^2 Q$. Also, we denote by $|\mathbb{V}_d|$ the volume of the d-dimensional unit ball, and by $\left|\mathbb{S}^{d-1}\right|$ the surface of the unit sphere in dimension d.*

If we suppose that $\left\|\Lambda^{-1}\nabla m_k(\mathbf{x})\right\| \ll 1$ then the system (1.17) is found with the following closure of the interfacial terms

$$\left\{\begin{array}{rl}
\mathbf{u}_I = & \dfrac{\check{\Sigma}^{-1}}{|\mathbb{V}_d|\left|\det\check{\Sigma}\right|^{1/2}}\displaystyle\int_{\mathbf{x}_d\in Q^T\Lambda(\mathbb{S}^{d-1})} u^\star_{\tilde{k}k}\left(\frac{\mathbf{x}_d}{\|\mathbf{x}_d\|}\right)\|\mathbf{x}_d\|\,\mathbf{x}_d\mathrm{d}\mathbf{x}_d \\
\mathbf{P}_I = & \dfrac{1}{|\mathbb{V}_d|\left|\det\check{\Sigma}\right|^{1/2}}\left(\displaystyle\int_{\mathbf{x}_d\in Q^T\Lambda(\mathbb{S}^{d-1})} P^\star_{\tilde{k}k}\left(\frac{\mathbf{x}_d}{\|\mathbf{x}_d\|}\right)\mathbf{x}_d\mathbf{x}_d^T\mathrm{d}\mathbf{x}_d\right)\check{\Sigma}^{-1} \\
(P\mathbf{u})_I = & \dfrac{\check{\Sigma}^{-1}}{|\mathbb{V}_d|\left|\det\check{\Sigma}\right|^{1/2}}\displaystyle\int_{\mathbf{x}_d\in Q^T\Lambda(\mathbb{S}^{d-1})} (Pu)^\star_{\tilde{k}k}\left(\frac{\mathbf{x}_d}{\|\mathbf{x}_d\|}\right)\|\mathbf{x}_d\|\,\mathbf{x}_d\mathrm{d}\mathbf{x}_d
\end{array}\right.$$

and with the following nonlinear relaxation terms

$$\left\{\begin{array}{rl}
\mathcal{R}_k^{(\alpha)} = & \dfrac{\exp(-m_k(\mathbf{x})^2/2)}{2\pi\,|\mathbb{V}_{d-1}|\left|\det\check{\Sigma}\right|^{1/2}}\displaystyle\int_{\mathbf{x}_d\in Q^T\Lambda(\mathbb{S}^{d-1})} (-u^\star_{\tilde{k}k})\left(\frac{\mathbf{x}_d}{\|\mathbf{x}_d\|}\right)\|\mathbf{x}_d\|\,\mathrm{d}\mathbf{x}_d \\
\mathcal{R}_k^{(\rho\mathbf{u})} = & \dfrac{\exp(-m_k(\mathbf{x})^2/2)}{2\pi\,|\mathbb{V}_{d-1}|\left|\det\check{\Sigma}\right|^{1/2}}\displaystyle\int_{\mathbf{x}_d\in Q^T\Lambda(\mathbb{S}^{d-1})} P^\star_{\tilde{k}k}\left(\frac{\mathbf{x}_d}{\|\mathbf{x}_d\|}\right)\mathbf{x}_d\mathrm{d}\mathbf{x}_d \\
\mathcal{R}_k^{(\rho E)} = & \dfrac{\exp(-m_k(\mathbf{x})^2/2)}{2\pi\,|\mathbb{V}_{d-1}|\left|\det\check{\Sigma}\right|^{1/2}}\displaystyle\int_{\mathbf{x}_d\in Q^T\Lambda(\mathbb{S}^{d-1})} (Pu)^\star_{\tilde{k}k}\left(\frac{\mathbf{x}_d}{\|\mathbf{x}_d\|}\right)\|\mathbf{x}_d\|\,\mathrm{d}\mathbf{x}_d.
\end{array}\right.$$

Note that the relaxation terms look rather as relaxation terms if the integral on the sphere $\mathbb{S}^{d-1}$ is done on a semi-sphere, by assembling the terms of opposite direction. Proposition 4 is obtained by doing hypothesis on the *quality of the mixture of the phases*. It can be further developed if we make hypothesis on the *relative state* of the two phases. As we are in the short memory case, the relaxation terms are dominant with respect to the nonconservative terms. It is therefore reasonable to consider that, even if the pressures and velocities of the two phases are not equal, they are likely to be close. In this case, the solution of the Riemann

problem can be linearized with the acoustic approximation:

$$\left\{\begin{array}{l} u^{\star}_{\bar{k}k}(\mathbf{x}_d) = \dfrac{Z_k\mathbf{u}_k + Z_{\bar{k}}\mathbf{u}_{\bar{k}}}{Z_k + Z_{\bar{k}}} \cdot \dfrac{\mathbf{x}_d}{\|\mathbf{x}_d\|} + \dfrac{P_{\bar{k}} - P_k}{Z_k + Z_{\bar{k}}} \\ P^{\star}_{\bar{k}k}(\mathbf{x}_d) = \dfrac{Z_{\bar{k}}P_k + Z_kP_{\bar{k}}}{Z_k + Z_{\bar{k}}} + \dfrac{Z_kZ_{\bar{k}}(\mathbf{u}_{\bar{k}} - \mathbf{u}_k)}{Z_k + Z_{\bar{k}}} \cdot \dfrac{\mathbf{x}_d}{\|\mathbf{x}_d\|} \end{array}\right. \tag{1.18}$$

where $Z_k = \rho_k c_k$ is the acoustic impedance.

Proposition 5 (Proposition 5.4 of [117]). *Suppose that hypothesis of Proposition 4 hold, and that (1.18) holds also. Then, defining the interfacial linearized expressions*

$$\mathbf{u}_I^L = \frac{Z_k\mathbf{u}_k + Z_{\bar{k}}\mathbf{u}_{\bar{k}}}{Z_k + Z_{\bar{k}}} \qquad P_I^L = \frac{Z_{\bar{k}}P_k + Z_kP_{\bar{k}}}{Z_k + Z_{\bar{k}}},$$

the relaxation coefficient

$$\mathscr{L}(\Lambda) = \frac{1}{|\mathbb{S}^{d-1}|}\int_{\mathbf{x}_d\in\mathbb{S}^{d-1}} \|\Lambda\mathbf{x}_d\| \, \mathrm{d}\mathbf{x}_d$$

and the relaxation matrix

$$\tilde{\Lambda} = Q^T\Lambda^2\left(\frac{1}{|\mathbb{S}^{d-1}|}\int_{\mathbf{y}_d\in\mathbb{S}^{d-1}} \frac{\mathbf{y}_d\mathbf{y}_d^T}{\|\Lambda\mathbf{y}_d\|}\mathrm{d}\mathbf{y}_d\right)Q$$

then system (1.17) is found with the following closure of the interfacial terms

$$\left\{\begin{array}{c} \mathbf{P}_I = P_I^L\mathrm{I}_d \\ \mathbf{u}_I = \mathbf{u}_I^L \\ (P\mathbf{u})_I = \dfrac{(Z_{\bar{k}}P_k + Z_kP_{\bar{k}})(Z_k\mathbf{u}_k + Z_{\bar{k}}\mathbf{u}_{\bar{k}}) + Z_kZ_{\bar{k}}(P_{\bar{k}} - P_k)(\mathbf{u}_{\bar{k}} - \mathbf{u}_k)}{(Z_k + Z_{\bar{k}})^2} \end{array}\right.$$

and the following linear, anisotropic relaxation terms

$$\left\{\begin{array}{l} \mathcal{R}_k^{(\alpha)} = \dfrac{\exp(-m_k(\mathbf{x})^2/2)\,|\mathbb{S}^{d-1}|\,\mathscr{L}(\Lambda)}{2\pi\,|\mathbb{V}_{d-1}|}\,\dfrac{P_k - P_{\bar{k}}}{Z_k + Z_{\bar{k}}} \\ \mathcal{R}_k^{(\rho\mathbf{u})} = \dfrac{\exp(-m_k(\mathbf{x})^2/2)\,|\mathbb{S}^{d-1}|}{2\pi\,|\mathbb{V}_{d-1}|}\,\dfrac{Z_kZ_{\bar{k}}}{Z_k + Z_{\bar{k}}}\,\tilde{\Lambda}\,(\mathbf{u}_{\bar{k}} - \mathbf{u}_k) \\ \mathcal{R}_k^{(\rho E)} = \mathbf{u}_I^L \cdot \mathcal{R}_k^{(\rho\mathbf{u})} - P_I^L\,\mathcal{R}_k^{(\alpha)} \end{array}\right.$$

Note that if $(P_{\bar{k}} - P_k)(\mathbf{u}_{\bar{k}} - \mathbf{u}_k)$ of $(P\mathbf{u})_I$ is neglected (which makes sense as it is a second order fluctuation), this leads to the interfacial terms of [131, 133].

Last, adding hypothesis on the isotropy of $\check{\Sigma}$, which is equivalent to considering isotropic spherical inclusions, the relaxation coefficients are isotropic

Proposition 6 (Proposition 5.5 of [117]). *In the same conditions as in Proposition 4, and if we suppose moreover that the auto-correlation function is isotropic: $\check{\Sigma} = \nu^2\mathrm{I}_d$, then the same closure for the interfacial terms as Proposition 4 is found, and the relaxation terms can be further simplified for giving a form similar as (1.1) with*

$$\left\{\begin{array}{l} \mu = \dfrac{\exp(-m_k(\mathbf{x})^2/2)\,|\mathbb{S}^{d-1}|\,\nu}{2\pi\,|\mathbb{V}_{d-1}|}\,\dfrac{1}{Z_k + Z_{\bar{k}}} \\ \lambda = \dfrac{\exp(-m_k(\mathbf{x})^2/2)\,|\mathbb{S}^{d-1}|\,\nu}{2\pi\,|\mathbb{V}_{d-1}|}\,\dfrac{Z_kZ_{\bar{k}}}{Z_k + Z_{\bar{k}}} \end{array}\right.$$

With the isotropic hypothesis, the expression of [132], for which the λ and μ relaxation coefficients of (1.1) are such that $\lambda = Z_kZ_{\bar{k}}\mu$, is recovered.

1.3.4 Properties of the models obtained

In this section, the properties of the new system (1.17) are discussed.

A first concern is the hyperbolicity of the model. Actually, the system (1.17) is so close of (1.1) that the same property holds: the system has a full set of eigenvectors in the direction $\boldsymbol{\xi}$ provided the system is not resonant, i.e. $\mathbf{u}_I \cdot \boldsymbol{\xi}$ is different from $\mathbf{u}_i \cdot \boldsymbol{\xi} \pm c_i$.

Another concern for the new system is the phasic entropy dissipation. In general, for (1.17), the following phasic entropy equation holds

$$\begin{aligned} &\partial_t \left(\alpha_k S_k(\mathbf{U}_k)\right) + \operatorname{div}_{\mathbf{x}} \left(\alpha_k S_k(\mathbf{U}_k)\right) \\ &\quad + \frac{1}{T_k} \left((P\mathbf{u})_I + P_k \mathbf{u}_k - P_k \mathbf{u}_I - \mathbf{u}_k \mathbf{P}_I\right) \nabla \alpha_k \\ &\qquad = \frac{1}{T_k} \left(-P_k \mathcal{R}_k^{(\alpha)} + \mathbf{u}_k \cdot \mathcal{R}_k^{(\rho \mathbf{u})} - \mathcal{R}_k^{(\rho E)}\right) \end{aligned} \tag{1.19}$$

Two types of terms may contribute to the phasic entropy production (1.19): the nonconservative contribution (in front of $\nabla \alpha_k$, in the left hand side of (1.19)), and the relaxation contribution (right hand side of (1.19)). It can then be proven that

- for the relaxation terms defined in Proposition 2, Proposition 4, Proposition 5, the relaxation contribution of the entropy equation is dissipative

  $$\frac{1}{T_k} \left(-P_k \mathcal{R}_k^{(\alpha)} + \mathbf{u}_k \cdot \mathcal{R}_k^{(\rho \mathbf{u})} - \mathcal{R}_k^{(\rho E)}\right) \leqslant 0,$$

  whereas for the model of Proposition 3, the relaxation terms are 0.

- for the case of Proposition 3, the non-conservative contribution of the entropy equation can be proven to be dissipative

  $$\frac{1}{T_k} \left((P\mathbf{u})_I + P_k \mathbf{u}_k - P_k \mathbf{u}_I - \mathbf{u}_k \mathbf{P}_I\right) \cdot \nabla \alpha_k \geqslant 0,$$

  whereas for the closures that were proposed in Proposition 5 and Proposition 6, the non-conservative contribution is 0.

Note that for the closure of [22], which matches with the closure of Proposition 5 where the second order terms are neglected, or for the closure of Proposition 4, the nonconservative contribution of entropy has no sign. However, these closures match with a flow for which the relaxation terms are dominants, and the relaxation terms are entropy dissipative. It can then be expected that the global phasic entropy contribution is dissipative.

1.4 Development of numerical schemes

The models derived (1.1),(1.17) raise several problems as far as their numerical approximation is concerned: beyond the definition of the modeling parameters, which was addressed in the previous sections, the diffuse interface models include nonconservative products, for which no satisfactory general method for their numerical approximation has ever been proposed. The numerical approximation relies on the method of [3], which circumvent the discretization of the nonconservative products by performing the numerical approximation on (1.9) and then averaging it, instead of approximating directly (1.1). The derivation is first exposed for finite volume schemes; this derivation leads to the numerical scheme of [3], but with a clearer stochastic model. Then its discontinuous Galerkin version is shown. Last, numerical limitation of the discontinuous Galerkin scheme is discussed.

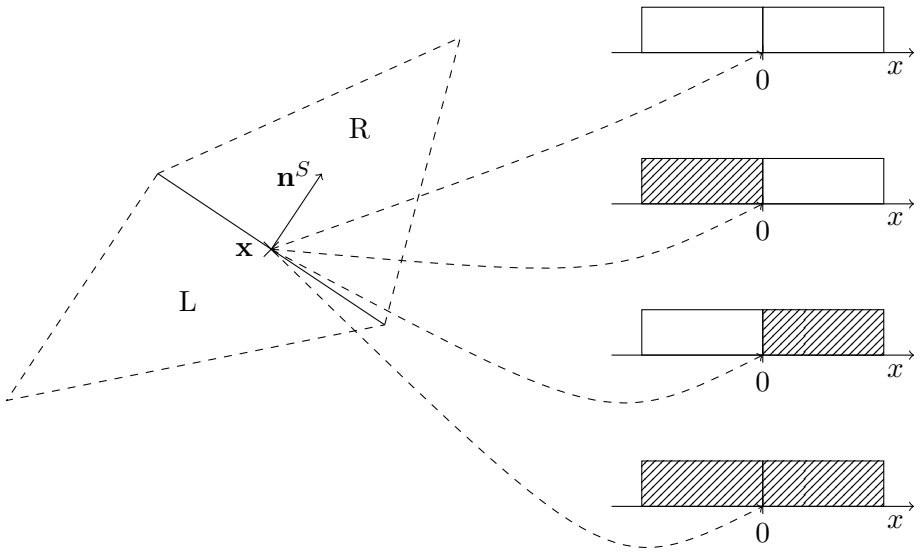


Figure 1.6: Decomposition of the side integral into four Riemann problems. The side S has a normal $\mathbf{n}^S$ and two adjacent cells, which we denote by L (where $\mathbf{n}^S$ is outward) and R (where $\mathbf{n}^S$ is inward). At the point $\mathbf{x}$, there might be four possible Riemann problems, which initial conditions are shown on the right of the figure. The first phase is in white whereas the second phase is shaded.

1.4.1 Finite volume scheme derivation

The aim of this section is the derivation of a numerical scheme for (1.1). Given a mesh $\mathcal{T}_h$, the solution is supposed to be piecewise constant; For the cell K_i, the numerical solution for the phase k is denoted by $\alpha_k^{(i)}\hat{\mathbf{U}}_k^{(i)}$. Also, we will denote by $\mathscr{U}^{(i)}$ the full vector of unknowns in the cell K_i. Based on the idea of [3], the numerical scheme relies on the approximation of (1.9), which is then averaged. We denote by $\mathcal{V}(i)$ the set of the neighbors of the cell K_i, by Γ_{ij} the side between the cells K_i and K_j, and by $\mathbf{n}_{ij}$ the normal to Γ_{ij} going from K_i to K_j. Then the evolution of the data on cell K_i reads

$$|K_i|\,\partial_t\left(\alpha_k^{(i)}\mathbf{U}_k^{(i)}\right) + \sum_{j\in\mathcal{V}(i)} |\Gamma_{ij}|\,\mathscr{F}_k\left(\mathscr{U}^{(i)}, \mathscr{U}^{(j)}, \mathbf{n}_{ij}\right) = 0.$$

It remains to compute the flux $\mathscr{F}$ on the side Γ_{ij}.

1.4.1.1 Integration of (1.9)

Starting from (1.9), four situations may happen on a side, as depicted in Figure 1.6. We denote by $\mathcal{F}_k^{[mn]}$ the flux that will be added for the fluid k on the cell K_i from the integration of the Riemann problem between the phases m and n.

$\chi_k^L(\mathbf{x}) = 1$ **and** $\chi_k^R(\mathbf{x}) = 1$ In this case, an example of the solution of the Riemann problem is shown on Figure 1.7. The integration is led as for a classical finite volume method [141]:

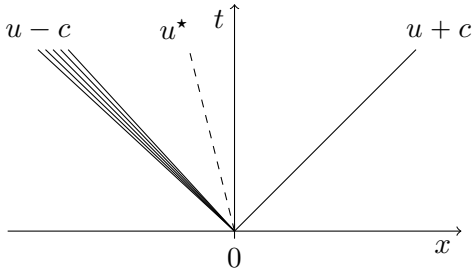


Figure 1.7: Example of solution of a Riemann problem involving a pure phase. In the case shown on the picture, the left wave is a rarefaction wave and the right wave is a shock. The integration is led classically by computing the flux at $x = 0$, and by adding this flux on the left and withdrawing it on the right.

the intermediate state U^{int} at $x = 0$ is computed, and then the flux $\mathbf{F}_k(U^{\text{int}})$ is added on the cell K_i. We denote this flux by $\mathbf{F}_k^{\text{eul}}(\mathbf{U}_k^L, \mathbf{U}_k^R, \mathbf{n}_{ij})$, and find

$$\mathcal{F}_k^{[kk]} = \mathbf{F}_k^{\text{eul}}(\mathbf{U}_k^L, \mathbf{U}_k^R, \mathbf{n}_{ij})$$

$\chi_k^L(\mathbf{x}) = 0$ **and** $\chi_k^R(\mathbf{x}) = 1$ This case is more difficult than the previous one, because the discontinuity of χ_k has to be taken into account in the integration. The position of this discontinuity leads to two cases which are shown in Figure 1.8, depending on the sign of the contact surface $u_{\bar{k}k}^\star$.

1. If $u_{\bar{k}k}^\star$ **is negative** then the fluid k is entering inside cell K_i. Integration is performed by adding the flux $\mathbf{F}_k(U_{\bar{k}k}^{\text{int}})$, where U^{int} is the state at $x = 0$ in the left picture of Figure 1.8. We denote this flux by $\mathbf{F}_k^{\text{eul}}(\mathbf{U}_{\bar{k}}^L, \mathbf{U}_k^R, \mathbf{n}_{ij})$. Also, the discontinuity of χ_k inside the cell K_i has to be taken into account. As for the derivation of the continuous system, this leads to a Lagrangian flux. Then the integration on the left leads to a flux

$$\mathbf{F}^{\text{lag}}(\mathbf{U}_{\bar{k}}^L, \mathbf{U}_k^R, \mathbf{n}_{ij}) = \begin{pmatrix} 0 \\ P_{\bar{k}k}^\star \mathbf{n}_{ij} \\ P_{\bar{k}k}^\star u_{\bar{k}k}^\star \end{pmatrix}$$

   Note that an index k is useless for the Lagrangian flux $\mathbf{F}^{\text{lag}}$, because this flux does not depend on the equation of state. From now on, we denote this flux by $\mathbf{F}^{\text{lag}}(\mathbf{U}_{\bar{k}}^L, \mathbf{U}_k^R, \mathbf{n}_{ij})$. Finally, the integration of this Riemann problem gives, on the phase k

$$\mathcal{F}_k^{[\bar{k}k]} = \mathbf{F}_k^{\text{eul}}(\mathbf{U}_{\bar{k}}^L, \mathbf{U}_k^R, \mathbf{n}_{ij}) - \mathbf{F}^{\text{lag}}(\mathbf{U}_{\bar{k}}^L, \mathbf{U}_k^R, \mathbf{n}_{ij})$$

2. If $u_{\bar{k}k}^\star$ **is positive** the integration on the cell K_i for the fluid k is straightforward: it is 0, as no fluid k is entering in the cell K_i.

To summarize the paragraph, the total flux added on the cell i, on the phase k, that comes from the integration of the Riemann problem with phase $\bar{k}$ on the left and phase k on the right is

$$\mathcal{F}_k^{[\bar{k}k]} = \mathscr{H}(u_{\bar{k}k}^\star(\mathbf{n}_{ij})) \left(\mathbf{F}_k^{\text{eul}}(\mathbf{U}_{\bar{k}}^L, \mathbf{U}_k^R, \mathbf{n}_{ij}) - \mathbf{F}^{\text{lag}}(\mathbf{U}_{\bar{k}}^L, \mathbf{U}_k^R, \mathbf{n}_{ij}) \right)$$

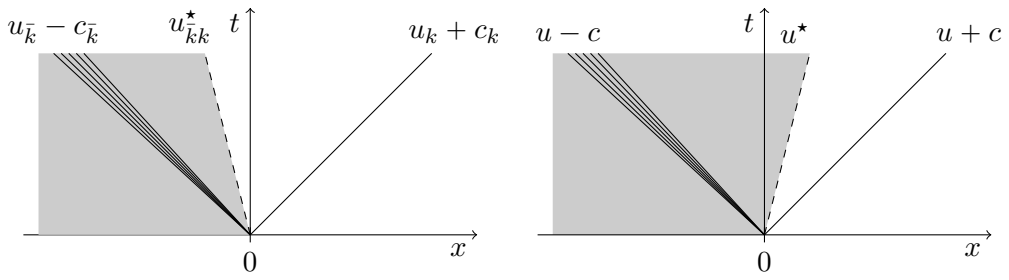


Figure 1.8: Example of solution of a Riemann problem involving two different phases. The phase k is in white whereas the phase $\bar{k}$ is shaded. The flux integration strongly depends on whether the velocity of the surface contact $u^\star_{\bar{k}k}$ is positive or negative. This contact surface matches with a jump of $\nabla\chi_k$, which integration raises a Lagrangian flux.

where $\mathscr{H}$ denotes the Heaviside function, i.e. $\mathscr{H}(x) = 1$ if $x > 0$, and $\mathscr{H}(x) = 0$ otherwise.

$\chi_k^L(\mathbf{x}) = 1$ and $\chi_k^R(\mathbf{x}) = 0$ Reasoning as for the case $\chi_k^L(\mathbf{x}) = 0$ and $\chi_k^R(\mathbf{x}) = 1$, we find that the total flux added on the cell i on the phase k, that comes from the integration of the Riemann problem with phase k on the left and phase $\bar{k}$ on the right is

$$\mathcal{F}_k^{[k\bar{k}]} = \mathscr{H}(u^\star_{k\bar{k}}(\mathbf{n}_{ij}))\mathbf{F}_k^{\text{eul}}(\mathbf{U}_k^L, \mathbf{U}_{\bar{k}}^R, \mathbf{n}_{ij}) + \mathscr{H}(-u^\star_{k\bar{k}}(\mathbf{n}_{ij}))\mathbf{F}^{\text{lag}}(\mathbf{U}_k^L, \mathbf{U}_{\bar{k}}^R, \mathbf{n}_{ij})$$

$\chi_1^L(\mathbf{x}) = 0$ and $\chi_1^R(\mathbf{x}) = 0$ In this case, the flux added on the phase k is 0

$$\mathcal{F}_k^{[\bar{k}\bar{k}]} = 0.$$

1.4.1.2 Averaging

The flux $\mathscr{F}_k$ at the side Γ_{ij} is the sum of the contributions of the four Riemann problems:

$$\mathscr{F}_k = \sum_{i=1}^{2}\sum_{j=1}^{2}\mathbb{P}(i,j)\mathcal{F}_k^{[ij]}.$$

It remains to compute the weights $\mathbb{P}(i,j)$.

1.4.1.3 Computation of the weights as in [51]

In this section, we aim at computing the weights between the Riemann problems on the sides. In [3], the weights were only *assumed* (see [3, p.376]). In [51], we proved that these weights could actually be *computed* by defining an appropriate stochastic modeling based on the volume fraction.

Proposition 7 (Proposition 1 of [51])**.** *If $g_{\mathbf{x}}$ is a stochastic Gaussian process such that for all $\mathbf{x}$ where α_k is continuous,*

$$\mathbb{E}\left[\frac{1 + \text{sgn}(g_{\mathbf{x}})}{2}\right] = \alpha_k(\mathbf{x})$$

with variance 1. *α is continuous inside the cells of a mesh, and might be discontinuous on a side. Then on a point of discontinuity of $\alpha_k(\mathbf{x})$,*

$$\lim_{t\to 0} \mathbb{E}\left[\chi_k^L(\mathbf{x}-t\mathbf{n})\chi_k^R(\mathbf{x}+\mathbf{n})\right] = \min(\alpha_k^L(\mathbf{x}), \alpha_k^R(\mathbf{x})). \tag{1.20}$$

Proof. For computing (1.20), we are interested in the joint behavior of $g_{\mathbf{x}-t\mathbf{n}}$ and $g_{\mathbf{x}+t\mathbf{n}}$. As $g_{\mathbf{x}}$ is a Gaussian process, the vector $[g_{\mathbf{x}-t\mathbf{n}}, g_{\mathbf{x}+t\mathbf{n}}]$ is a Gaussian vector. Its mean is $[m(\mathbf{x}-t\mathbf{n}), m(\mathbf{x}+t\mathbf{n})]$. Its covariance matrix is

$$C = \begin{pmatrix} 1 & R(\mathbf{x}-t\mathbf{n}, \mathbf{x}+t\mathbf{n}) \\ R(\mathbf{x}-t\mathbf{n}, \mathbf{x}+t\mathbf{n}) & 1 \end{pmatrix},$$

where R is the autocorrelation function. For the sake of simplicity, $R(\mathbf{x}-t\mathbf{n}, \mathbf{x}+t\mathbf{n})$ will be shortly noted as R also. Actually, the only property we will use is $\lim_{t\to 0} R(\mathbf{x}-t\mathbf{n}, \mathbf{x}+t\mathbf{n}) = 1$. We have

$$\det C = 1 - R^2 \qquad \text{and} \qquad C^{-1} = \frac{1}{1-R^2}\begin{pmatrix} 1 & -R \\ -R & 1 \end{pmatrix}.$$

Then we aim at evaluating the integral

$$I = \frac{1}{2\pi\,|\det C|^{1/2}} \int\!\!\int_{\mathbb{R}^2} \frac{1+\operatorname{sgn}(u_1)}{2}\frac{1+\operatorname{sgn}(u_2)}{2} \exp\left(-\frac{X^T C^{-1} X}{2}\right) \mathrm{d}u_1 \mathrm{d}u_2$$

with $X = (u_1 - m(\mathbf{x}-t\mathbf{n}), u_2 - m(\mathbf{x}+t\mathbf{n}))^T$. Note that C^{-1} can be factorized as

$$C^{-1} = A^T A \qquad \text{with} \qquad A = \frac{1}{\sqrt{2}}\begin{pmatrix} \dfrac{1}{\sqrt{1+R}} & \dfrac{1}{\sqrt{1+R}} \\ \dfrac{1}{\sqrt{1-R}} & -\dfrac{1}{\sqrt{1-R}} \end{pmatrix}.$$

Note also that

$$A^{-1} = \frac{1}{\sqrt{2}}\begin{pmatrix} \sqrt{1+R} & \sqrt{1-R} \\ \sqrt{1+R} & -\sqrt{1-R} \end{pmatrix}.$$

The variable change $V = AX$ is performed. Then

$$I = \frac{1}{2\pi} \int\!\!\int_{D_t} \exp\left(-\frac{V^T V}{2}\right) \mathrm{d}v_1 \mathrm{d}v_2,$$

where the domain D_t is obtained as the image of $\mathbb{R}^2_+$ by the variable change. The domain D_t is defined by the following inequalities

$$\begin{aligned} &\frac{1}{\sqrt{2}}\left(v_1\sqrt{1+R} + v_2\sqrt{1-R}\right) + m(\mathbf{x}-t\mathbf{n}) \geqslant 0, \\ &\frac{1}{\sqrt{2}}\left(v_1\sqrt{1+R} - v_2\sqrt{1-R}\right) + m(\mathbf{x}+t\mathbf{n}) \geqslant 0. \end{aligned}$$

When $t \to 0$, $R \to 1$, and the domain D_t converges to the domain D_0 defined as

$$D_0 = \left\{V \in \mathbb{R}^2 \qquad v_1 \geqslant \max(-m(\mathbf{x}^-), -m(\mathbf{x}^+))\right\}.$$

This leads to

$$\lim_{t\to 0} \mathbb{E}\left[\chi_k^L(\mathbf{x}-t\mathbf{n})\chi_k^R(\mathbf{x}+\mathbf{n})\right] = \frac{1}{2\pi}\left(\int_{-\infty}^{\infty} \mathrm{e}^{-v_2^2/2}\,\mathrm{d}v_2\right) \int_{\max(-m(\mathbf{x}^+), -m(\mathbf{x}^-))}^{\infty} \mathrm{e}^{-v_1^2/2}\,\mathrm{d}v_1.$$

It can be simplified as

$$\frac{1}{\sqrt{2\pi}}\int_{\max(-m(\mathbf{x}^+),m(\mathbf{x}^-))}^{\infty} \mathrm{e}^{-v_1^2/2}\,\mathrm{d}v_1.$$

By considering (1.3), it can be further simplified as

$$\lim_{t\to 0}\mathbb{E}\left[\chi_k^L(\mathbf{x}-t\mathbf{n})\chi_k^R(\mathbf{x}+\mathbf{n})\right] = \min(\alpha_k^L(\mathbf{x}),\alpha_k^R(\mathbf{x})).$$

□

1.4.1.4 Some remarks on the weights

By doing a finite differences consistency study on the weights found in Proposition 7, we can prove that these weights are consistent with the long memory continuous model of Proposition 3. It may be surprising to find only a single continuous model whereas a large variety of models was found in section 1.3, depending on the relative behavior of $\partial^2_{\mathbf{xy}}R(\mathbf{x},\mathbf{x})$ and $\nabla_{\mathbf{x}}\alpha_k$; this is even more surprising because no assumption on the autocorrelation function R is needed for the proof of Proposition 7, except for continuity. Actually, this is a direct consequence of the hypothesis of Proposition 7: indeed, α (and so m) is supposed to be discontinuous, whereas R is supposed to be continuous. This is then equivalent to have a infinite derivative of α across the sides, so that we recover the hypothesis $\|\nabla_{\mathbf{x}}m\| \gg \left\|\partial^2_{\mathbf{xy}}R(\mathbf{x},\mathbf{x})\right\|$, matching with the long memory case of Proposition 3.

The derivation of weights consistent with Proposition 4 and its simplifications of Proposition 5 and Proposition 6 is still an open question, however, it is clear that it should rely on the numerical approximation of the lifting of a normal gradient of the volume fraction on the sides, in order to recover a fair estimate of the ratio between $\|\nabla_{\mathbf{x}}m\|$ and $\left\|\partial^2_{\mathbf{xy}}R(\mathbf{x},\mathbf{x})\right\|$. As the process on the sides should be one dimensional, and as the one dimensional model of Proposition 2 is fully explicit, it may even be possible to derive weights consistent with the most general case (1.16).

1.4.2 Discontinuous Galerkin formulation

The aim of this section is to derive a discontinuous Galerkin scheme for the system (1.1). We first introduce some notations. The open set Ω is supposed to be meshed by a conforming mesh $\mathcal{T}_h$. We denote by $\mathcal{S}_i$ the set of the interior sides, and for simplifying, we suppose that we have only interior sides (which holds for example when periodic boundary conditions are used). All sides of $\mathcal{S}_i$ are supposed to be oriented, and its associated normal is denoted by $\mathbf{n}^S$. The approximation space is composed of functions that are piecewise polynomials: polynomials inside the cells of the mesh $\mathcal{T}_h$, and maybe discontinuous across the sides of $\mathcal{T}_h$. The approximation space is denoted by V_h. For all functions φ of V_h, and all sides S of $\mathcal{S}_i$, we denote:

$$\forall \mathbf{x}\in S \qquad \begin{cases} \varphi^R(\mathbf{x}) = \lim\limits_{\varepsilon\to 0,\varepsilon>0}\varphi(\mathbf{x}+\varepsilon\mathbf{n}^S) \\ \varphi^L(\mathbf{x}) = \lim\limits_{\varepsilon\to 0,\varepsilon>0}\varphi(\mathbf{x}-\varepsilon\mathbf{n}^S) \end{cases},$$

and for all $\mathbf{x}$ in an interior side S, we denote by:

$$[\![\,\varphi\,]\!](\mathbf{x}) = \varphi^L(\mathbf{x}) - \varphi^R(\mathbf{x}) \qquad \text{and} \qquad \{\!\{\,\varphi\,\}\!\}(\mathbf{x}) = \frac{\varphi^R(\mathbf{x})+\varphi^L(\mathbf{x})}{2}.$$

Then the space discretization of a general conservative hyperbolic system

$$\partial_t \mathbf{U} + \mathrm{div}_{\mathbf{x}} \mathbf{F}(\mathbf{U}) = 0,$$

consists in finding $\mathbf{U}_h \in V_h$ such that

$$\forall \varphi \in V_h \qquad \sum_{K\in\mathcal{T}_h} \int_K \varphi \partial_t \mathbf{U}_h - \sum_{K\in\mathcal{T}_h} \int_K \mathbf{F}(\mathbf{U}_h)\cdot\nabla\varphi + \sum_{S\in\mathcal{S}_i} \int_S [\![\varphi]\!] \{\!\{ \mathbf{F}(\mathbf{U}_h) \}\!\} \cdot \mathbf{n}^S = 0. \quad (1.21)$$

Note that formulation (1.21) is not L^2 stable, as centered numerical fluxes are used on the interior sides.

One particularity of the system (1.1) is that it includes nonconservative products. Because of the nonconservative products, the formulation (1.21) cannot be applied, and we propose to follow the formulation of [125]. This requires to define the jumps, thanks to the DalMaso-LeFloch-Murat [28] theory, which we briefly recall. The definition of nonconservative products developed in [28] consists in first defining a family of path Φ that connect two states $\mathbf{U}^L$ and $\mathbf{U}^R$ that are on both side of a discontinuity. For a nonconservative product $\sum_i A_i \partial_{\mathbf{x}_i} \mathbf{U}$, a family of path is defined as having the following properties:

- **Jump property**

$$\forall \mathbf{U}^L, \mathbf{U}^R \in \mathbb{R}^p \qquad \Phi(0; \mathbf{U}^L, \mathbf{U}^R, \mathbf{n}) = \mathbf{U}^L \quad \text{and} \quad \Phi(1; \mathbf{U}^L, \mathbf{U}^R, \mathbf{n}) = \mathbf{U}^R$$

- **Consistency**

$$\forall \mathbf{U} \in \mathbb{R}^p \quad \forall s \in \,]\,0\,;1\,[\qquad \Phi(s; \mathbf{U}, \mathbf{U}, \mathbf{n}) = \mathbf{U}$$

- **Lipschitz regularity** For any bounded set $\mathscr{U}$ of $\mathbb{R}^p$, there exists a constant $k > 0$ such that:

$$\left| \frac{\partial \Phi}{\partial s}\ (s; \mathbf{U}^L, \mathbf{U}^R, \mathbf{n}) - \frac{\partial \Phi}{\partial s}\ (s; \mathbf{V}^L, \mathbf{V}^R, \mathbf{n}) \right| \leqslant k \left| (\mathbf{U}^L - \mathbf{U}^L) - (\mathbf{V}^R - \mathbf{V}^R) \right|$$

with all these hypothesis, it is proved in [28] that the nonconservative product can be defined as a Borel measure μ. When $\mathbf{U}$ is smooth, this measure corresponds to a classical integration. When $\mathbf{U}$ is discontinuous at a point $\{\mathbf{x}_0\}$ in the direction $\mathbf{n}$, it is equal to:

$$\mu(\{\mathbf{x}_0, \mathbf{n}\}) = \int_0^1 \sum_i A_i(\Phi(s, \mathbf{U}^L(\mathbf{x}_0), \mathbf{U}^R(\mathbf{x}_0), \mathbf{n}))\mathbf{n}_i \frac{\partial \Phi}{\partial s}(s, \mathbf{U}^L(\mathbf{x}_0), \mathbf{U}^R(\mathbf{x}_0), \mathbf{n})\, \mathrm{d}s$$

This definition allows to define the following discontinuous Galerkin formulation with centered fluxes: find $\mathbf{U}_h \in V_h$ such that

$$\begin{aligned} \forall \varphi \in V_h \qquad & \sum_{K\in\mathcal{T}_h} \int_K \varphi \partial_t \mathbf{U}_h - \sum_{K\in\mathcal{T}_h} \int_K \mathbf{F}(\mathbf{U}_h)\nabla\varphi \\ & + \sum_{K\in\mathcal{T}_h} \int_K \varphi \sum_{i=1}^{3} A_i(\mathbf{U}_h) \frac{\partial \mathbf{U}_h}{\partial x_i} + \sum_{S\in\mathcal{S}_i} \int_S [\![\varphi]\!] \{\!\{ \mathbf{F}(\mathbf{U}_h) \}\!\} \cdot \mathbf{n}^S \\ & + \sum_{S\in\mathcal{S}_i} \int_S \{\!\{ \varphi \}\!\} \int_0^1 \sum_i A_i(\mathbf{U}_h) \mathbf{n}_i^S \frac{\partial \Phi}{\partial s}(s, u^L, u^R, \mathbf{n}^S). \end{aligned} \quad (1.22)$$

For obtaining L^2 stability, integration on the interior sides are replaced by numerical fluxes. By considering the numerical fluxes derived for the finite volume scheme of subsection 1.4.1, we directly see that the Eulerian fluxes match with the conservative flux (last term of second line of (1.22)), whereas the Lagrangian fluxes match with the nonconservative fluxes (third line of (1.22)); the fluxes $\mathcal{F}^{[ij]}$ are split into

- Their Eulerian part $\mathcal{F}^{[ij],\text{eul}}$
- The Lagrangian part added on the left $\mathcal{F}^{[ij],\text{lag,L}}$
- The Lagrangian part added on the right $\mathcal{F}^{[ij],\text{lag,R}}$

Then we define

$$\begin{cases} \mathscr{F}_k^{\text{Eul}} = \displaystyle\sum_{i=1}^{2}\sum_{j=1}^{2} \mathbb{P}(i,j)\mathcal{F}_k^{[ij],\text{Eul}} \\ \mathscr{F}_k^{\text{lag},L} = \displaystyle\sum_{i=1}^{2}\sum_{j=1}^{2} \mathbb{P}(i,j)\mathcal{F}_k^{[ij],\text{Lag},L} \\ \mathscr{F}_k^{\text{lag},R} = \displaystyle\sum_{i=1}^{2}\sum_{j=1}^{2} \mathbb{P}(i,j)\mathcal{F}_k^{[ij],\text{Lag},R} \end{cases}.$$

and use the following upwinding version for the numerical fluxes

$$\begin{cases} \displaystyle\int_S [\![\varphi]\!] \{\!\{ \mathbf{F}(\mathbf{U}_h) \}\!\} \cdot \mathbf{n}^S \approx \int_S [\![\varphi]\!] \mathscr{F}_k^{\text{Eul}} \\ \displaystyle\int_S \{\!\{ \varphi \}\!\} \int_0^1 \sum_i A_i(\mathbf{U}_h)\mathbf{n}_i^S \frac{\partial \Phi}{\partial s}(s, u^L, u^R, \mathbf{n}^S) \approx \int_S \varphi^L \mathscr{F}_k^{\text{lag},L} + \int_S \varphi^R \mathscr{F}_k^{\text{lag},R} \end{cases},$$

which is the natural extension of the finite volume scheme of subsection 1.4.1.

1.4.3 Limiters

1.4.3.1 Stabilization for shocks

As explained in subsection 1.4.1, the weights found in [51] are matching with the model of Proposition 4, suitable only for interface flows. This justifies to perform the limitation as follows

- If the volume fraction is either equal to 1 or equal to 0, then the conservative variables of the phase that exists are limited with classical limiters for the Euler system.
- If the volume fraction is neither equal to 0 nor to 1, then in each cell, the following steps are followed
  1. Compute the average of each of the conservative variables $\alpha_k\rho_k$, $\alpha_k\rho_k\mathbf{u}_k$ and $\alpha_k\rho_k E_k$. We denote the average by $\langle\alpha_k\rho_k\rangle$, $\langle\alpha_k\rho_k\mathbf{u}_k\rangle$ and $\langle\alpha_k\rho_k E_k\rangle$
  2. Limit the volume fraction. This is done as for example in [24]. The new volume fraction is denoted by $\tilde{\alpha}_k(\mathbf{x})$.

3. Rebuild the conservative variables as

$$\begin{aligned}(\alpha_k\rho_k)(\mathbf{x}) &= \tilde{\alpha}_k(\mathbf{x})\frac{\langle\alpha_k\rho_k\rangle}{\langle\alpha_k\rangle}\\ (\alpha_k\rho_k\mathbf{u}_k)(\mathbf{x}) &= \tilde{\alpha}_k(\mathbf{x})\frac{\langle\alpha_k\rho_k\mathbf{u}_k\rangle}{\langle\alpha_k\rangle}\\ (\alpha_k\rho_k E_k)(\mathbf{x}) &= \tilde{\alpha}_k(\mathbf{x})\frac{\langle\alpha_k\rho_k E_k\rangle}{\langle\alpha_k\rangle}\end{aligned}$$

In the practical applications, the minmod limiter is used.

1.4.3.2 Maximum preserving limiter

This limiter is designed for interface flows in which no shock may occur. In this case, the only concern is to preserve the volume fraction between 0 and 1. The limiter is based on the ideas of [148, 149], which consists in first proving a maximum preserving property on the average on each cell, and then on using a scaling limiter for ensuring the maximum property on a set of points of the cell.

We give here some more details on this method. Given a cell K_i and the set of its neighbors $\mathcal{V}(i)$, a finite volume methods with explicit time stepping reads

$$u_i^{n+1} = u_i^n - \delta t \sum_{j\in\mathcal{V}(i)} \frac{|\Gamma_{ij}|}{|K_i|}\,\tilde{f}(u_i^n, u_j^n, \mathbf{n}_{ij}). \tag{1.23}$$

For a linear problem with velocity $\mathbf{c}$, a CFL estimate for ensuring positivity is

$$\delta t \leqslant \frac{|K_i|}{|\mathbf{c}|\sum_{j\in\mathcal{V}(i)}|\Gamma_{ij}|}.$$

For nonlinear problems, provided $\tilde{f}$ is monotone, we denote by $\lambda(u_i, u_j, \mathbf{n}_{ij})$ the scalar such that the numerical scheme

$$u_i^{n+1} = u_i^n - \frac{\delta t}{L}\tilde{f}(u_i^n, u_j^n, \mathbf{n}_{ij})$$

ensures the maximum preservation property for $\delta t \leqslant L\lambda(u_i, u_j, \mathbf{n}_{ij})$. Then rewriting the multidimensional finite volume scheme (1.23) as

$$u_i^{n+1} = \sum_{j\in\mathcal{V}(i)} \frac{1}{|\mathcal{V}(i)|}\left(u_i^n - \frac{\delta t\,|\mathcal{V}(i)|\,|\Gamma_{ij}|}{|K_i|}\tilde{f}(u_i^n, u_j^n, \mathbf{n}_{ij})\right)$$

ensures the maximum preservation property under the CFL condition

$$\delta t \leqslant \min_{j\in\mathcal{V}(i)} \frac{|K_i|\,\lambda(u_i, u_j, \mathbf{n}_{ij})}{|\mathcal{V}(i)|\,|\Gamma_{ij}|}.$$

We consider now the discontinuous Galerkin method with the explicit Euler method for time integration; we denote by u_{K_i} the polynomial approximation in the cell K_i

$$\forall\varphi \qquad \int_{K_i}\varphi u_{K_i}^{n+1} = \int_{K_i}\varphi u_{K_i}^n + \delta t\int_{K_i} f(u_{K_i})\cdot\nabla\varphi - \delta t\sum_{j\in\mathcal{V}(i)}\int_{\Gamma_{ij}}\tilde{f}(u_{\{K_i;\Gamma_{ij}\}}, u_{\{K_j;\Gamma_{ij}\}}, \mathbf{n}_{ij})$$

where $u_{\{K_l;\Gamma_{ij}\}}$ stands for the value of the cell l on the side Γ_{ij}. Taking $\varphi = 1$ gives the evolution equation on the average on the cell $\bar{u}_{K_i}$.

$$|K_i|\,\bar{u}_{K_i}^{n+1} = |K_i|\,\bar{u}_{K_i}^{n} - \delta t \sum_{j\in\mathcal{V}(i)} \int_{\Gamma_{ij}} \tilde{f}(u_{\{K_i;\Gamma_{ij}\}}, u_{\{K_j;\Gamma_{ij}\}}, \mathbf{n}_{ij})$$

Denoting by $\left\{\omega_k^j, \mathbf{x}_k^j\right\}$ the quadrature formula on the reference element matching with the face Γ_{ij}, the fully discrete scheme is

$$\bar{u}_{K_i}^{n+1} = \bar{u}_{K_i}^{n} - \delta t \sum_{j\in\mathcal{V}(i)} \frac{|\Gamma_{ij}|}{|K_i|} \sum_k \omega_k^j \tilde{f}(u_{K_i}(\mathbf{x}_k^j), u_{K_j}(\mathbf{x}_k^j), \mathbf{n}_{ij}).$$

Supposing that a k–exact quadrature formula exists for the cell K_i, including all the side quadrature points, and with strictly nonnegative weights. We denote by $\bar{\omega}_l$ the weights of this quadrature formula, and by $\bar{\mathbf{x}}_l$ the interior points of the quadrature formula. Then

$$\bar{u}_{K_i}^{n} = \sum_l \bar{\omega}_l u_{K_i}(\bar{\mathbf{x}}_l) + \sum_{j\in\mathcal{V}(i)} \sum_k \bar{\omega}_k^j u_{K_i}(\bar{\mathbf{x}}_k^j),$$

and the numerical scheme can be written

$$\bar{u}_{K_i}^{n+1} = \sum_l \bar{\omega}_l u_{K_i}(\bar{\mathbf{x}}_l) + \sum_{j\in\mathcal{V}(i)} \sum_k \bar{\omega}_k^j \left(u_{K_i}(\bar{\mathbf{x}}_k^j) - \frac{\delta t\,|\Gamma_{ij}|\,\omega_k^j}{|K_i|\,\bar{\omega}_k^j}\, \tilde{f}(u_{K_i}(\mathbf{x}_k^j), u_{K_j}(\mathbf{x}_k^j), \mathbf{n}_{ij}) \right).$$

and the maximum preserving property for the average in the cell is ensured under the CFL condition

$$\delta t \leqslant \min_{j\in\mathcal{V}(i)} \frac{|K_i|\,\lambda(u_i, u_j, \mathbf{n}_{ij})\omega_k^j}{|\Gamma_{ij}|\,\bar{\omega}_k^j}.$$

The existence of such a cell quadrature formula is ensured for triangles [149], and quads or hexas (Gauss-Lobatto points match all the required properties), but is still an open question in general up to our knowledge. Last, by denoting by $\mathcal{P}$ the set of points on which the maximum property is required, which should include not only the points of the k-exact formula, but also the Gauss points used for integrating the cell terms of the discontinuous Galerkin method, then the following scaling limiter is used for ensuring the bound $\leqslant M$

$$\tilde{u}_{K_i} = \bar{u}_{K_i} + \alpha\,(u_{K_i} - \bar{u}_{K_i})$$

with

$$M_i = \max_{\mathbf{x}\in\mathcal{P}} u_{K_i}(\mathbf{x}) \qquad \alpha = \min\left(\frac{M - \bar{u}_{K_i}}{M_i - \bar{u}_{K_i}}, 1\right).$$

This limiter is applied with $M = 1$ on the volume fraction, and a similar fix is used for ensuring that the volume fraction is positive.

1.5 Numerical results

1.5.1 Consistency test

This first test case is done for ensuring that the weights of subsection 1.4.1 match with the closure of Proposition 4 in one dimension. For this, a self similar solution based on the wave

$\mathbf{u}_I$ is used. The volume fraction is set to an analytical function, and the other variables are set by using the Riemann invariants associated with the wave $\mathbf{u}_I$. The analytical solution at time t is computed by the characteristics method. For ensuring that the characteristics never cross, $\mathbf{u}_I$ is set to a strictly increasing function. All the parameters are detailed in [51]. The computation is led until $t = 0.15s$, with a mesh with 100, 200, 400 and 800 cells. The results on each of the variables is shown in Figure 1.9, whereas the convergence order is shown in Figure 1.10. Both prove the convergence of the numerical scheme of subsection 1.4.1 with the analytical solution computed with the model of Proposition 4.

1.5.2 Multiphase vortex advection

When it comes to test the convergence order of a numerical method for a nonlinear hyperbolic model, a usual test consists in the advection of a vortex. This test is interesting because it is smooth, but is more complicated than an entropic wave. In this case, the two phases have the same pressure and velocity, so that the multiphase test case of [39] (which deals with the original Baer-and-Nunziato model) is suitable. Results obtained for the convergence order are shown in Figure 1.11. The optimal order $k + 1$ is observed for DG^k.

1.5.3 Shock/Bubble interaction

Our aim here is to compute the test proposed by Quirk & Karni [123], based on the experiments of Haas & Sturtevant [62]. This test is a classical benchmark for diffuse interface methods [66, 65]. As described in Figure 1.12 a shock wave moving to the left from the right part of the domain will interact with an Helium bubble. The Mach number of the shock is 1.22, which implies that behind the shock wave, air has a density of $1.92691\,\mathrm{kg}\cdot\mathrm{m}^{-3}$, a velocity of $-0.33361\,\mathrm{m}\cdot\mathrm{s}^{-1}$, and a pressure of $1.5698\,\mathrm{Pa}$. The fluids involved are such that the Atwood number is negative, which is referred as the divergent case in the previous publications. It is then known that it will lead to a reversal of the bubble that is penetrated by a jet of surrounding fluid and then gives two vortex downstream of the flow.

In this test, an unstructured triangular conforming mesh is used, and contains 14086 triangles, generated with `Gmsh`.

Numerical results are shown in Figure 1.13. Mixture density evolution is shown, with the corresponding isovalues, for the DG^0 and the DG^1 schemes. Here again, this test is well described by both schemes which give qualitatively good results. The reversal of the bubble and the interface is better described with the DG^1 scheme. Finally, a zoom on the isovalues of the volume fraction is presented in Figure 1.14 at time $t = 230\,\mathrm{ms}$. The interface is clearly more sharpen in the DG^1 case.

1.5.4 Accuracy of the maximum preserving limiter

For testing the accuracy of the maximum preserving limiter, two tests are proposed. The first one consists in checking the effect of the limiter on the accuracy of the advection of a discontinuity. A uniform velocity of 1 is imposed. The computation is led until time 1. The solution obtained for a mesh of 64 cells is depicted on the left of Figure 1.15. The convergence order is shown on the right of Figure 1.15. Of course, we do not find an order accuracy of 2 and 3 for the degree 1 and 2 approximation, because the solution is not regular. Nevertheless, we can observe that the convergence order is not destroyed by the slope limiter.

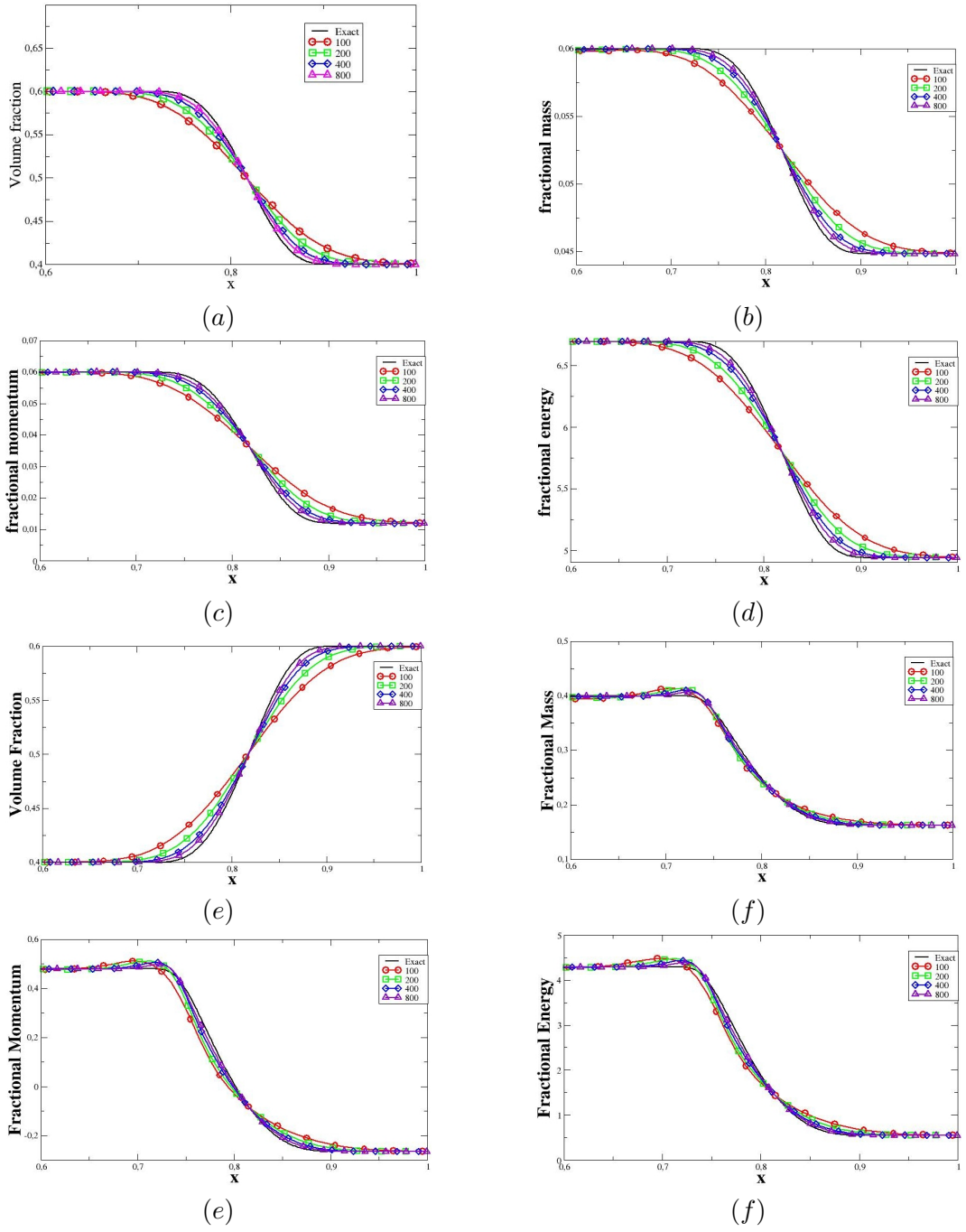


Figure 1.9: Results for the consistency test for all the variables. Exact solutions are compared with the solutions computed with 100, 200, 400 800 cells. Convergence to the exact solution is observed for all the variables of fluid 1 ((*a*),(*b*),(*c*),(*d*)), and of fluid 2 ((*e*),(*f*),(*g*),(*h*)).

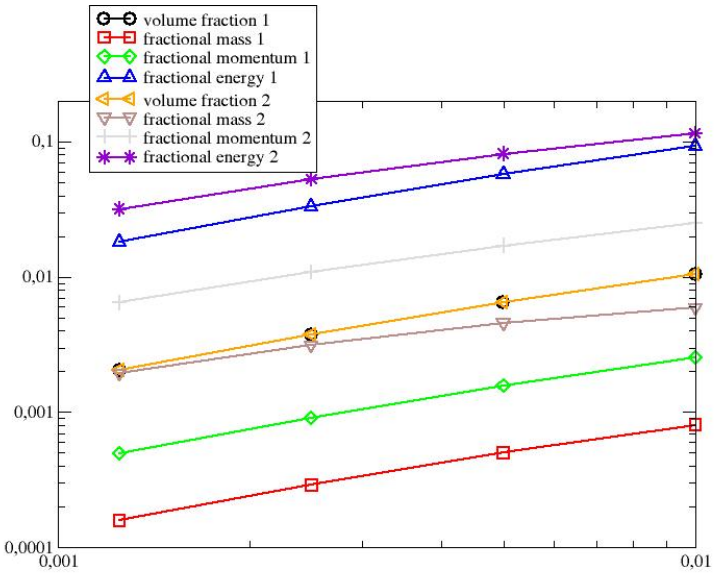


Figure 1.10: The convergence order of the consistency test is drawn for all the conservative variables computed: the L^2 error is computed for the mesh with 100, 200, 400 and 800 cells. For all the variables, the convergence is proved to be close to 1.

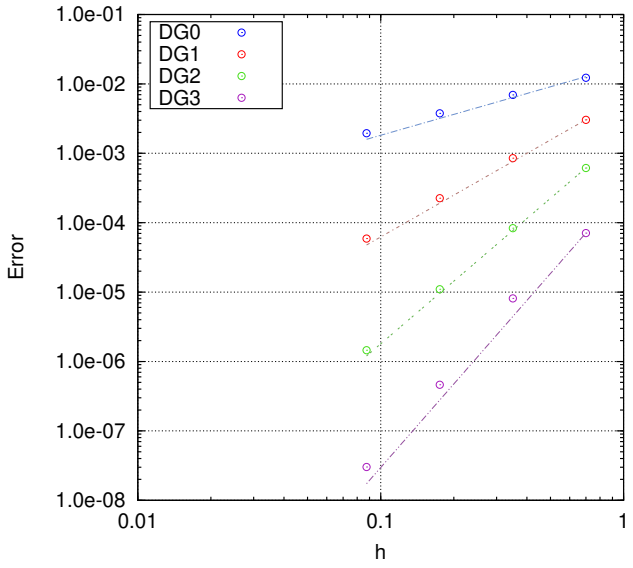


Figure 1.11: Convergence order for the test case of the multiphase vortex.

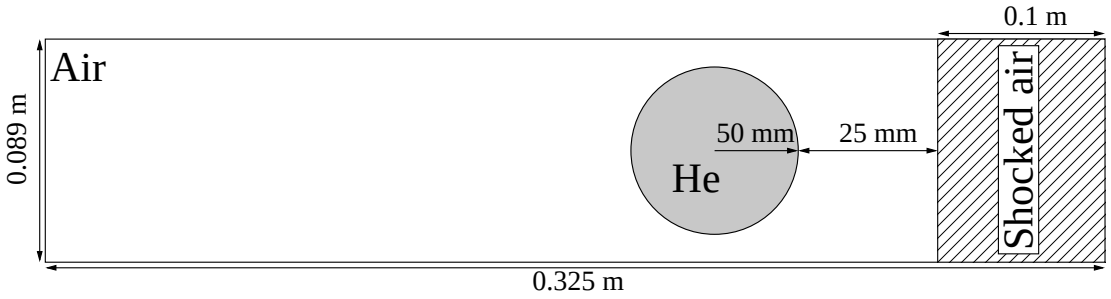
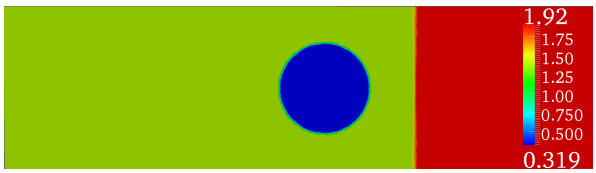


Figure 1.12: Computational domain for the shock bubble interaction.

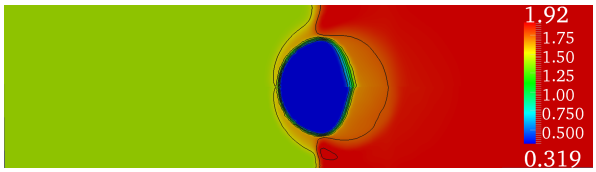
A second test is performed with the maximum preserving limiter: the Zalesak test [147]. This test is usually used for showing the ability of level-set methods to sharply resolve interfaces. Level-set methods often rely on high order non compact schemes for the advection of the level-set function (typically WENO-5), which is best suited with structured meshes. We show the results obtained with one rotation of the shape on an unstructured mesh with 10 points on each side of the square domain. Numerical results are shown in Figure 1.16. This Figure proves the much stronger accuracy of the third order computation, which is the only one able to conserve the initial shape. This test proves also that provided high order methods are used, methods based on diffuse interface methods are competitive with respect to Level-set methods. We recall that diffuse interface methods are conservative, contrarily to Level-Set methods, and so are better suited for the approximation of flows with shocks.

1.5.5 Detonation test case

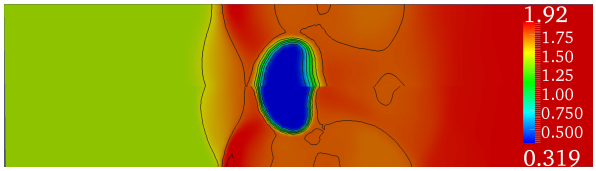
This test case was proposed in [52]. The numerical method developed in this chapter was extended to the simulation of permeable fronts by using the method of [83]. The Figure 1.7 and Figure 1.8 with classical Riemann problems are replaced by the reactive Riemann problems depicted in Figure 1.17. In such case, if a non-monotone limiter is used, the scheme fails to determine the correct state (velocity of the detonation front, CJ state...), as shown in Figure 1.18, and that is why our computations were made with a monotone limiter. This behavior is of particular importance to be mentioned. The need for a monotone limiter can be understood as follows. In the numerical scheme, the weights depend on the sign of $\nabla\alpha_k \cdot \mathbf{n}$ where $\mathbf{n}$ is the normal to the side. In the consistent continuous limit of Proposition 3, expressions of $\mathbf{u}_I$ and P_I, which are $\mathbf{u}_I = u^\star(\nabla\alpha_1)\mathbf{n}(\nabla\alpha_1)$ and $P_I = P^\star(\nabla\alpha_1)$, depend singularly on the sign of $\nabla\alpha_1$. For example, in one dimension, P_I and u_I switch from the interfacial pressure and velocity of the Riemann problem $1-2$ to $2-1$ when the sign of $\partial_x\alpha_1$ changes. Consequently, it is important that $\partial_x\alpha_1$ remains monotone at the discrete level. We point out that as the singularity depends only on $\nabla\alpha_1$, the requirement of a monotone limiter lies only on the volume fractions. In the reactive case, the problem has essentially the same origin.



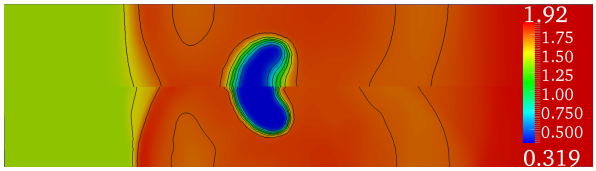
(a) $t = 0\,\text{ms}$.



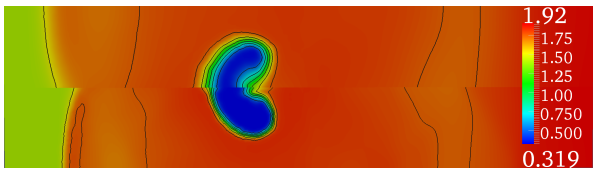
(b) $t = 43.954\,\text{ms}$.



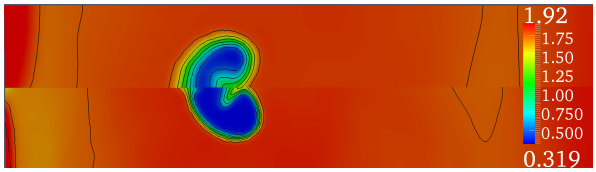
(c) $t = 82.923\,\text{ms}$.



(d) $t = 121.507\,\text{ms}$.



(e) $t = 158.71\,\text{ms}$.



(f) $t = 195.029\,\text{ms}$.

Figure 1.13: Interaction of a shock wave with an helium bubble in air. Contours of the mixture density for the DG^0 (top) and the DG^1 (bottom) schemes. Isovalues are also represented in black lines.

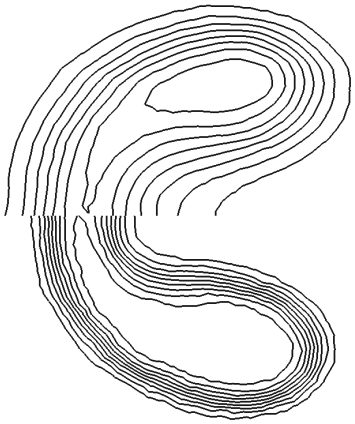


Figure 1.14: Interaction of a shock wave with an helium bubble in air. Isovalues of the volume fraction for the DG^0 (top) and the DG^1 (bottom) solutions.

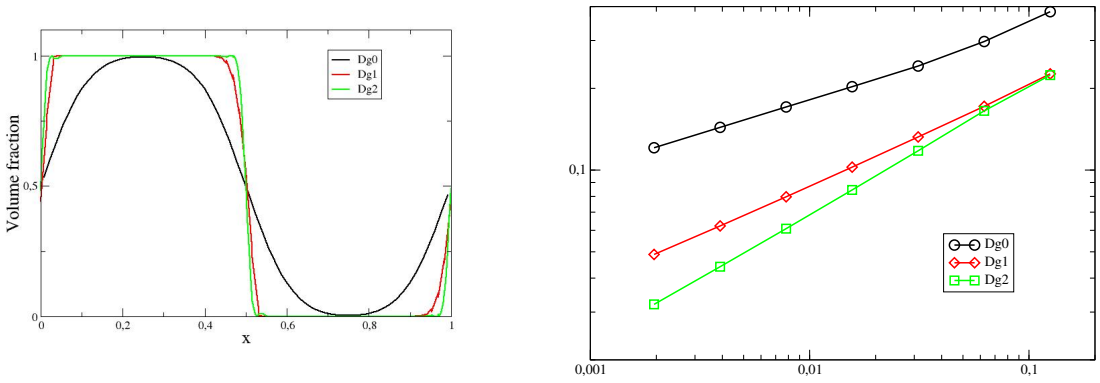


Figure 1.15: **Left:** solution obtained for the advection of a one dimensional discontinuity on a mesh of 64 cells with approximation degree of 0, 1 and 2. **Right:** Convergence error for the advection of a one dimensional discontinuity, represented on a log-log scale.

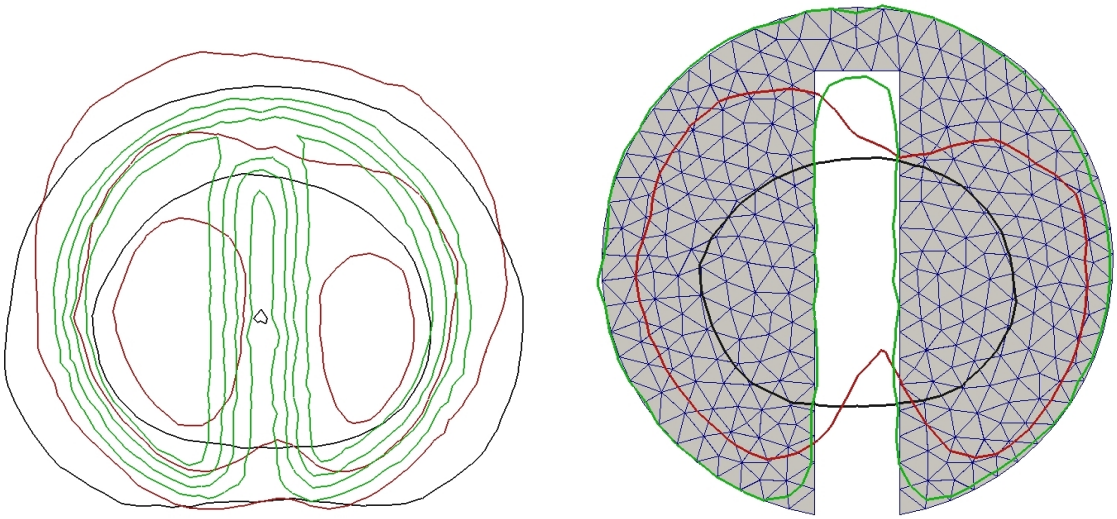


Figure 1.16: Results for Zalesak test. On the left figure, we represented the isovalues 0.2,0.4,0.6 and 0.8 for the Dg0 (black), Dg1 (red) and Dg2 (green) computations. Note that some of these isovalues are not in the Figure because they are too far from the initial shape. On the right side, the isovalue 0.5 of α_1 is drawn, and compared with the initial shape.

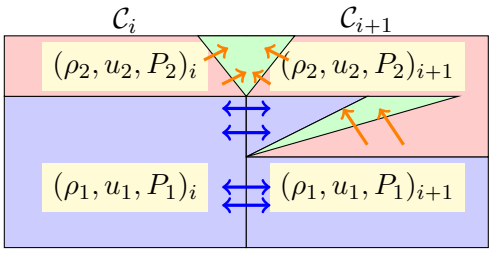
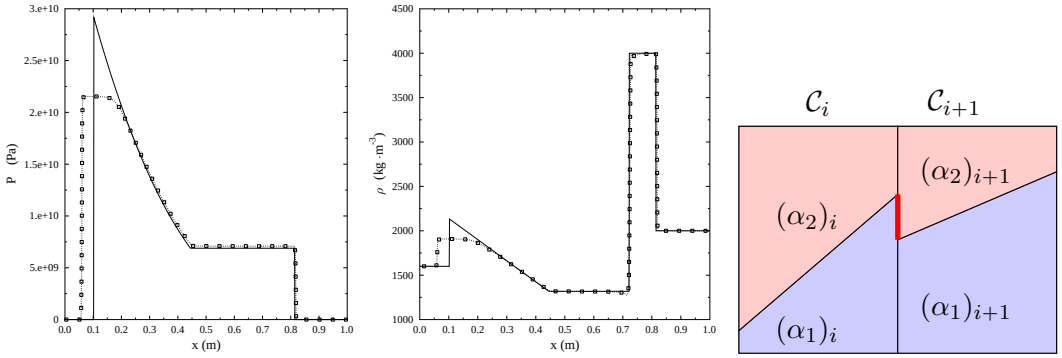
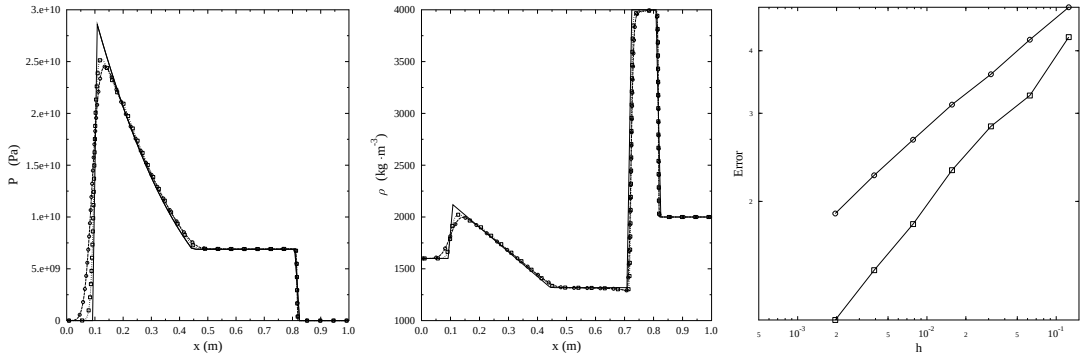


Figure 1.17: Configuration of the reactive Riemann problem on a side between two adjacent cells. A reactive front (materialized by orange arrows) is responsible for the disappearance of fluid 2 which reacts to form a new state composed of fluid 1 (green).



(a) Pressure profiles. (b) Density profiles. (c) Configuration of the Riemann problem on a frontier.

Figure 1.18: 1D detonation. Incorrect solutions obtained with a non-monotone limiter. As shown on Fig. 1.18(a) and 1.18(b), the numerical scheme might converge to a wrong solution if a non-monotone limiter is used. Origin of the problem is shown on Fig. 1.18(c): the non-monotone behavior of the limiter induces a blue-red Riemann problem (represented by a strong line in red), which does not match with the physical reaction which is normally driven by reactive red-blue Riemann problems.



(a) Pressure profiles. (b) Density profiles. (c) Convergence order.

Figure 1.19: 1D detonation. dG^0 (circle), dG^1 (square) and exact (line) solutions.

1.6 Conclusion and prospects on compressible multiphase flows

In section 1.2, a simple way of modeling a random mixture of two non miscible media from a stochastic process was exposed. Depending on the parameters of the stochastic process (and more precisely on the Hessian of its auto-correlation function) the modeling is able to deal both with well defined interfaces and with well mixed media. From this modeling, and by relying only on the solution of the Riemann problem, several diffuse interface models were obtained in section 1.3. This derivation method allowed to propose modifications of previous closure for taking into account anisotropy of subscales, and gave clear hypothesis for the range of validity of the models. All the models obtained include nonconservative products, which are ambiguous to discretize. The finite volume method of [3] was revisited by using the same stochastic modeling as for the derivation of the continuous model. This allowed to give explicitly the continuous model consistent with [3], which was mandatory for defining a discontinuous Galerkin numerical scheme. The numerical scheme was implemented and validated. Two types of stabilization have been tried: classical minmod limiters and positivity preserving limiters of [149].

Starting from the methods developed in this chapter, the following research directions will be considered in the coming years

Address the modeling for more than two phases. The modeling of two phase flows relies on the Heaviside function applied to a Gaussian process depending on a single random variable. Addressing more than two phases flows by keeping a symmetry between the phases is for the moment an open question. A major difference with respect to the two phase flow model is that most of the models proposed [63, 64, 67, 96] do not link the closure parameters with the subscale topology of the flow. Therefore, finding such a stochastic model for several phases, and deriving the associated multiphase flow would be a decisive step forward in the modeling of multiphase flows. Having already tried to design a stochastic model able to simulate three phase flow (and having failed until now) makes me think that the gap between two and three phases flows is very challenging.

Address the modeling of a broader range of physical phenomena. A quite simple problem to address with the same type of modeling is the surface tension between the phases. Some models have already been proposed for this topic [131, 132, 134]. This problem could be addressed by directly applying the Young-Laplace equation inside the Riemann problem that is then averaged.

Also, a natural extension of the models presented here consists in considering diffusive terms, namely using Navier-Stokes equations instead of Euler equations for including viscous effects in the closure and relaxation terms. In this case, relaxation terms similar as the ones proposed in [15] should be found, at least in some regimes.

Address the problem of the modeling of the subscales. All the models but the one of Proposition 3 depend on subscale parameters, namely $\partial^2_{\mathbf{xy}} R(\mathbf{x}, \mathbf{x})$. For closing the model, an equation for the evolution of these subscale parameters is required. This problem was addressed in [53] for an isotropic subscale behavior, but the general case is an open question. If experimental data were available, this problem could be a good candidate for being addressed with machine learning techniques.

Develop numerical schemes for a broader range of models. The finite volume scheme and the discontinuous Galerkin scheme proposed in this chapter are all consistent with the system of Proposition 3. As discussed in subsubsection 1.4.1.4, a larger diversity of models was found in section 1.3. By changing the numerical scheme, but keeping the ideas that drove its derivation, it should be possible to derive finite volume and discontinuous Galerkin schemes consistent with the other models of section 1.3. Also, the numerical scheme should be adapted to a number of phase greater than two, and to other physical phenomena such as capillarity or diffusion, once the derivation of the continuous models will be clarified.

Analyze the numerical schemes obtained. Up to our knowledge, few work has been driven on the theoretical analysis of schemes for Baer-and-Nunziato models. Some properties are easy to address for first order schemes, like the positivity of volume fraction or positivity of the densities. Some other aspects, like fully discrete entropy inequalities or temperature positivity have hardly been addressed. As the numerical schemes we should obtain by a similar method as in section 1.4 rely on averaging Godunov methods with positive weights, it should makes easier the derivation of such inequalities.

Develop more sophisticated stabilizations. The stabilization of high order methods is a difficult task in general, and we saw that addressing multiphase flows makes the problem harder.

From a general point of view, the limiters we developed are well suited with interface flows. Once our numerical scheme will be able to deal with a broader range of multiphase flows, the limiter should be also adapted. A first direction could consist in using classical stabilizations [24, 153, 57, 23] on the continuous models derived. This would mean that the stabilization would be performed *after averaging*. Still in the spirit of the numerical scheme, which consists in averaging the equations after having discretized it, we could consider stabilizing the scheme *before averaging*.

Chapter 2

Low Mach number flows

2.1 Introduction

The starting point of my interest in the low Mach number flows was driven by the stability of jets in cross flow under acoustic forcing, see [47] and also section 3.4 for the configuration without acoustic forcing. In this chapter, we will be interested in inviscid flows, described by the Euler system

$$\begin{cases} \partial_t \rho + \mathrm{div}_{\mathbf{x}}(\rho \mathbf{u}) = 0 \\ \partial_t (\rho \mathbf{u}) + \mathrm{div}_{\mathbf{x}}(\rho \mathbf{u} \otimes \mathbf{u} + p \mathrm{I}_d) = 0 \\ \partial_t (\rho E) + \mathrm{div}_{\mathbf{x}}((\rho E + p)\mathbf{u}) = 0 \end{cases}, \tag{2.1}$$

where ρ is the density, $\mathbf{u}$ the velocity, E the total specific energy, and p the pressure. The total specific energy E is

$$E = \varepsilon + \frac{|\mathbf{u}|^2}{2},$$

where ε is the specific internal energy. ε, ρ and p are linked by an equation of state $\varepsilon = \varepsilon(p, \rho)$. The wave velocities of the (2.1) in a unitary direction $\mathbf{n}$ are given by $\mathbf{u} \cdot \mathbf{n}$ (with multiplicity d in dimension d) and $\mathbf{u} \cdot \mathbf{n} \pm c$ (with multiplicity 1), where c is the sound velocity. When the Mach number $M = |\mathbf{u}| / c$ goes to 0, the material velocity wave $\mathbf{u} \cdot \mathbf{n}$ becomes small compared with the acoustic waves $\mathbf{u} \cdot \mathbf{n} \pm c$. For studying (2.1) in this limit, we denote by ρ_0 a reference density, x_0 a length scale, t_0 a time scale, and p_0 a reference pressure. Then, denoting by $c_0 = c(\rho_0, p_0)$ and $u_0 = x_0/t_0$, and adimensioning the density by ρ_0, the pressure and total energy by p_0, and the velocity by u_0, and denoting by $\tilde{a}$ the adimensioned version of the variable a, we find

$$\begin{cases} \partial_{\tilde{t}} \tilde{\rho} + \mathrm{div}_{\tilde{\mathbf{x}}}(\tilde{\rho} \tilde{\mathbf{u}}) = 0 \\ \partial_{\tilde{t}} (\tilde{\rho} \tilde{\mathbf{u}}) + \mathrm{div}_{\tilde{\mathbf{x}}}(\tilde{\rho} \tilde{\mathbf{u}} \otimes \tilde{\mathbf{u}}) + \dfrac{1}{\gamma M^2} \nabla_{\tilde{\mathbf{x}}} \tilde{p} = 0 \\ \partial_{\tilde{t}} (\tilde{\rho} \tilde{E}) + \mathrm{div}_{\tilde{\mathbf{x}}}((\tilde{\rho} \tilde{E} + \tilde{p}) \tilde{\mathbf{u}}) = 0 \end{cases} \tag{2.2}$$

with $M = u_0/c_0$ and $\gamma = \rho_0 c_0^2 / p_0$. In (2.2), the adimensioned momentum equation includes the invert of the Mach number M, and so a singular limit. For the sake of simplicity, we will rather consider the barotropic Euler system

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0, \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = 0 \end{cases} \tag{2.3}$$

where the pressure law $p = p(\rho)$ is supposed to be monotone and convex, and we denote by $c^2 = p'(\rho)$. System (2.3) has the same wave velocities as (2.1) (with one order of multiplicity less for $\mathbf{u} \cdot \mathbf{n}$ though). Given one time scale t_0, one length scale x_0 and one density scale ρ_0, the following dimensionless variables may be defined

$$\tilde{t} = \frac{t}{t_0}, \tilde{\mathbf{x}} = \frac{\mathbf{x}}{x_0}, \tilde{\rho} = \frac{\rho}{\rho_0}. \tag{2.4}$$

It is natural to scale the velocity by $u_0 = x_0/t_0$, and the pressure by $p_0 = p(\rho_0)$. If the corresponding dimensionless variables are used instead of the original ones, the following system is obtained

$$\begin{cases} \partial_{\tilde{t}}\tilde{\rho} + \mathrm{div}_{\tilde{\mathbf{x}}}(\tilde{\rho}\tilde{\mathbf{u}}) = 0 \\ \partial_{\tilde{t}}(\tilde{\rho}\tilde{\mathbf{u}}) + \mathrm{div}_{\tilde{\mathbf{x}}}(\tilde{\rho}\tilde{\mathbf{u}} \otimes \tilde{\mathbf{u}}) + \dfrac{1}{\gamma M^2} \nabla_{\tilde{\mathbf{x}}}\tilde{p} = 0 \end{cases} \tag{2.5}$$

with $\tilde{\mathbf{u}} = \mathbf{u}/u_0$, $\tilde{p} = p/p_0$, $M = u_0/c_0$ and $\gamma = \rho_0 c_0^2/p_0$. Note that (2.5) includes the same type of singular limit as (2.2) when $M \to 0$.

If acoustic phenomena are neglected, (2.5) can be replaced by an incompressible system. However, dealing with the incompressible system removes any acoustic effect, whereas we are interested in interactions between a turbulent flow and acoustic waves, so that we are interested in the behavior of (2.5) when $M \to 0$ rather than the model obtained in this limit.

From a theoretical point of view, the singular limit was studied in [78, 135]. From a numerical point of view, the approximation of (2.5) raises several problems. In the literature, the following problems have been extensively discussed

- **Accuracy problems.** It is known that in the low Mach number limit, the pressure perturbations scales as $\mathcal{O}\left(M^2\right)$, whereas most of the schemes give a spurious mode in $\mathcal{O}\left(M\right)$ [61, 127]. Several fixes have been proposed, see [61, 86, 87, 31, 128, 30, 32, 107, 19, 71]. Higher order aspects, based on the discontinuous Galerkin method, have been discussed in [12, 99, 100]. See also [109] for an example in the shallow water context, and [60] for a review on the accuracy problem of compressible solvers in the low Mach number limit.

- **Time integration strategy.** As already explained, when the Mach number is low, the ratio between the acoustic and convective wave velocities is high, which gives a CFL number for explicit schemes much lower than the CFL number based on the convection velocity. In this case, implicit-explicit strategies have been proposed [79, 105, 110, 35, 70]. Implicit integration can be simplified by considering compressible equations as perturbations of the incompressible system [97], see also [56] where the implicit time stepping is replaced by a modification of the model.

In this chapter, we propose to summarize the work published or submitted in [16, 75]. All the work will rely on the formal asymptotic development of (2.5). The different asymptotic developments will be considered in section 2.2. Then in section 2.3, we will test the classical low Mach number fixes in the case of an acoustic wave propagation, and will prove that no previously proposed low Mach number fixes is able to solve correctly acoustic waves. A new fix, able to both preserve a good accuracy at low Mach number and able also to propagate acoustic waves will be proposed and tested. Then in section 2.4, we will propose a totally different approach for computing low Mach number problems: instead of trying to fix the Riemann

solver, we will thoroughly study the nature of the spurious mode that is observed with classical Riemann solvers, and remove it from the obtained solution for recovering accuracy.

In the special case of triangular meshes, the accuracy with the Roe scheme is recovered [129, 59, 33] for finite volume schemes. We will show in section 2.5 that this result can be extended to high order discontinuous Galerkin schemes.

2.2 Asymptotic expansion of the Euler system at low Mach number

In this section, two different types of asymptotic developments of the system (2.5) are exposed. These developments made on the continuous system are important, because manipulations performed on the numerical scheme will be based on these developments.

2.2.1 One scale asymptotic expansion

In this section, the results proven theoretically in [78] are formally recalled. We are interested in the solutions of (2.5) when $M \to 0$. All the variables of the system, $\varphi \in \{\rho, \mathbf{u}\}$ are developed as expansions in power of the Mach number M:

$$\tilde{\varphi}(\tilde{\mathbf{x}}, \tilde{t}, M) = \sum_{n=0}^{N} M^n \tilde{\varphi}^{(n)}(\tilde{\mathbf{x}}, \tilde{t}) + \mathcal{O}\left(M^{N+1}\right). \tag{2.6}$$

By injecting these quantities in (2.5), the momentum equation at order M^{-2} and M^{-1} gives

$$\nabla_{\tilde{\mathbf{x}}} \tilde{p}^{(0)} = \nabla_{\tilde{\mathbf{x}}} \tilde{p}^{(1)} = 0, \tag{2.7}$$

and then, since p is a regular function of ρ,

$$\nabla_{\tilde{\mathbf{x}}} \tilde{\rho}^{(0)} = \nabla_{\tilde{\mathbf{x}}} \tilde{\rho}^{(1)} = 0.$$

This leads to

$$\tilde{\rho}^{(0)}(\tilde{\mathbf{x}}, \tilde{t}) = \tilde{\rho}^{(0)}(\tilde{t}) \quad \text{and} \quad \tilde{\rho}^{(1)}(\tilde{\mathbf{x}}, \tilde{t}) = \tilde{\rho}^{(1)}(\tilde{t}). \tag{2.8}$$

At order M^0, we get

$$\partial_{\tilde{t}} \tilde{\rho}^{(0)} + \nabla_{\tilde{\mathbf{x}}} \cdot (\tilde{\rho}^{(0)} \tilde{\mathbf{u}}^{(0)}) = 0, \tag{2.9a}$$

$$\partial_{\tilde{t}} (\tilde{\rho}^{(0)} \tilde{\mathbf{u}}^{(0)}) + \nabla_{\tilde{\mathbf{x}}} \cdot (\tilde{\rho}^{(0)} \tilde{\mathbf{u}}^{(0)} \otimes \tilde{\mathbf{u}}^{(0)}) + \nabla_{\tilde{\mathbf{x}}} \tilde{p}^{(2)} = 0. \tag{2.9b}$$

If the initial and boundary conditions are *well prepared* [78], which means that the initial and boundary conditions on $\rho^{(1)}$ are equal to 0, then $\rho^{(1)}$ is uniformly and constantly equal to 0. This can be expressed as

$$\tilde{\rho}(\tilde{\mathbf{x}}, \tilde{t}, M) = \tilde{\rho}^{(0)}(\tilde{t}) + M^2 \tilde{\rho}^{(2)}(\tilde{\mathbf{x}}, \tilde{t}) + \mathcal{O}\left(M^3\right). \tag{2.10}$$

2.2.2 Two scales asymptotic expansion

In this section, we are interested in deriving a system which allows to separate the material time scale information (moving at velocity u_0) from the acoustic time scale which matches with phenomena that move approximately at the sound velocity c_0 when the Mach number is sufficiently low. It means that we have now two reference time scales $T_{mat} = x_0/u_0 = t_0$ and $T_{ac} = x_0/c_0$ yielding two dimensionless time scales, namely

$$\tilde{t} = \frac{t}{T_{mat}} \quad \text{and} \quad \tau = \frac{t}{T_{ac}} = \frac{t}{x_0/c_0} = \frac{t}{u_0 \times T_{mat}/c_0} = \frac{t}{M \times T_{mat}} = \frac{\tilde{t}}{M}. \tag{2.11}$$

The material time $\tilde{t}$ and the acoustic time τ have a ratio of order M. This justifies to split the time dependency of the variables into two times: $\tilde{t}$ but also τ. Details on this development can be found in [95]. Then $\varphi \in \{\rho, \mathbf{u}\}$ can be written as an expansion in exponent of the Mach number M:

$$\tilde{\varphi}(\tilde{\mathbf{x}}, \tilde{t}; M) = \sum_{n=0}^{N} M^n \tilde{\varphi}^{(n)}(\tilde{\mathbf{x}}, \tilde{t}, \tau) + \mathcal{O}\left(M^{N+1}\right), \tag{2.12}$$

with $\tau = \tilde{t}/M$. Then the derivative with respect to the time is

$$\partial_{\tilde{t}}\tilde{\varphi}\left(\tilde{\mathbf{x}}, \tilde{t}; M\right) = \sum_{n=0}^{N} M^n \left(\partial_{\tilde{t}}\tilde{\varphi}^{(n)}(\tilde{\mathbf{x}}, \tilde{t}, \tau) + \frac{1}{M}\partial_{\tau}\tilde{\varphi}^{(n)}(\tilde{\mathbf{x}}, \tilde{t}, \tau)\right) + \mathcal{O}\left(M^{N+1}\right). \tag{2.13}$$

By injecting the development (2.13) in (2.5), the momentum equation at order M^{-2} gives $\nabla_{\tilde{\mathbf{x}}}\tilde{p}^{(0)} = 0$ and then, since p is a regular function of ρ,

$$\nabla_{\tilde{\mathbf{x}}}\tilde{\rho}^{(0)} = 0. \tag{2.14}$$

At order M^{-1}, we obtain

$$\partial_{\tau}\tilde{\rho}^{(0)} = 0, \tag{2.15a}$$

$$\partial_{\tau}(\tilde{\rho}^{(0)}\tilde{\mathbf{u}}^{(0)}) + \frac{1}{\gamma}\nabla_{\tilde{\mathbf{x}}}\tilde{p}^{(1)} = 0 \tag{2.15b}$$

and then, using (2.14) and (2.15a), leads to

$$\tilde{\rho}^{(0)}(\tilde{\mathbf{x}}, \tilde{t}, \tau) = \tilde{\rho}^{(0)}(\tilde{t}). \tag{2.16}$$

At order M^0, we get

$$\mathbf{d}_{\tilde{t}}\tilde{\rho}^{(0)} + \partial_{\tau}\tilde{\rho}^{(1)} + \nabla_{\tilde{\mathbf{x}}} \cdot (\tilde{\rho}^{(0)}\tilde{\mathbf{u}}^{(0)}) = 0, \tag{2.17a}$$

$$\partial_{\tilde{t}}\left(\tilde{\rho}^{(0)}\tilde{\mathbf{u}}^{(0)}\right) + \partial_{\tau}(\tilde{\rho}\tilde{\mathbf{u}})^{(1)} + \nabla_{\tilde{\mathbf{x}}} \cdot (\tilde{\rho}^{(0)}\tilde{\mathbf{u}}^{(0)} \otimes \tilde{\mathbf{u}}^{(0)}) + \frac{1}{\gamma}\nabla_{\tilde{\mathbf{x}}}\tilde{p}^{(2)} = 0. \tag{2.17b}$$

With (2.15b) and (2.17a), we obtain that $(\tilde{\rho}^{(1)}, \tilde{\rho}^{(0)}\tilde{\mathbf{u}}^{(0)})$ satisfies

$$\partial_{\tau}\tilde{\rho}^{(1)} + \nabla_{\tilde{\mathbf{x}}} \cdot (\tilde{\rho}^{(0)}\tilde{\mathbf{u}}^{(0)}) = -\mathbf{d}_{\tilde{t}}\tilde{\rho}^{(0)}, \tag{2.18a}$$

$$\partial_{\tau}(\tilde{\rho}\tilde{\mathbf{u}})^{(0)} + \frac{1}{\gamma}\nabla_{\tilde{\mathbf{x}}}\tilde{p}^{(1)} = 0. \tag{2.18b}$$

By multiplying (2.18a) by $\mathbf{d}_{\tilde{\rho}}\tilde{p}(\tilde{\rho}^{(0)})$ where $\tilde{\rho}^{(0)} = \tilde{\rho}^{(0)}(\tilde{t})$, we obtain the first order wave equation on $(\tilde{p}^{(1)}, (\tilde{\rho}\tilde{\mathbf{u}})^{(0)})$ with a source term

$$\begin{cases} \partial_\tau \tilde{p}^{(1)} + \mathbf{d}_{\tilde{\rho}}\tilde{p}(\tilde{\rho}^{(0)})\nabla_{\tilde{\mathbf{x}}} \cdot (\tilde{\rho}\tilde{\mathbf{u}})^{(0)} = -\mathbf{d}_{\tilde{t}}\tilde{p}^{(0)}, \\ \partial_\tau (\tilde{\rho}\tilde{\mathbf{u}})^{(0)} + \dfrac{1}{\gamma}\nabla_{\tilde{\mathbf{x}}}\tilde{p}^{(1)} = 0. \end{cases} \tag{2.19}$$

Instead of replacing $\tilde{\rho}^{(1)}$ by $\tilde{p}^{(1)}$ in (2.18), $\tilde{p}^{(1)}$ can be replaced by $\tilde{\rho}^{(1)}$, to get a wave system in $\left(\tilde{\rho}^{(1)}, (\tilde{\rho}\tilde{\mathbf{u}})^{(0)}\right)$

$$\begin{cases} \partial_\tau \tilde{\rho}^{(1)} + \nabla_{\tilde{\mathbf{x}}} \cdot (\tilde{\rho}\tilde{\mathbf{u}})^{(0)} = -\mathbf{d}_{\tilde{t}}\tilde{\rho}^{(0)}, \\ \partial_\tau (\tilde{\rho}\tilde{\mathbf{u}})^{(0)} + \left(\tilde{a}^{(0)}\right)^2 \nabla_{\tilde{\mathbf{x}}}\tilde{\rho}^{(1)} = 0. \end{cases} \tag{2.20}$$

2.3 Development of Accurate schemes for low Mach acoustic

2.3.1 A stationary test case

In this section, we are interested in the results obtained with the Roe scheme in the case of a scattering of a low Mach flow by a cylinder of radius r_0. This test is useful since the analytical solution for the incompressible system can be used as a reference solution. The domain Ω is an annulus $[r_0, r_1] \times [0, 2\pi[$. Here, we used $r_0 = 0.5$ and $r_1 = 5.5$. The initial data are uniform and set equal to

$$\rho_0 = 1, \quad \mathbf{u}_0 = (u_0, 0)^T, \tag{2.21}$$

with $u_0 = c_0 M_\infty$ where $c_0 = \sqrt{p'(\rho_0)}$ and M_∞ is the Mach number at infinity. The solution at infinity is uniform, equal to (2.21). We consider wall boundary condition on the internal cylinder of radius r_0 and inlet or outlet boundary condition on the external cylinder on radius r_1. On an infinite domain, the reference solution is given by

$$\begin{cases} p_{\text{ref}}(r,\theta) = p_\infty + \dfrac{p_\infty}{2}\left(2\dfrac{r_0^2}{r^2}\cos(2\theta) - \dfrac{r_0^4}{r^4}\right) M^2 + \mathcal{O}(M^3) \\ (u_x)_{\text{ref}}(r,\theta) = u_\infty - u_\infty \dfrac{r_0^2}{r^2}\cos(2\theta) + \mathcal{O}(M), \\ (u_y)_{\text{ref}}(r,\theta) = -u_\infty \dfrac{r_0^2}{r^2}\sin(2\theta) + \mathcal{O}(M). \end{cases}$$

The numerical solution obtained on a quadrangular mesh and the reference solution are shown in Figure 2.1. The numerical solution shows higher fluctuations (one order of magnitude higher) than the reference solution. In Figure 2.2, the behavior of $\tilde{\rho}_h - \tilde{\rho}_\infty$ with respect to the Mach number is drawn. The slope obtained is $\mathcal{O}(M)$ whereas (2.10) suggests a $\mathcal{O}\left(M^2\right)$ behavior.

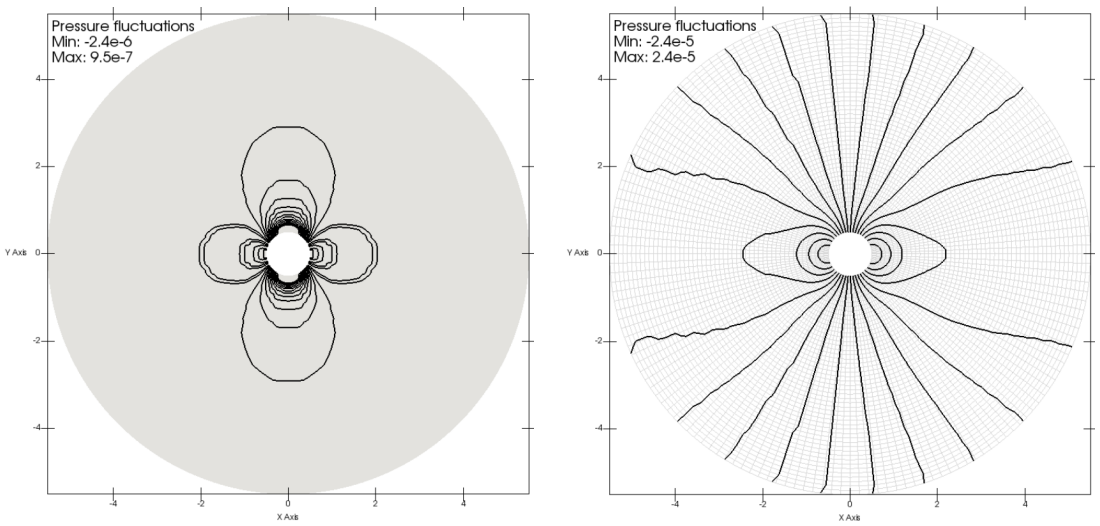


Figure 2.1: **Left:** Isolines of the reference pressure fluctuations $p_{\text{ref}} - p_\infty$ obtained at $M_\infty = 10^{-3}$. **Right :**Isolines of the pressure fluctuations $p - p_\infty$ obtained at $M_\infty = 10^{-3}$ with a quadrangular mesh for the Roe scheme.

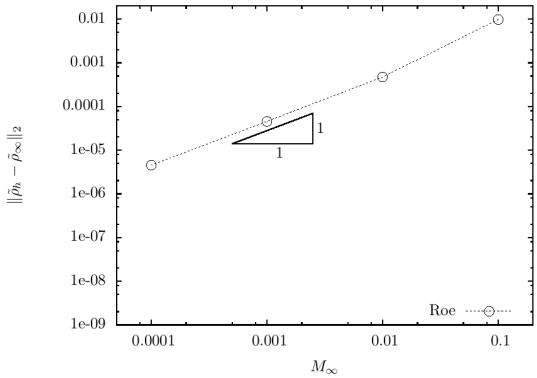


Figure 2.2: L^2 norm of the difference $\tilde{\rho}_h - \tilde{\rho}_\infty$ obtained for Mach number at infinity M_∞ varying from 10^{-1} to 10^{-4}.

2.3.2 Fixes for the stationary test case

For recovering ac accurate solution, fixes have been proposed, and we review them. The Roe solver in the subsonic case can be expressed as

$$\frac{d}{dt}\rho_i + \frac{1}{|K_i|}\sum_{\Gamma_{ij}\subset\partial K_i}|\Gamma_{ij}|\left[\frac{\rho_i\mathbf{u}_i+\rho_j\mathbf{u}_j}{2}\cdot\mathbf{n}_{ij} + \frac{\rho_{ij}}{2a_{ij}}(\mathbf{u}_{ij}\cdot\mathbf{n}_{ij})(\mathbf{u}_i-\mathbf{u}_j)\cdot\mathbf{n}_{ij} + \frac{a_{ij}}{2}(\rho_i-\rho_j)\right] = 0, \tag{2.22a}$$

$$\begin{aligned}\frac{d}{dt}(\rho_i\mathbf{u}_i) + \frac{1}{|K_i|}\sum_{\Gamma_{ij}\subset\partial K_i}|\Gamma_{ij}|\Bigg[&\frac{\rho_i(\mathbf{u}_i\cdot\mathbf{n}_{ij})\mathbf{u}_i+\rho_j(\mathbf{u}_j\cdot\mathbf{n}_{ij})\mathbf{u}_j}{2} \\ &+ \frac{a_{ij}}{2}(\rho_i-\rho_j)\left[\mathbf{u}_{ij}+(\mathbf{u}_{ij}\cdot\mathbf{n}_{ij})\mathbf{n}_{ij}\right] \\ &+ \frac{\rho_{ij}}{2}|\mathbf{u}_{ij}\cdot\mathbf{n}_{ij}|(\mathbf{u}_i^{\perp}(\mathbf{n}_{ij})-\mathbf{u}_j^{\perp}(\mathbf{n}_{ij})) \\ &+ \frac{\rho_{ij}(\mathbf{u}_{ij}\cdot\mathbf{n}_{ij})}{2a_{ij}}\left[(\mathbf{u}_i-\mathbf{u}_j)\cdot\mathbf{n}_{ij}\right]\mathbf{u}_{ij} \\ &+ \left[\frac{p_i+p_j}{2}+\frac{\rho_{ij}a_{ij}}{2}(\mathbf{u}_i-\mathbf{u}_j)\cdot\mathbf{n}_{ij}\right]\mathbf{n}_{ij}\Bigg] = 0\end{aligned} \tag{2.22b}$$

where $p_k = p(\rho_k)$ and the states $(\cdot)_{ij}$ are defined by

$$\rho_{ij} = \sqrt{\rho_i\rho_j}, \quad \mathbf{u}_{ij} = \frac{\sqrt{\rho_i}\mathbf{u}_i+\sqrt{\rho_j}\mathbf{u}_j}{\sqrt{\rho_i}+\sqrt{\rho_j}} \quad \text{and} \quad a_{ij}^2 = \begin{cases}\dfrac{\Delta p}{\Delta\rho}, & \text{if } \Delta\rho\neq 0,\\ p'(\rho_i), & \text{otherwise}\end{cases} \tag{2.23}$$

2.3.2.1 Analysis of the problem for the Roe scheme

Performing at the discrete case the adimensioning and the same asymptotic analysis as done in subsection 2.2.1 leads to uniformity of $\tilde{p}^{(0)}$ and $\tilde{\rho}^{(0)}$, and to the following system coupling $\tilde{p}^{(1)}$ and $\tilde{\mathbf{u}}^{(0)}$

$$\begin{cases}\displaystyle\sum_{\Gamma_{ij}\subset\partial K_i}|\tilde{\Gamma}_{ij}|\left[-\tilde{\rho}^{(0)}\tilde{a}^{(0)}\left(\tilde{\mathbf{u}}_i^{(0)}-\tilde{\mathbf{u}}_j^{(0)}\right)\cdot\mathbf{n}_{ij}+\left(\tilde{p}_i^{(1)}-\tilde{p}_j^{(1)}\right)\right] = 0\\ \displaystyle\sum_{\Gamma_{ij}\subset\partial K_i}|\tilde{\Gamma}_{ij}|\left[-\tilde{\rho}^{(0)}\tilde{a}^{(0)}(\tilde{\mathbf{u}}_i^{(0)}-\tilde{\mathbf{u}}_j^{(0)})\cdot\mathbf{n}_{ij}+(\tilde{p}_i^{(1)}-\tilde{p}_j^{(1)})\right]\mathbf{n}_{ij} = 0\end{cases} \tag{2.24}$$

Then in the general case, as explained in [61, 60], the system (2.24) admits non-zero solutions implying that the discrete solution can contain a non constant pressure fluctuation $p^{(1)}$, which is the spurious mode observed in Figure 2.1 and leading to the wrong order in $\mathcal{O}(M)$ in Figure 2.2.

2.3.2.2 Fixes for the Roe scheme

Several fixes have been proposed over the last two decades. We recall three of these

- The all-Mach correction of Dellacherie and al. [32] consists in replacing the term

$$\Delta(\mathbf{u}\cdot\mathbf{n})$$

in the momentum equation (2.22b) of the Roe scheme by the term

$$\min(M_{ij}, 1)\Delta\left(\mathbf{u}\cdot\mathbf{n}\right)$$

where M_{ij} is a term of the order of the Mach number computed at each edge of the mesh. In the future numerical tests, we will take $M_{ij} = \max(M_i, M_j) = \max(\|\mathbf{u}_i\|/c_i, \|\mathbf{u}_j\|/c_j\|)$.

- The low Mach correction of [128] consists in replacing *every* occurrence of the term

$$\Delta\left(\mathbf{u}\cdot\mathbf{n}\right)$$

in the Roe scheme by

$$\min(M_{ij}, 1)\Delta\left(\mathbf{u}\cdot\mathbf{n}\right),$$

where M_{ij} is of the order of the Mach number computed at each edge of the mesh.

- The Roe flux, which can be shortly expressed as a centered part and a diffusive part as

$$\Phi_{ij}^{\text{Roe}} = \frac{\mathbf{f}(\mathcal{U}_i)+\mathbf{f}(\mathcal{U}_j)}{2}\cdot\mathbf{n}_{ij} - \frac{|A_{\mathbf{n}_{ij}}(\mathcal{U}_i,\mathcal{U}_j)|}{2}\cdot(\mathcal{U}_j-\mathcal{U}_i). \tag{2.25}$$

is modified by preconditioning the diffusion matrix as follows

$$\Phi_{ij}^{\text{Roe-Turkel}} = \frac{\mathbf{f}(\mathcal{U}_i)+\mathbf{f}(\mathcal{U}_j)}{2}\cdot\mathbf{n}_{ij} - P(\mathcal{U}_{ij})^{-1}\frac{|P(\mathcal{U}_{ij})A_{\mathbf{n}_{ij}}(\mathcal{U}_{ij})|}{2}\cdot(\mathcal{U}_j-\mathcal{U}_i) \tag{2.26}$$

where P is a preconditioning matrix proposed by Turkel [142, 61]. In primitive variables $\mathcal{V}$, it is given by

$$P(\mathcal{V}) = \begin{pmatrix} \beta^2 & 0 \\ 0 & \mathcal{I}_3 \end{pmatrix}$$

where β is a parameter of the order of the Mach number. For the conservative variables $\mathcal{U}$, the corresponding form is

$$P(\mathcal{U}) = \frac{\partial\mathcal{U}}{\partial\mathcal{V}}P(\mathcal{V})\frac{\partial\mathcal{V}}{\partial\mathcal{U}}.$$

Results with these fixes are shown in Figure 2.3 for the isovalues, and in Figure 2.4 for the convergence of the density fluctuations when $M \to 0$. A good approximation with the three corrected Roe schemes is obtained, and the behavior of the density in $\mathcal{O}\left(M^2\right)$ is recovered with the fixes.

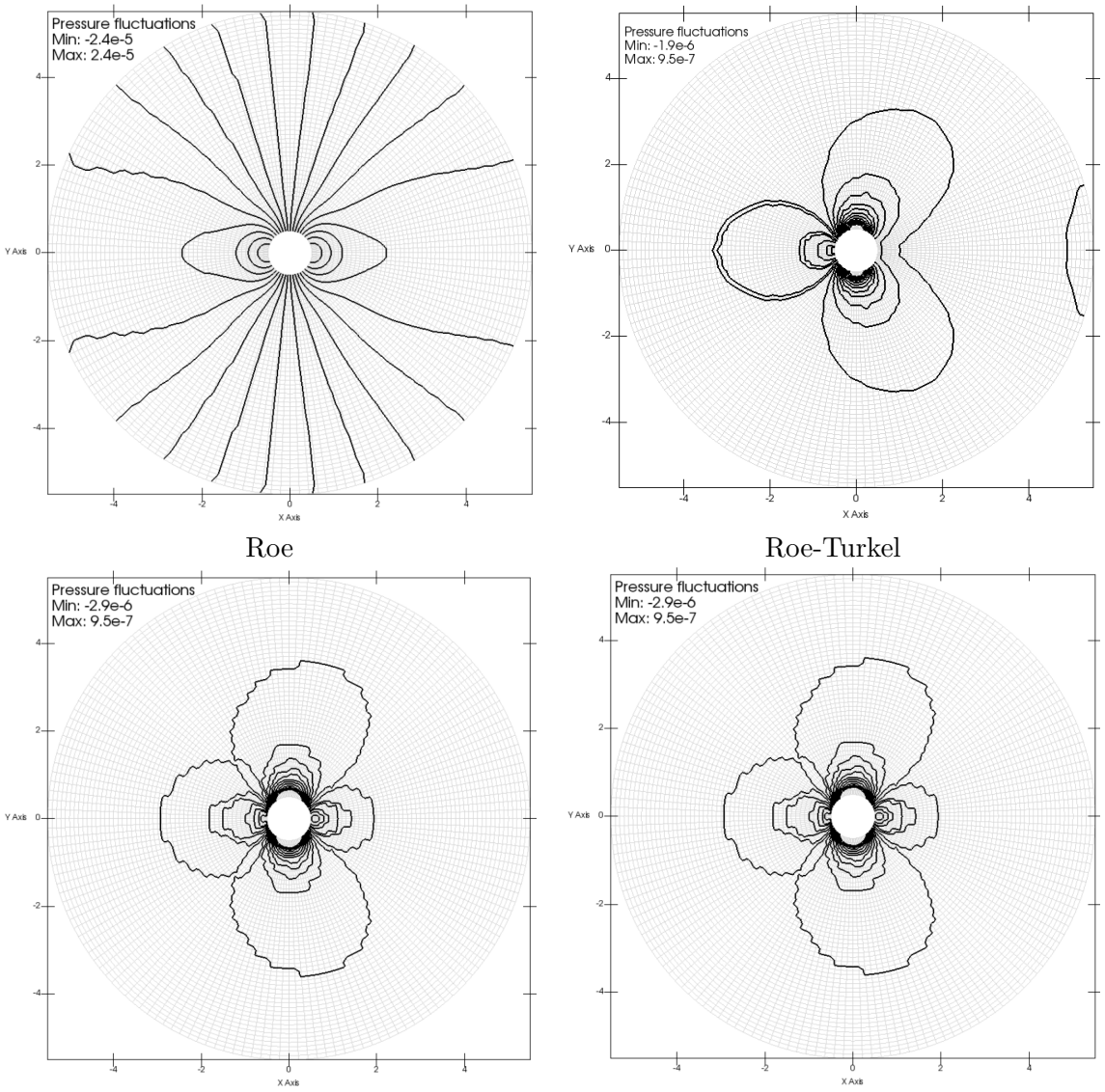
2.3.3 An acoustic test case

In this section, we address the problem of acoustic waves propagation in a low Mach number flow. We consider the following profile for the density

$$\rho_0(x) = \rho_\infty\left(1 + M_\infty\sin(2\pi f x)\right) \tag{2.27}$$

and since $u - \frac{2c}{\gamma-1}$ is a Riemann invariant of the wave associated to $x/t = u + c$, the initial velocity is taken as

$$u_0(x) = u_\infty + \frac{2}{\gamma-1}\left(c\left(\rho_0(x)\right) - c_\infty\right) \tag{2.28}$$



Roe | Roe-Turkel

Roe with Dellacherie and al. fix | Roe with Rieper fix

Figure 2.3: Isolines of the pressure fluctuations $p - p_\infty$ obtained at $M_\infty = 10^{-3}$ with a quadrangular mesh for the Roe scheme and different classical low Mach fixes. 20 isolines were drawn, between -2.4×10^{-5} and 2.4×10^{-5} for the original Roe scheme, and 20 isolines were drawn, between -1.5×10^{-6} and 10^{-6} for the other results.

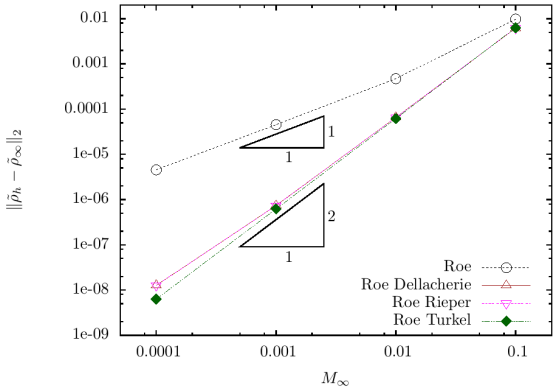


Figure 2.4: L^2 norm of the difference $\tilde{\rho}_h - \tilde{\rho}_\infty$ obtained for Mach number at infinity M_∞ varying from 10^{-1} to 10^{-4} for the Roe scheme and different classical low Mach fixes. With all low Mach fixes, a convergence rate of 2 is obtained as the reference solution.

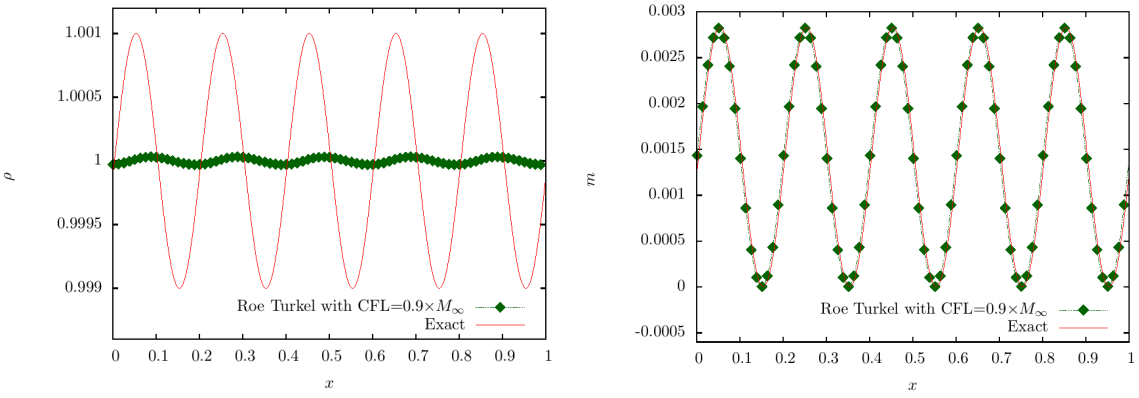


Figure 2.5: Density ρ (left) and momentum m (right) obtained at time $t = 0.003$ s on the acoustic wave test case with $N = 400$ regular cells and with a first order Roe-Turkel scheme.

where $u_\infty = M_\infty \times c_\infty$. We choose $\rho_\infty = 1$, $M_\infty = 10^{-3}$ and $f = 5$. An exact solution at a given time can be computed with the method of characteristics.

The different numerical schemes are tested with a regular mesh $\Delta x = 1/N$ and with a time explicit solver where the time step $(\Delta t)_n$ satisfies

$$(\Delta t)_n = \text{CFL} \times \frac{\Delta x}{\max_i \left(|u_i^n| + c_i^n\right)}.$$

The CFL number will be specified in each case. Results obtained with the Roe-Turkel fix are shown in Figure 2.5. For a time of the order of the Mach number, the density obtained with the Roe-Turkel scheme is strongly damped. For studying the damping of the acoustic waves, the dimensionless amplitude is drawn for a Mach number equal to 1, 10^{-1}, 10^{-2}, 10^{-3} for the Roe and Roe-Turkel schemes, for $t \in [0, 0.003]$ in Figure 2.6. With the Roe-Turkel scheme, for both the explicit and implicit time stepping, the lower the Mach number is, the faster the amplitude is going to 0. With the Roe scheme, the dimensionless amplitude stays close to 1,

and is nearly Mach independent.

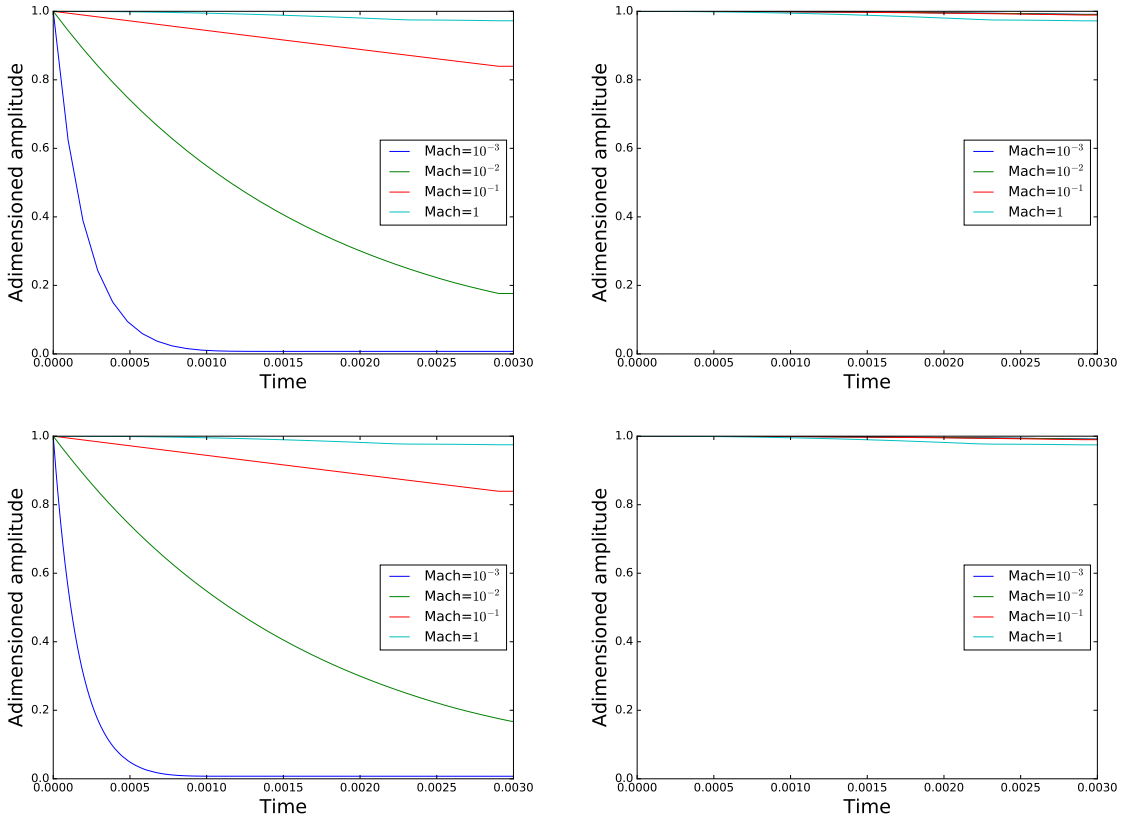


Figure 2.6: Damping of the density component on the acoustic test case with implicit time stepping (top) and explicit time stepping (bottom), for the Roe-Turkel scheme (left) and the Roe scheme (right), for the Mach number M_∞ equal to 1, 10^{-1}, 10^{-2}, 10^{-3}.

Results obtained with the Dellacherie and al. and Rieper fixes are shown in Figure 2.7, and compared with the results obtained with the Roe scheme. Numerical results show that the numerical scheme with the different fixes is stable only with a degraded CFL number, approximately half of the CFL number for the original Roe scheme. Last, the numerical error is computed on the density and on the momentum, still with these numerical fluxes, for a first order finite volume computation, and also with a second order approximation. The second order is achieved with a discontinuous Galerkin with piecewise linear approximation in space, and a SSP integration in time [25]. Results are shown in Figure 2.8. As expected, the Roe-Turkel finite volume approximation does not converge. Convergence is observed with the finite volume approximation for the other fixes. But only a first order of accuracy is observed for the momentum with the second order scheme.

To summarize this section, none of the fixes tested gives satisfactory results on the acoustic wave test case: the Roe-Turkel fix induces a strong damping of one of the component of the wave, whereas with the other fixes, the acoustic wave is propagated, but with a degraded CFL number, and with a suboptimal convergence order for the discontinuous Galerkin method. It is important to note that the original Roe scheme propagates correctly acoustic waves, and with the right order of accuracy, and so that low Mach fixes *destroy* this property.

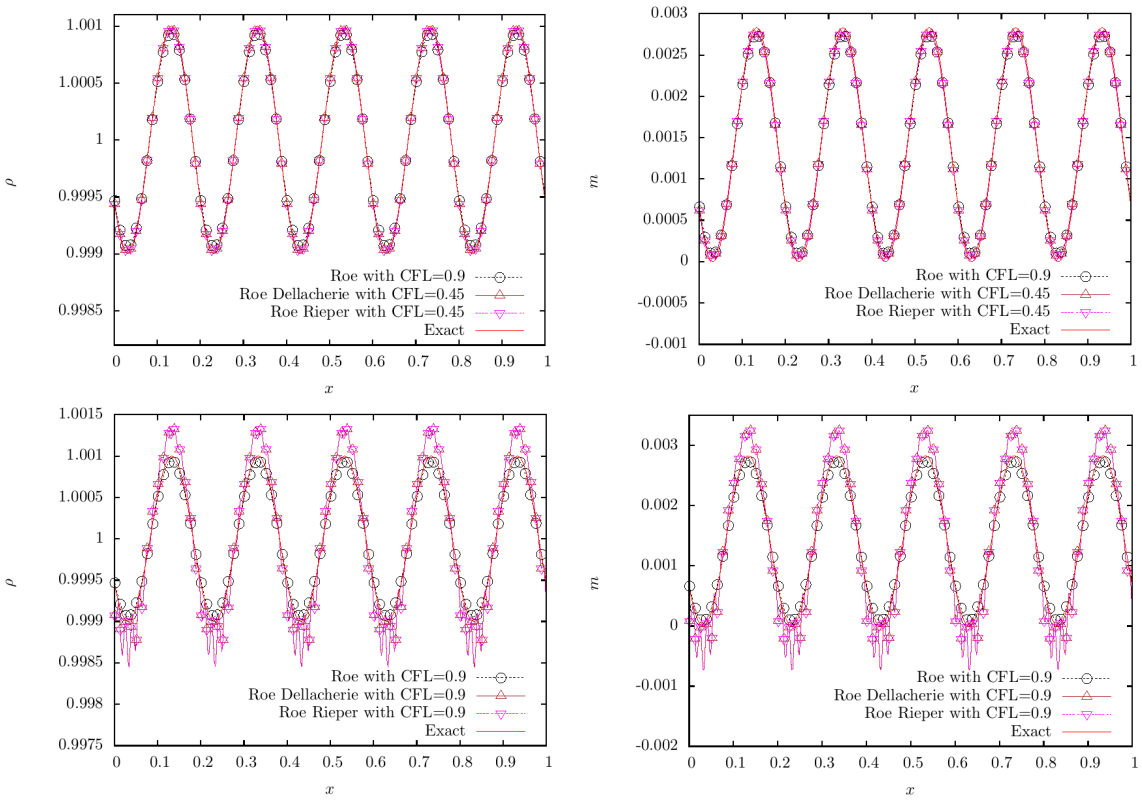


Figure 2.7: Density ρ (left) and momentum m (right) obtained at time $t = 0.3408$ s on the acoustic wave test case with $N = 400$ regular cells and with a first order scheme. The Roe scheme corrected with Dellacherie and al. and Rieper fixes is not stable if we consider a CFL number of 0.9 (bottom figures). Indeed, it seems to be stable for a CFL condition twice smaller than the Roe scheme (top figures).

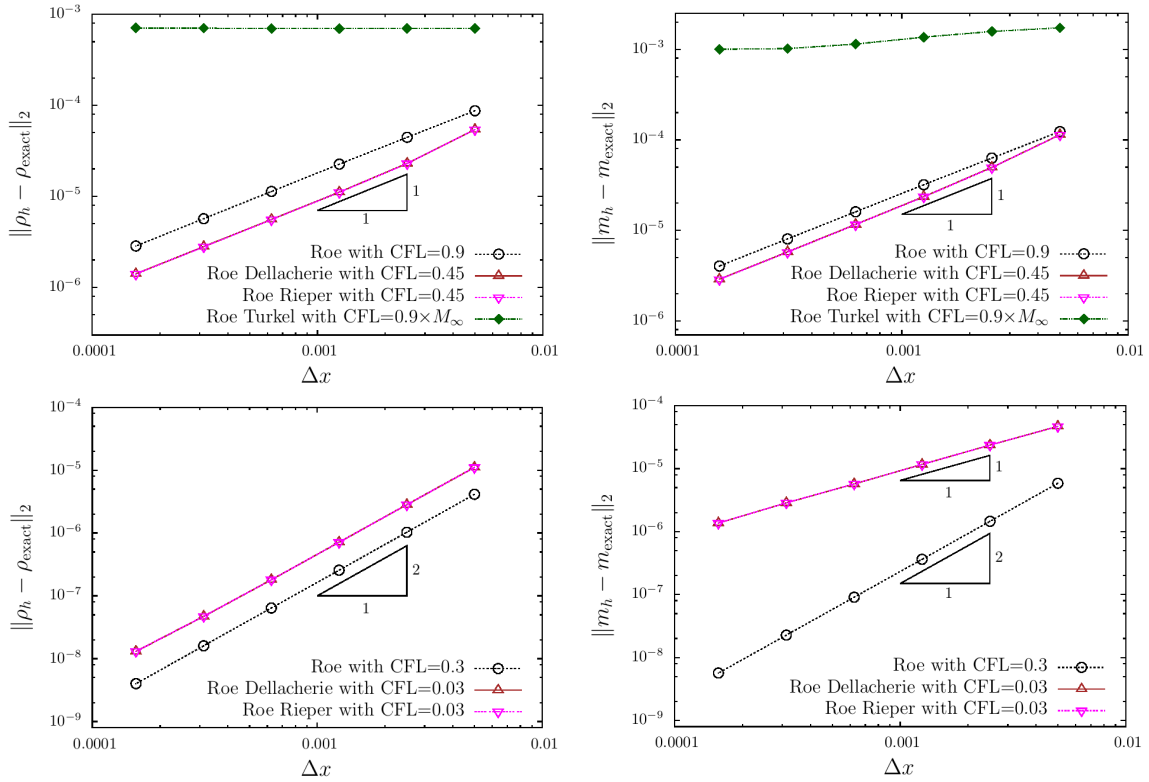


Figure 2.8: Order obtained on the acoustic wave test case at time $t = 0.3408$ s with the Roe scheme, and different fixes. The coarser and finer meshes contain $N = 200$ and $N = 6400$ regular cells respectively, and $\Delta x = 1/N$. On the two top figures, the error obtained on the density (left) and on the momentum (right) are shown with a first order finite volume scheme. On the bottom figures, the error obtained on the density (left) and on the momentum (right) are shown with a second order discontinuous Galerkin method.

2.3.4 Fixes for the acoustic waves propagation

2.3.4.1 Analysis of the problem raised by acoustic propagation

For explaining the results of the previous section, a double time scale asymptotic expansion is performed on the numerical schemes as in subsection 2.2.2.

In the case of the Roe-Turkel case, it is possible to prove

Proposition 8 (Proposition 2.1 of [16]). *If the scales in $\mathcal{O}\left(M^k\right)$ are supposed to be separated, and if we suppose that the boundary conditions are either wall or periodic, then the Roe-Turkel solver is such that*

$$\forall i, \quad \tilde{\rho}_i^{(0)} = \tilde{\rho}^{(0)} \qquad \text{and} \qquad \forall i, \quad \tilde{\rho}_i^{(1)} = \tilde{\rho}^{(1)}.$$

which explains the damping of the acoustic waves with this fix. As far as the Roe scheme and its other fixes are concerned, it is possible to prove that $\tilde{\rho}^{(0)}$ is uniform and depends neither on $\tilde{t}$ nor on τ. This is similar to what is found in the continuous case in (2.16). We can then prove that

Proposition 9 (Proposition 2.3 of [16]). *$(\tilde{p}^{(1)}/\gamma, (\tilde{\rho}\tilde{\mathbf{u}})^{(0)})$ satisfies the semi-discrete (discrete in space and continuous in time) wave equation*

$$\left\{\begin{array}{l} \partial_\tau \dfrac{\tilde{p}_i^{(1)}}{\gamma} + \dfrac{1}{|\tilde{K}_i|}\displaystyle\sum_{\Gamma_{ij}\subset\partial K_i} |\tilde{\Gamma}_{ij}| \left[(\tilde{a}^{(0)})^2 \frac{(\tilde{\rho}\tilde{\mathbf{u}})_i^{(0)} + (\tilde{\rho}\tilde{\mathbf{u}})_j^{(0)}}{2} \cdot \mathbf{n}_{ij} + \frac{\tilde{a}^{(0)}}{2}\left(\frac{\tilde{p}_i^{(1)}}{\gamma} - \frac{\tilde{p}_j^{(1)}}{\gamma}\right)\right] \\ \qquad\qquad = -\dfrac{1}{\gamma}\mathbf{d}_{\tilde{t}}\tilde{p}^{(0)}, \\ \partial_\tau (\tilde{\rho}\tilde{\mathbf{u}})_i^{(0)} + \dfrac{1}{|\tilde{K}_i|}\displaystyle\sum_{\Gamma_{ij}\subset\partial K_i} |\tilde{\Gamma}_{ij}| \left[\frac{\tilde{p}_i^{(1)} + \tilde{p}_j^{(1)}}{2\gamma}\mathbf{n}_{ij} \right. \\ \qquad\qquad \left. + \dfrac{\delta\tilde{a}^{(0)}}{2}\left[\left((\tilde{\rho}\tilde{\mathbf{u}})_i^{(0)} - (\tilde{\rho}\tilde{\mathbf{u}})_j^{(0)}\right)\cdot\mathbf{n}_{ij}\right]\mathbf{n}_{ij}\right] = 0 \end{array}\right. \tag{2.29}$$

where $\delta = 1$ for the Roe scheme, $\delta = 0$ for the Dellacherie et al. and Rieper fixes and $\tilde{a}^{(0)} = \sqrt{\mathbf{d}_{\tilde{\rho}}\tilde{p}(\tilde{\rho}^{(0)})/\gamma}$.

This last proposition explains why a lower order of accuracy is found with the low Mach number fixes: in the case of the Roe flux ($\delta = 1$), the diffusion is not degenerated, whereas with a fix ($\delta = 0$) the numerical diffusion for the momentum equation is degenerated.

2.3.4.2 Proposition of a fix for the acoustic propagation

In [16], a general stabilization was searched in the following form

$$\left\{\begin{array}{l} \partial_\tau \dfrac{\tilde{p}_i^{(1)}}{\gamma} + \dfrac{1}{|\tilde{K}_i|}\displaystyle\sum_{\Gamma_{ij}\subset\partial K_i} |\tilde{\Gamma}_{ij}| \left[(\tilde{a}^{(0)})^2 \frac{(\tilde{\rho}\tilde{\mathbf{u}})_i^{(0)} + (\tilde{\rho}\tilde{\mathbf{u}})_j^{(0)}}{2} \cdot \mathbf{n}_{ij} + C_{11}\left(\frac{\tilde{p}_i^{(1)}}{\gamma} - \frac{\tilde{p}_j^{(1)}}{\gamma}\right)\right. \\ \qquad\qquad \left. + \mathbf{C}_{12,d}\cdot\left[\left((\tilde{\rho}\tilde{\mathbf{u}})_i^{(0)} - (\tilde{\rho}\tilde{\mathbf{u}})_j^{(0)}\right)\right]\right] = -\dfrac{1}{\gamma}\mathbf{d}_{\tilde{t}}\tilde{p}^{(0)}, \\ \partial_\tau (\tilde{\rho}\tilde{\mathbf{u}})_i^{(0)} + \dfrac{1}{|\tilde{K}_i|}\displaystyle\sum_{\Gamma_{ij}\subset\partial K_i} |\tilde{\Gamma}_{ij}| \left[\frac{\tilde{p}_i^{(1)} + \tilde{p}_j^{(1)}}{2\gamma}\mathbf{n}_{ij} + \mathbf{C}_{21,d}\left(\frac{\tilde{p}_i^{(1)}}{\gamma} - \frac{\tilde{p}_j^{(1)}}{\gamma}\right)\right. \\ \qquad\qquad \left. + C_{22}\left[\left((\tilde{\rho}\tilde{\mathbf{u}})_i^{(0)} - (\tilde{\rho}\tilde{\mathbf{u}})_j^{(0)}\right)\cdot\mathbf{n}_{ij}\right]\mathbf{n}_{ij}\right] = 0 \end{array}\right. \tag{2.30}$$

The general stabilization of (2.30) was studied in [16], which shows that for ensuring the dissipation of the acoustic energy, we must have

$$\begin{aligned} &C_{11} \geqslant 0, \\ &\mathbf{C}_{12,d} + \left(\tilde{a}^{(0)}\right)^2 \mathbf{C}_{21,d} = 0, \\ &C_{22} \geqslant 0. \end{aligned}$$

All the previous schemes which are accurate at low Mach number in the steady case match with $C_{22} = 0$, and so we decide to set $C_{22} = 0$. The last coefficients are obtained by optimizing the CFL number, and lead to, in one dimension,

$$C_{11} = \left(\tilde{a}^{(0)}\right)^2 \qquad \text{and} \qquad \mathbf{C}_{21,d} = \pm\frac{1}{2}.$$

Note that the stabilization is *not Galilean invariant*; the two possible orientations in one dimension give two different stabilizations, which are obtained by changing the sign of $\mathbf{C}_{21,d}$. In dimension d, the following coefficients can be used

$$C_{11} = a_\star, \quad C_{22} = 0, \quad \mathbf{C}_{12,d} = \pm\frac{\left(\tilde{a}^{(0)}\right)^2}{2\sqrt{d}}\mathbf{1}_d \quad \text{and} \quad \mathbf{C}_{21,d} = \mp\frac{1}{2\sqrt{d}}\mathbf{1}_d \quad \text{where} \quad \mathbf{1}_d = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{2.31}$$

This leads to the following fix for the Roe scheme

$$\begin{aligned} \Phi_{ij}^{\text{New}} = \Phi_{ij}^{\text{Roe}} - (1-\theta_{ij})\frac{\rho_{ij}a_{ij}}{2}\begin{pmatrix} 0 \\ \left[(\mathbf{u}_i - \mathbf{u}_j)\cdot\mathbf{n}_{ij}\right]\mathbf{n}_{ij} \end{pmatrix} \\ + (1-\theta_{ij})\begin{pmatrix} \frac{1}{2}a_{ij}(\rho_i - \rho_j) \pm \frac{1}{2\sqrt{d}}\mathbf{1}_d\cdot(\rho_i\mathbf{u}_i - \rho_j\mathbf{u}_j) \\ \mp\frac{a_{ij}^2}{2\sqrt{d}}(\rho_i - \rho_j)\mathbf{1}_d \end{pmatrix} \end{aligned} \tag{2.32}$$

with $\theta_{ij} = \min(M_{ij}, 1)$ and $M_{ij} = \max(M_i, M_j) = \max(\|\mathbf{u}_i\|/a_i, \|\mathbf{u}_j\|/a_j)$.

2.3.4.3 Numerical results

Convergence order obtained on the acoustic wave test case are shown in Figure 2.9, and prove that the optimal order is recovered with respect to the other low Mach fixes. Results obtained on the stationary test case are shown in Figure 2.10 and show a similar behavior as the Roe scheme with low Mach number fixes. We thus achieved to build a fix that is able to accurately solve not only stationary low Mach number flows, but also propagation of acoustic waves in low Mach number flows.

2.4 A filtering method for low Mach number flows

2.4.1 Link between the ω-limit of the wave system and the spurious mode at low Mach number

In this section, a link between the ω-limit of the wave system and the spurious mode observed in Figure 2.1 is made. It relies on how the computation for the steady state test case is

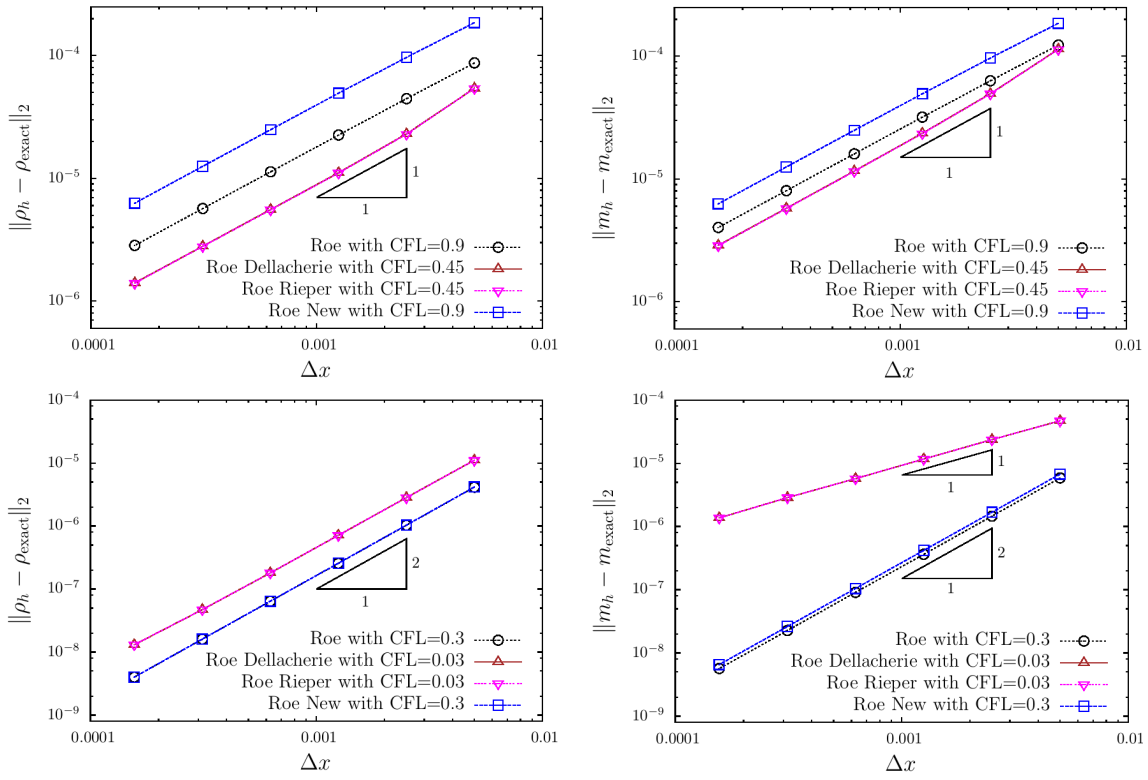


Figure 2.9: Order obtained on the density (left) and on the momentum (right) on acoustic wave test case at time $t = 0.3408$ s with the Roe scheme and different fixes with a first (top) and a second order (bottom) approximations.

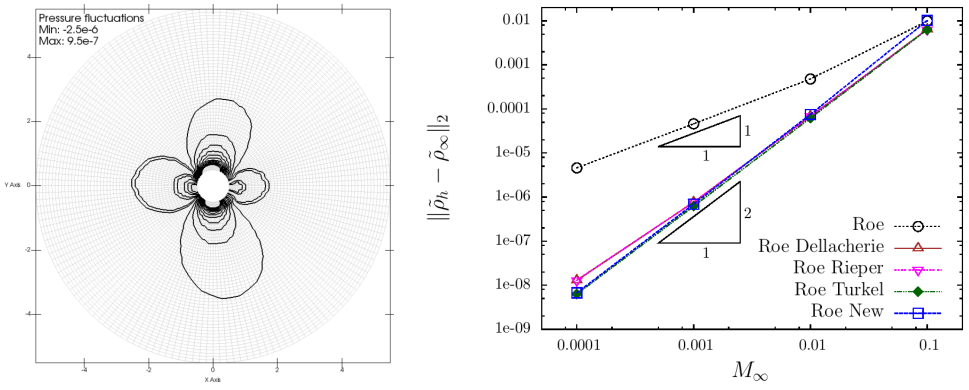


Figure 2.10: **Left:** Isolines of the pressure fluctuations $p - p_\infty$ obtained at $M_\infty = 10^{-3}$ on the stationary test case with the new solver. 20 isolines were drawn, between -1.5×10^{-6} and 10^{-6} for the other results. Results should be compared with the ones of Figure 2.3 and Figure 2.1. **Right:** L^2 norm of the difference $\tilde{\rho}_h - \tilde{\rho}_\infty$ obtained for Mach number at infinity M_∞ varying from 10^{-1} to 10^{-4} with the new scheme, compared with the results obtained with the Roe scheme and different fixes.

led: as Euler equations are nonlinear, a time marching method is used until the residuals are sufficiently small for ensuring that a steady state is reached. When the solution is not accurate at low Mach number, the density is $\mathcal{O}(M)$ instead of $\mathcal{O}(M^2)$. If we consider the steady state of the two time scales asymptotic expansion of subsection 2.2.2, this means that the long time behavior (also called the ω-limit problem) of the wave system (2.20) is such that the density $\tilde{\rho}^{(1)}$ is not uniform. This means that, based on the consistency of the asymptotic numerical scheme with the wave system of Proposition 9, we are naturally interested in the discrete ω-limit set of (2.29). We make the following two remarks on this problem

- The ω-limit problem may be strongly different from the stationary problem. For example, if the ω-limit problem of (2.20) is considered, and if the solution is independent of time, then we find a uniform density, and a divergence free momentum. But by taking the curl of the momentum equation, we find additionally that the momentum is harmonic provided the curl of the momentum is initially zero. If the stationary solutions of (2.20) are considered, we find a uniform density, and a divergence free momentum, which is a much larger space of solutions than the solutions of the ω-limit set.

- When a continuous hyperbolic system is considered, the ω-limit set may be something very complicated. For example, considering the wave system in 1d with wall boundary conditions, the initial condition will travel and reflect inside the domain forever, leading to a large ω-limit set.

2.4.2 ω-limit of the finite volume discretization of a hyperbolic system

In [75], the ω-limit set of a diffusive discretization of a general hyperbolic system was considered: the general system is

$$\partial_\tau \mathbf{U} + \sum_{k=1}^{d} A_k \partial_{\mathbf{x}_k} \mathbf{U} = 0, \tag{2.33}$$

and is supposed to be symmetrizable. It is discretized by

$$\begin{aligned} \partial_t \mathbf{U}_i + \frac{1}{|K_i|} \sum_{j \in \mathcal{V}_{\text{int}}(i)} |\Gamma_{ij}| F(\mathbf{U}_i, \mathbf{U}_j, \mathbf{n}_{ij}) \\ + \frac{1}{|K_i|} \sum_{j \in \mathcal{V}_{\text{wall}}(i)} |\Gamma_{ij}| F^{\text{wall}}(\mathbf{U}_i, \mathbf{n}_{ij}) \\ + \frac{1}{|K_i|} \sum_{j \in \mathcal{V}_{\text{SW}}(i)} |\Gamma_{ij}| F^{\text{SW}}(\mathbf{U}_i, \mathbf{U}_b, \mathbf{n}_{ij}) = 0 \end{aligned} \tag{2.34}$$

where $\mathcal{V}_{\text{wall}}$ is the set of boundary sides on which a wall boundary condition is imposed, whereas $\mathcal{V}_{\text{SW}}$ is the set of boundary sides on which inlet/outlet boundary conditions are imposed with a Steger-Warming approach. On interior sides, the following numerical flux is used

$$F(\mathbf{U}_i, \mathbf{U}_j, \mathbf{n}_{ij}) = A(\mathbf{n}_{ij}) \{\!\{ \mathbf{U} \}\!\}_{ij} - D(\mathbf{n}_{ij}) (\mathbf{U}_j - \mathbf{U}_i).$$

Given a boundary state $\mathbf{U}_b$, and $\mathbf{n}$ the outgoing normal, the Steger-Warming boundary flux is defined as

$$F^{\text{SW}}(\mathbf{U}_i, \mathbf{U}_b, \mathbf{n}) = A^+(\mathbf{n})\mathbf{U}_i + A^-(\mathbf{n})\mathbf{U}_b.$$

Last, the wall boundary conditions are defined as

$$F^{\text{wall}}(\mathbf{U}_i, \mathbf{n}) = A(\mathbf{n})\frac{\mathbf{U}_i + \breve{\mathbf{U}}_i}{2} - \frac{1}{2}\left(A^+(\mathbf{n}) - A^-(\mathbf{n})\right)\left(\breve{\mathbf{U}}_i - \mathbf{U}_i\right)$$

where $\breve{\mathbf{U}}_i$ is obtained from a transformation of $\mathbf{U}_i$: $\breve{\mathbf{U}}_i = (\mathrm{I}_n - P_{\text{wall}}(\mathbf{n}))\,\mathbf{U}_i$ where $P_{\text{wall}}(\mathbf{n})$ is nonnegative.

Due to the inlet/outlet boundary conditions, the existence and structure of the ω-limit set cannot be studied with an energy method, because the energy has no defined monotony across time. Instead of this, the ω-limit set is studied by a purely algebraic method, by proving that (2.34), seen as a coupled system of ODE

$$\boldsymbol{\mathcal{U}}' = \boldsymbol{\mathcal{B}} - \mathcal{A}\boldsymbol{\mathcal{U}} \tag{2.35}$$

follows some necessary conditions on $\mathcal{A}$, summarized in

Hypothesis 1. *Suppose that $\mathcal{A}$ follows the following conditions*

1. $\operatorname{rank}\mathcal{A} \cap \ker\mathcal{A} = \{0\}$
2. *All the eigenvalues λ of $\mathcal{A}$ are such that* $\mathrm{Re}(\lambda) \geqslant 0$.
3. 0 *is the only eigenvalue of $\mathcal{A}$ such that* $\mathrm{Re}(\lambda) = 0$

We can then prove that

Proposition 10 (Proposition 4 of [75])**.** *Suppose that Hypothesis 1 hold. Then*

- *A given solution of (2.35) is bounded in positive time, if and only if $\boldsymbol{\mathcal{B}} \in \operatorname{rank}\mathcal{A}$*
- *Denoting by $\boldsymbol{\mathcal{U}}^\infty$ the unique vector of $\operatorname{rank}\mathcal{A}$ such that $\boldsymbol{\mathcal{B}} = \mathcal{A}\boldsymbol{\mathcal{U}}^\infty$, (2.35) with initial data $\boldsymbol{\mathcal{U}}_0$ converges towards $\boldsymbol{\mathcal{U}}^\infty + \mathbb{P}(\boldsymbol{\mathcal{U}}_0)$, where $\mathbb{P}$ is the projection on $\ker\mathcal{A}$.*
- *The ω-limit set is an affine space, equal to $\boldsymbol{\mathcal{U}}^\infty + \ker\mathcal{A}$.*

Then it can be proven that under mild hypothesis, 1, 2 and 3 hold for (2.34) (see propositions 5 and 6 of [75]).

2.4.3 Filtering method for low Mach number flows

The classical numerical schemes for the stabilization of the wave system, like the Roe scheme or the Lax-Friedrich scheme, follow the hypothesis developed in [75] for ensuring a structure of the ω-limit set as in Proposition 10 for the wave system. The stabilizations of the wave system obtained with the low Mach fixes (except for the Roe-Turkel one) follow also these hypothesis. This is depicted in Figure 2.11 on a test case of the scattering of the linear wave system by a cylinder.

We can now check the pressure of the ω-limit. The stabilizations of the wave system can be divided into two families (see Figures 2,3,4,5,6 of [75])

- The ones that match with an accurate scheme at low Mach number. For these schemes, the pressure of the ω-limit observed is uniform and equal to 0. Moreover, the velocity is close of the exact solution.

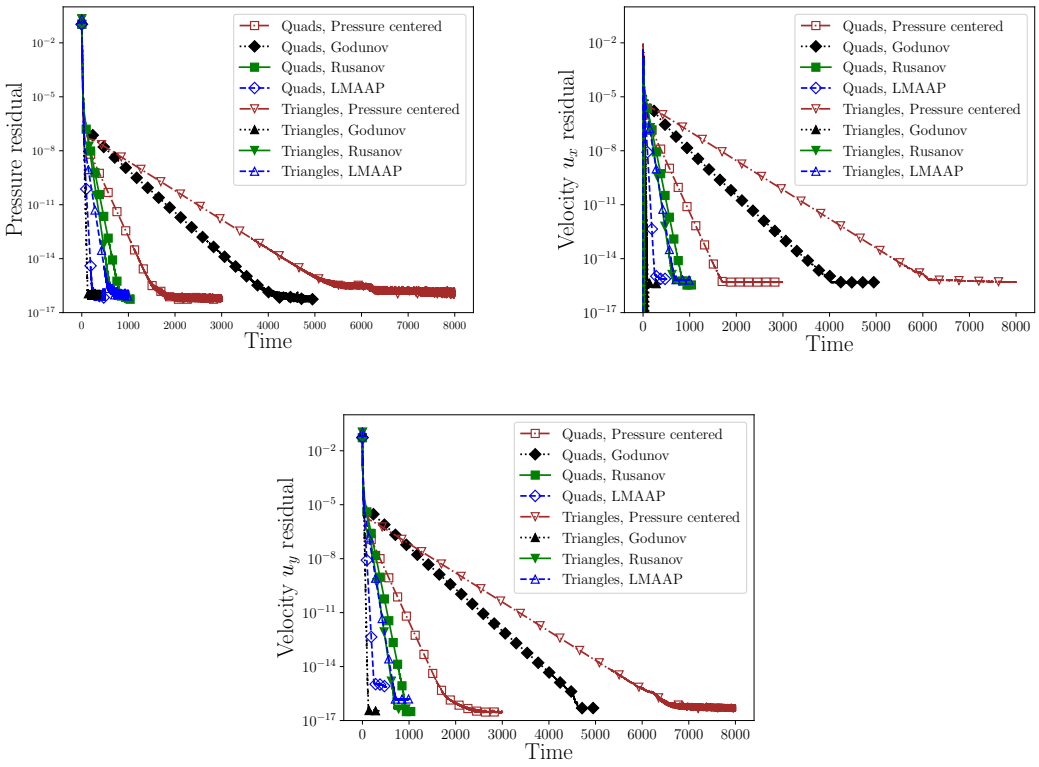


Figure 2.11: Residual obtained with quadrangular and triangular mesh on the pressure p and the velocities u_x and u_y as a function of the time. The LMAAP stabilization matches with the scheme exposed in section 2.3.

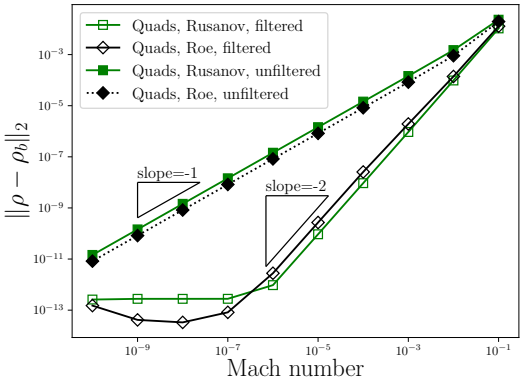


Figure 2.12: L^2 norm of density fluctuation $\rho - \rho_b$ with respect to the Mach number (for $M_b = 10^{-1}$ to $M_b = 10^{-10}$). Rusanov and Roe scheme are used on quadrangular mesh. A log-log plot is used. Results with and without filtering are shown.

- The ones that match with a scheme not accurate at low Mach number. For these schemes, the pressure of the ω-limit found is not 0, and the velocity is far from the exact solution.

This numerically confirms the link between the accuracy at low Mach number and the pressure and velocity of the ω-limit of the wave system.

Moreover, from the ω-limit of the wave system and a low accurate solution at low Mach number, it is possible to build an accurate solution at low Mach number. We take the example of the scattering of a flow by a cylinder. The boundary conditions are as in subsection 2.3.1, and we suppose that a stationary solution ρ^{Euler} and $(\rho\mathbf{u})^{\text{Euler}}$ was obtained. The ω-limit solution of the wave system is denoted by $(\widehat{\rho}^{\infty}, \widehat{\mathbf{u}}^{\infty})$. The Hodge-Helmoltz decomposition can then be applied to $\widehat{\mathbf{u}}^{\infty}$

$$\widehat{\mathbf{u}}^{\infty} = \widehat{\mathbf{u}}^{\infty}_H + \widehat{\mathbf{u}}^{\infty}_{\Psi} + \widehat{\mathbf{u}}^{\infty}_{\varphi}. \tag{2.36}$$

The spurious mode at low Mach number can then be identified as $\widehat{\rho}^{\infty}$ for the density part, and as the non harmonic component of $\widehat{\mathbf{u}}^{\infty}$, $\widehat{\mathbf{u}}^{\infty}_{\Psi} + \widehat{\mathbf{u}}^{\infty}_{\varphi}$ (or equivalently $\widehat{\mathbf{u}}^{\infty} - \widehat{\mathbf{u}}_H$) for the momentum part. This gives the following expression for the filtered solution

$$\begin{cases} \rho^{\text{Filtered}} = \rho^{\text{Euler}} - M_b \rho_b \widehat{\rho}^{\infty} \\ (\rho\mathbf{u})^{\text{Filtered}} = (\rho\mathbf{u})^{\text{Euler}} - \rho_b \|\mathbf{u}_b\| \left(\widehat{\mathbf{u}}^{\infty} - \widehat{\mathbf{u}}^{\infty}_H\right) \end{cases}$$

Numerical results obtained with and without filtering for the density fluctuations are shown with the Roe scheme and the Rusanov scheme in Figure 2.12. These results confirm that filtering allows to recover the $\mathcal{O}\left(M^2\right)$ scaling of the density fluctuations when the Mach number goes to 0. Isovalues of the norm of the velocity with and without filtering are shown with the same numerical fluxes in Figure 2.13, and show that the filtered solution looks similar as the incompressible solution when the filtering is used. Finally, the error between the velocity obtained and the incompressible velocity is depicted in Figure 2.14 for a Mach number of 10^{-4}. Results confirm that the filtered solutions converge, whereas non filtered solutions do not.

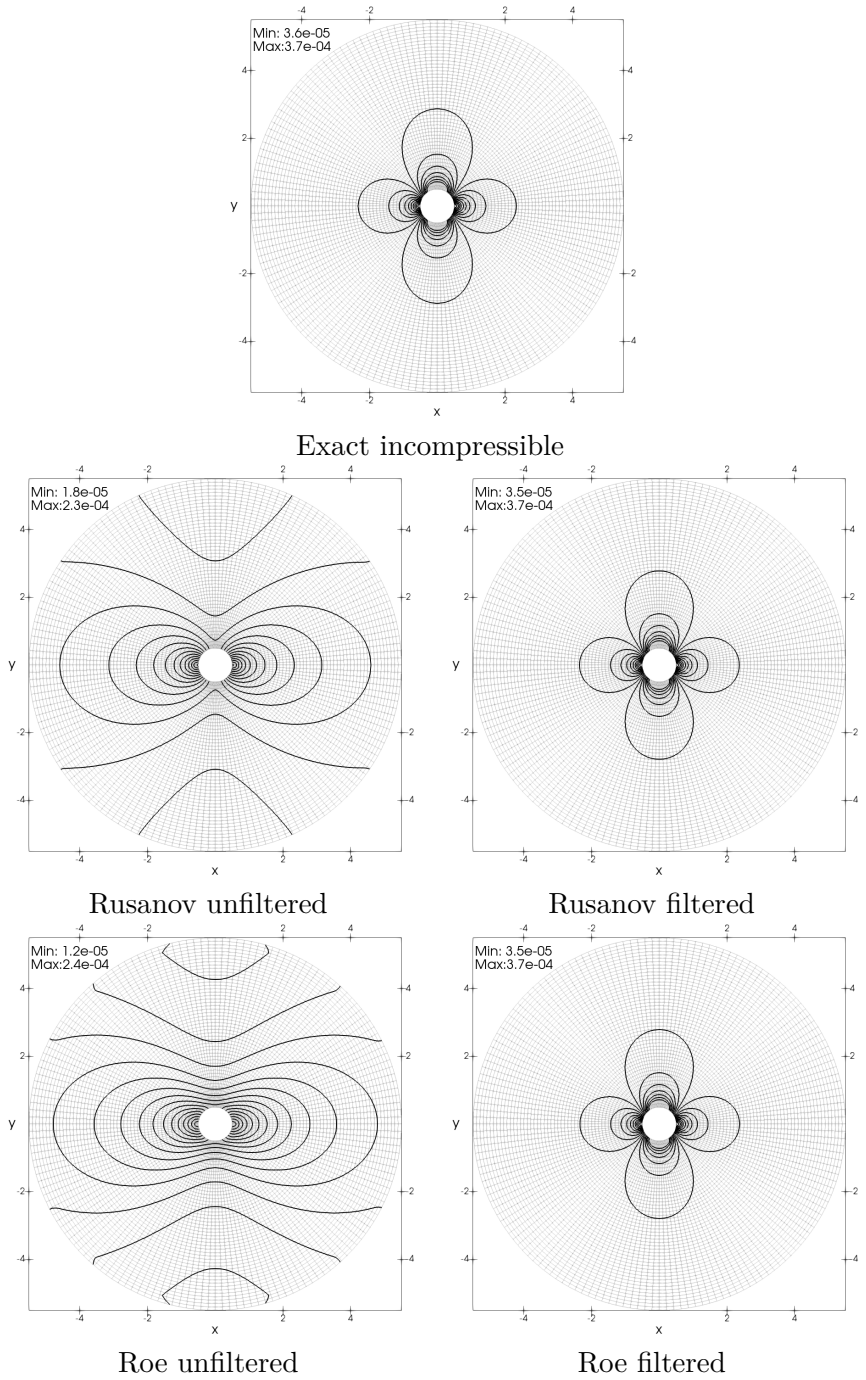


Figure 2.13: Iso-contours of the norm of the velocity obtained at Mach number $M_b = 10^{-4}$ on quadrangular mesh. Results with and without filtering are shown for the Rusanov and the Godunov scheme. Twenty equally distributed contours between 8×10^{-6} and 3×10^{-4} are plotted.

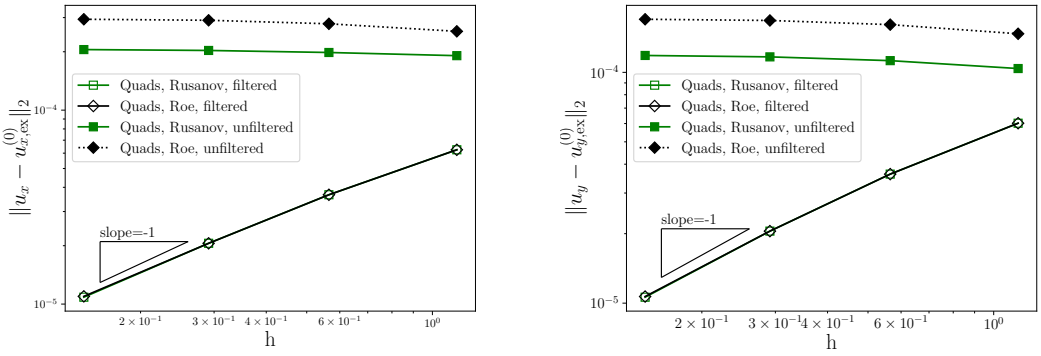


Figure 2.14: L_2 norm of the velocity error between the exact incompressible solution and the long time limit state obtained with the Rusanov and the Godunov scheme. Results with and without filtering are shown for a Mach number of $M_b = 10^{-4}$.

2.5 Behavior of the discontinuous Galerkin method on triangular meshes

It was remarked around ten years ago that the Roe scheme with finite volumes methods has a good accuracy at low Mach number on *triangular meshes* [59, 129, 32]. The aim of this section is to show that the same property holds with the discontinuous Galerkin method.

2.5.1 Notations for the discretization

We denote by V_h a a finite element basis that is composed of piecewise polynomial functions. For each $\varphi \in V_h$, and along any interior side, we denote by

$$[\![\varphi]\!] = \varphi^{Left} - \varphi^{Right},$$

and

$$\{\!\{ \varphi \}\!\} = \frac{\varphi^{Right} + \varphi^{Left}}{2}.$$

Last, we denote by $\mathcal{V}_h = (V_h)^d$ the finite element space of velocities, and by $\mathbb{V}_h = (V_h)^{d+1}$ the finite element space in which the numerical solutions of (2.5) will be searched. Then the discontinuous Galerkin formulation for (2.3) is

$$\begin{aligned} \text{Find } \mathbf{U}_h \in \mathbb{V}_h \quad \forall \boldsymbol{\varphi} \in \mathbb{V}_h \qquad & \sum_{K \in \mathcal{T}_h} \int_K (\boldsymbol{\varphi} \cdot \partial_t \mathbf{U}_h - \mathbf{f}(\mathbf{U}_h) \cdot \nabla \boldsymbol{\varphi}) \\ & + \sum_{S \in \mathcal{S}_i} \int_S [\![\boldsymbol{\varphi}]\!] \cdot \tilde{\mathbf{f}}(\mathbf{U}_h, \mathbf{n}_S) + \sum_{S \in \mathcal{S}_b} \int_S \boldsymbol{\varphi}^{Left} \cdot \mathbf{f}^b(\mathbf{U}_h) \cdot \mathbf{n}_S = 0. \end{aligned} \tag{2.37}$$

where $\mathbf{f}^b$ matches with the weak imposition of the boundary conditions, and $\tilde{\mathbf{f}}(\mathbf{U}_h, \mathbf{n}_S)$ is the numerical flux.

The numerical flux $\tilde{\mathbf{f}}(\mathbf{U}_h, \mathbf{n}_S)$ may be any of known numerical flux: Roe, Lax-Friedrich, HLL, HLLC, Godunov... For the purpose of notation, we will also denote by $\tilde{\mathbf{f}}(\mathbf{U}_L, \mathbf{U}_R, \mathbf{n}_S)$ the numerical flux acting on $\mathbb{R}^{d+1} \times \mathbb{R}^{d+1} \times \mathbb{R}^d$.

2.5.2 Asymptotic expansion of the numerical scheme

As for the finite volume scheme, it is possible to prove uniformity of $\tilde{\rho}_h^{(0)}$ and $\tilde{p}_h^{(0)}$. This is mandatory before considering the higher order moments of the expansion of the numerical scheme in power of the Mach number.

Proposition 11 (Proposition 1 of [74]). *If the numerical flux follows*

$$\tilde{\mathbf{f}}(\mathbf{U}_L,\mathbf{U}_R,\mathbf{n}_S)=\frac{1}{2}\begin{pmatrix}0\\ \frac{1}{\gamma M^2}\left(\tilde{p}_L^{(0)}+\tilde{p}_R^{(0)}\right)\mathbf{n}_S\end{pmatrix}+\begin{pmatrix}\frac{1}{M}\,\tilde{a}_0\left(\tilde{\rho}_L^{(0)},\tilde{\rho}_R^{(0)}\right)\left(\tilde{\rho}_L^{(0)}-\tilde{\rho}_R^{(0)}\right)+\mathcal{O}\left(1\right)\\ \mathcal{O}\left(\frac{1}{M}\right)\end{pmatrix}$$

with $\tilde{a}_0\left(\tilde{\rho}_L^{(0)},\tilde{\rho}_R^{(0)}\right)>0$ *for all* $\tilde{\rho}_L^{(0)}$,$\tilde{\rho}_R^{(0)}$, *and if the boundary conditions follow*

- *either (Steger-Warming boundary condition)*

$$\mathbf{f}_b(\mathbf{U}_L,\mathbf{n}_S)=\frac{1}{2}\begin{pmatrix}0\\ \frac{1}{\gamma M^2}\left(\tilde{p}_L^{(0)}+\tilde{p}_b^{(0)}\right)\mathbf{n}_S\end{pmatrix}+\begin{pmatrix}\frac{1}{M}\,\tilde{b}_0\left(\tilde{\rho}_L^{(0)},\tilde{\rho}_b^{(0)}\right)\left(\tilde{\rho}_L^{(0)}-\tilde{\rho}_b^{(0)}\right)+\mathcal{O}\left(1\right)\\ \mathcal{O}\left(\frac{1}{M}\right)\end{pmatrix}$$

  with $\tilde{b}_0\left(\tilde{\rho}_L^{(0)},\tilde{\rho}_b^{(0)}\right)>0$,

- *or (wall boundary condition)*

$$\mathbf{f}_b(\mathbf{U}_L,\mathbf{n}_S)=\begin{pmatrix}0\\ \frac{1}{\gamma M^2}\,\tilde{p}_L^{(0)}\mathbf{n}_S+\mathcal{O}\left(\frac{1}{M}\right)\end{pmatrix}$$

then $\tilde{\rho}_h^{(0)}$ *is uniform, equal to* $\tilde{\rho}_b^{(0)}$.

It is then possible to prove the following proposition, which ensures the accuracy on triangles and tetrahedra provided the expansion of the numerical flux in Mach number has a special form.

Proposition 12 (Proposition 1 of [74]). *If hypothesis of Proposition 11 hold with* $\tilde{\rho}_b^{(0)}=\rho_0$ *(independent of time), if we denote by* $c_0=\tilde{c}\left(\rho_0\right)$, *and if the numerical flux follows, for* $\tilde{\rho}_L^{(0)}=\tilde{\rho}_R^{(0)}=\rho_0$

$$\tilde{\mathbf{f}}(\mathbf{U}_L,\mathbf{U}_R,\mathbf{n})=\begin{cases}\rho_0\,\dfrac{\left(\tilde{\mathbf{u}}_L^{(0)}+\tilde{\mathbf{u}}_R^{(0)}\right)\cdot\mathbf{n}}{2}+\dfrac{c_0}{2}\left(\tilde{\rho}_L^{(1)}-\tilde{\rho}_R^{(1)}\right)+o\left(1\right)\\ \dfrac{1}{M}\left(c_0^2\,\dfrac{\left(\tilde{\rho}_L^{(1)}+\tilde{\rho}_R^{(1)}\right)\mathbf{n}}{2}+\dfrac{\rho_0 c_0}{2}\left(\tilde{\mathbf{u}}_L^{(0)}\cdot\mathbf{n}-\tilde{\mathbf{u}}_R^{(0)}\cdot\mathbf{n}\right)\mathbf{n}\right)+o\left(\dfrac{1}{M}\right)\end{cases}\tag{2.38}$$

then $\tilde{\rho}_h^{(1)}$ *is uniform, equal to* 0, *and* $\tilde{\mathbf{u}}_h^{(0)}$ *ensures the following side equality*

$$\forall S\in\mathcal{S}_i\qquad[\![\,\tilde{\mathbf{u}}_h^{(0)}\cdot\mathbf{n}\,]\!]=0,$$

and the following cell equality

$$\forall K \in \mathcal{T}_h \quad \forall \varphi \in \mathbb{P}_k(K) \qquad \int_K \tilde{\mathbf{u}}_h^{(0)} \cdot \nabla \varphi = 0.$$

As a consequence, it is solution of the following discrete divergence equation

$$\forall K_i \in \mathcal{T}_h \quad \forall \varphi \in \mathbb{P}_k(K_i) \qquad \int_{K_i} \tilde{\mathbf{u}}_h^{(0)} \cdot \nabla \varphi + \sum_{j \in \mathcal{N}(i)} \int_{\Gamma_{ij}} \varphi \left\{\!\!\left\{ \tilde{\mathbf{u}}_h^{(0)} \cdot \mathbf{n}_{ij} \right\}\!\!\right\} = 0.$$

Note that the asymptotic development (2.38) encloses a lot of Riemann solvers. If we consider the classical Galilean invariant stabilizations of the wave system, there are basically two types of stabilizations that ensure optimal convergence order: the one of (2.38), matching with the Godunov scheme for the wave system, and the one matching with the Lax-Friedrich scheme for the wave system which reads

$$\tilde{\mathbf{f}}(\mathbf{U}_L, \mathbf{U}_R, \mathbf{n}) = \begin{cases} \rho_0 \dfrac{\left(\tilde{\mathbf{u}}_L^{(0)} + \tilde{\mathbf{u}}_R^{(0)}\right) \cdot \mathbf{n}}{2} + \dfrac{c_0}{2} \left(\tilde{\rho}_L^{(1)} - \tilde{\rho}_R^{(1)}\right) + \mathrm{o}\left(1\right) \\ \dfrac{1}{M} \left(c_0^2 \dfrac{\left(\tilde{\rho}_L^{(1)} + \tilde{\rho}_R^{(1)}\right) \mathbf{n}}{2} + \dfrac{\rho_0 c_0}{2} \left(\tilde{\mathbf{u}}_L^{(0)} - \tilde{\mathbf{u}}_R^{(0)}\right) \right) + \mathrm{o}\left(\dfrac{1}{M}\right) \end{cases} \tag{2.39}$$

An exhaustive study for knowing which solver is of the family (2.39) and which is of the family (2.38) has not been conducted yet, but basically, usual Riemann solver that preserve the contacts (Godunov, HLLC, HLLE, Osher) have an asymptotic expansion matching with (2.38), whereas the ones that do not preserve contacts (HLL, Rusanov), are matching with (2.39).

2.5.3 Sketch of the proof

Once the asymptotic expansion has been performed, the system coupling $\tilde{\mathbf{u}}_h^{(0)}$ and $\tilde{\rho}_h^{(1)}$ can be rewritten

$$\forall \varphi \in \mathbb{P}_k(K_i) \qquad \int_{K_i} \rho_0 \varphi \mathrm{div}_{\tilde{\mathbf{x}}} \tilde{\mathbf{u}}_h^{(0)} + \sum_{j \in \mathcal{N}(i)} \int_{\Gamma_{ij}} \varphi \left(-\frac{\rho_0}{2} [\![\tilde{\mathbf{u}}_h^{(0)} \cdot \mathbf{n}_{ij}]\!] + \frac{c_0}{2} [\![\tilde{\rho}_h^{(1)}]\!] \right) = 0. \tag{2.40}$$

$$\forall \boldsymbol{\psi} \in BDM_k(K_i) \quad \int_{K_i} \frac{c_0^2}{\rho_0} \boldsymbol{\psi} \cdot \nabla_{\tilde{\mathbf{x}}} \tilde{\rho}_h^{(1)} - \frac{c_0}{\rho_0} \sum_{j \in \mathcal{N}(i)} \int_{\Gamma_{ij}} \boldsymbol{\psi} \cdot \mathbf{n}_{ij} \left(-\frac{\rho_0}{2} [\![\tilde{\mathbf{u}}_h^{(0)} \cdot \mathbf{n}_{ij}]\!] + \frac{c_0}{2} [\![\tilde{\rho}_h^{(1)}]\!] \right) = 0. \tag{2.41}$$

When $\nabla_{\tilde{\mathbf{x}}} \tilde{\rho}_h^{(1)} = 0$ and $\mathrm{div}_{\tilde{\mathbf{x}}} \tilde{\mathbf{u}}_h^{(0)}$, the equations (2.40) and (2.41) include the boundary integral of the same term multiplied either by $\boldsymbol{\psi} \cdot \mathbf{n}_{ij}$, or by φ. Then in [74], the following is proven

Proposition 13 (Proposition 3 of [74])**.** *If we denote by*

$$R_k(\partial K_i) = \left\{ \phi \quad \phi \in L^2(\partial K_i) \qquad \text{and} \qquad \forall j \in \mathcal{N}(i) \quad \phi_{|\Gamma_{ij}} \in \mathbb{P}_k(\Gamma_{ij}) \right\},$$

and by

$$\begin{array}{rcl} \mathrm{Tr} : p \in \mathbb{P}_k(K) & \longmapsto & p_{|\partial K} \in R_k(\partial K) \\ \mathscr{T} : \mathbf{u} \in BDM_k(K) & \longmapsto & \mathbf{u}_{|\partial K} \cdot \mathbf{n}_{ij} \in R_k(\partial K) \end{array}$$

then, if K is a simplex

$$R_k(\partial K) = \mathrm{Tr}(\mathrm{div}_{\mathbf{x}}^{\perp}) \oplus \mathscr{T}(\nabla_{\mathbf{x}}^{\perp}).$$

Using Proposition 13 on (2.40) and (2.41) leads to

$$-\frac{\rho_0}{2}[\![\,\tilde{\mathbf{u}}_h^{(0)}\cdot\mathbf{n}_{ij}\,]\!]+\frac{c_0}{2}[\![\,\tilde{\rho}_h^{(1)}\,]\!]=0$$

on each side. Using the same arguments as in [59] leads to $[\![\,\tilde{\rho}_h^{(1)}\,]\!]=0$ and $[\![\,\tilde{\mathbf{u}}_h^{(0)}\cdot\mathbf{n}_{ij}\,]\!]$. The remaining part of the proof is straightforward.

2.5.4 Numerical results

In this section, we test the results claimed in Proposition 12 on the test case of the scattering of a flow by a cylinder. This test case is performed with the Roe and HLLC schemes (which are of family (2.38)), and with the HLL and Rusanov schemes (which are of family (2.39)). In Figure 2.16, isocontours of the norm of the velocity have been drawn for the stationary solution obtained at Mach 10^{-6}. These isocontours confirm that when the HLLC and Roe scheme are used, the discontinuous Galerkin method is accurate, whereas it is not when the HLL and Rusanov numerical flux are used.

In Figure 2.15, the density fluctuations $\rho-\rho_b$ are plotted as a function of the Mach number M_b. As expected, a rate of 2 is obtained for the HLLC and the Roe schemes. A rate of 1 is obtained with the Rusanov and the HLL schemes. Note that the gap between accurate and inaccurate schemes when the Mach number goes to 0 appears obviously only for a very low Mach number when the order of accuracy increases.

2.6 Conclusion and prospects on low Mach number flows

In this chapter, several aspects of low Mach number flows have been addressed. First, the problem of the propagation of an acoustic wave in a low Mach number flow, on which few work has been done until now was addressed. It was shown that usual low Mach number fixes are unable to correctly solve acoustic wave propagation in low Mach number flows: either acoustic waves are quickly damped, or the order of accuracy is suboptimal. A solution was proposed, which allows to recover an accurate solution, both for stationary low Mach number flows, and for acoustic wave propagation.

Then another solution was proposed for the low Mach stationary case: instead of trying to fix the Riemann solver, the spurious mode that jeopardizes the convergence at low Mach number was thoroughly studied: it appears as being the long time limit of a discretization of a wave system. Based on this knowledge, a low accurate solution at low Mach number can be filtered with this long time limit for recovering accuracy.

Last, an extension of the good accuracy of some Riemann solvers on triangles and tetrahedra for the finite volume method was proven for the discontinuous Galerkin method.

The results obtained within this chapter will be extended in the following directions over the next years

Extend the work on acoustic wave propagation in a low Mach number flow to the full Euler system. In this chapter, the low Mach number problem was addressed only with the barotropic Euler model. The full Euler model raises some challenges, especially concerning the two time scales asymptotic expansion which we addressed in this chapter. Indeed the same type of wave system coupling is found between the first order pressure and the zeroth order momentum. However, the additional equation on density (or entropy) is not straightforward to deal with.

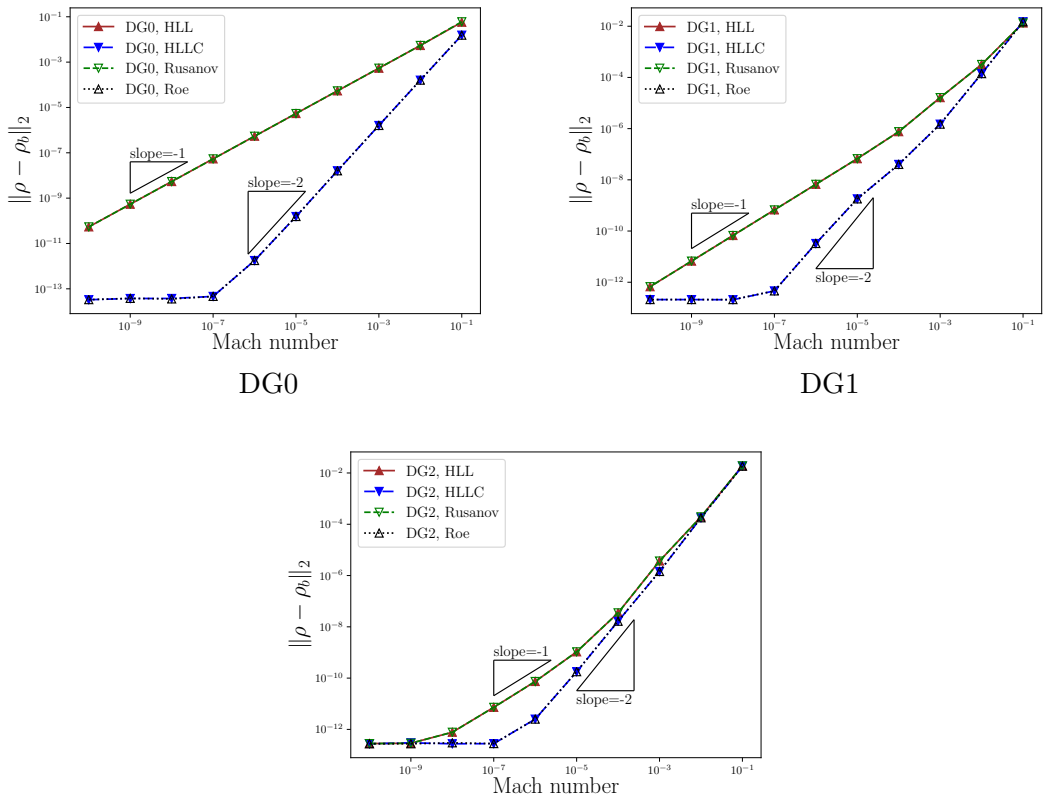


Figure 2.15: L^2 norm of the density fluctuation $\rho - \rho_b$ with respect to the Mach number (for $M_b = 10^{-1}$ to $M_b = 10^{-10}$).

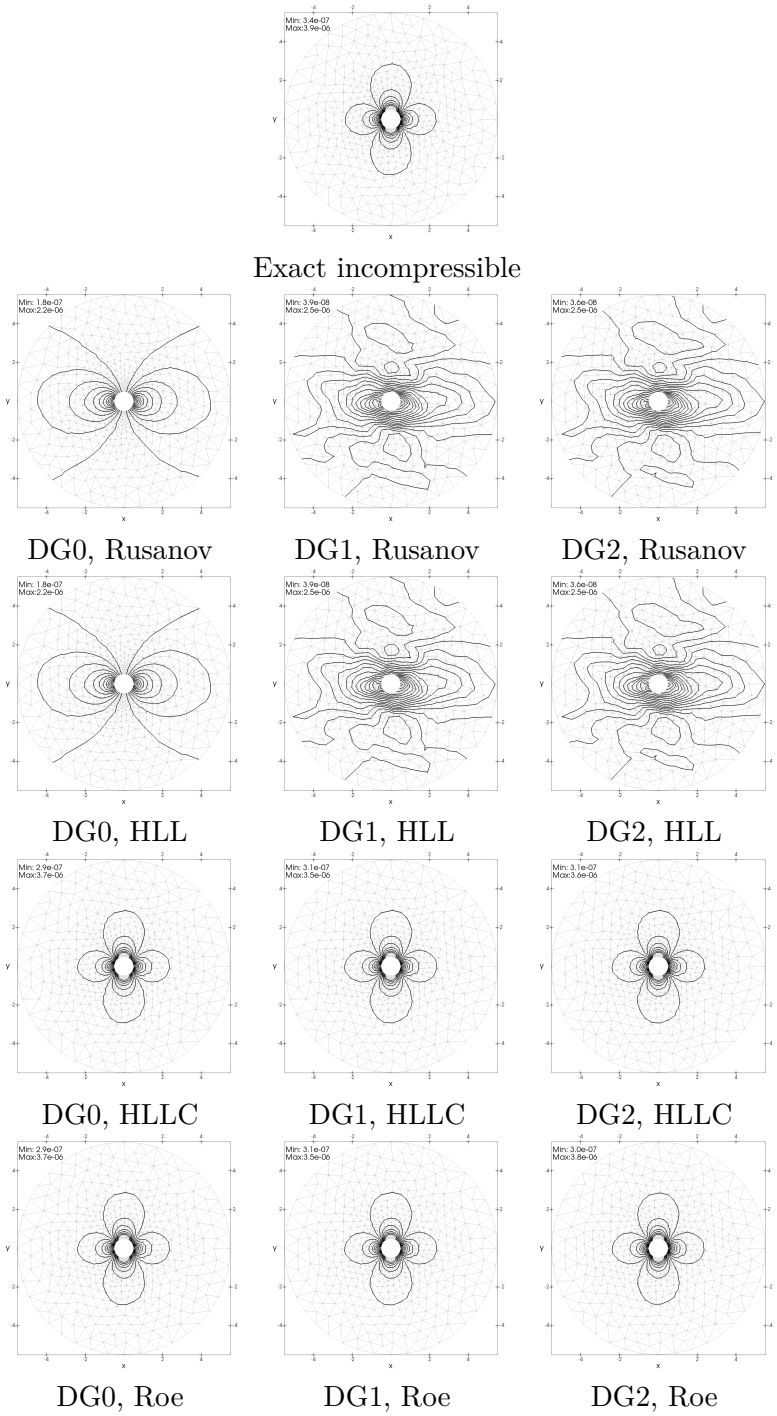


Figure 2.16: Iso-contours of the norm of the velocity obtained at Mach number $M_b = 10^{-6}$ on triangular mesh. Twenty equally distributed contours between 8×10^{-8} and 3×10^{-6} are plotted.

Develop a well balanced approach for low Mach number flows. Well balanced discretization have been developed over the three last decades for ensuring a good resolution of hyperbolic systems with source terms, for example, shallow water system with topography or Euler system with gravity. This well-balanced approach can be recast as searching a solution as the deviation from an equilibrium state. As we have been able to identify the spurious mode when a scheme is not low Mach number accurate, it may be interesting to consider a new problem, formulated as the deviation with respect to this spurious mode.

Disseminate to other type of hyperbolic systems with similar problems. In section 2.3, we saw that the spurious mode obtained at low Mach number is characterized by a nonzero first order density, and a momentum that is not divergence free. Other types of system suffer from the same problem. For example, the Magnetohydrodynamics system includes a $\mathrm{div}_{\mathbf{x}}\mathbf{B} = 0$ constraint which is hard to ensure from a numerical point of view [29]. The work proposed in the previous item could be another way to address this problem.

Chapter 3

Development of the AeroSol library and applications

3.1 Introduction

The INRIA Cagire project began in 2011. The initial project consisted in performing cross comparisons between numerical computations and experiments on a special flow configuration, relevant for effusion cooling of wall of combustion chambers: the jet in cross flow. The experimental part was already developed within the experimental bench Maveric. For performing the numerical part, a new library began to be developed. We wanted the library to be scalable and high order based on discontinuous Galerkin methods. Moreover, in order to mesh the boundary layers with a mesh structured in the orthogonal direction of the wall, which requires to use hexahedron or prisms near walls, we wanted to work with hybrid meshes. Last, the cells had to be curvilinear for reaching the optimal approximation order on curved geometries.

In the meanwhile, the INRIA team Bacchus had similar plans, but wanted to work with continuous finite elements. Moreover, this team used to include researchers on high performance computing and wanted to explore NUMA aspects and accelerators for finite elements computations. Two PhD students, Damien Genet and Cédric Lachat had begun their work around this: Cédric had begun to develop the library PaMPA, a library dedicated to the distributed management of mesh based methods and to the parallel mesh adaptation, and Damien had began to work on the link between PaMPA and a new C++ high order finite element library which would become AeroSol. The PaMPA library was supposed to handle the distributed parallelism, whereas the node or node+accelerator level was supposed to be done with the runtime scheduler StarPU. The library relies on external libraries for solving linear systems. The global design of the library is shown in Figure 3.1.

This chapter is organized as follows: in section 3.2, the general design of the code is explained; the library was jointly designed with Damien Genet during his PhD thesis [55]. In section 3.3, some validations and performance results are shown. In section 3.4, some of the results obtained within the Impact-ae European project are shown; these results were obtained jointly with Simon Delmas during his PhD thesis [34]. In section 3.5 some preliminary results obtained with StarPU are shown; these results were obtained jointly with the authors of [44]. The chapter ends with the prospects on this library in section 3.6.

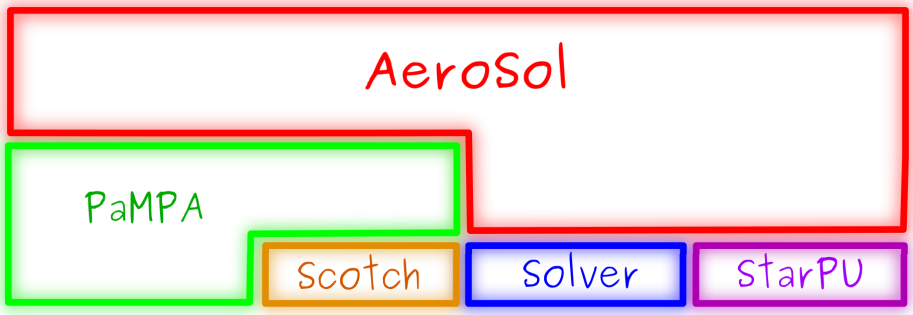


Figure 3.1: Global initial design of the AeroSol library. For the memory management, it relies on the PaMPA library, a superlayer of the parallel graph manager scotch/pt-scotch. It relies on external linear solvers for system resolutions, and on StarPU for addressing NUMA architectures and host/accelerators architectures.

3.2 Code design

3.2.1 Memory management

The AeroSol library is designed for being able to deal with continuous and discontinuous high order finite elements methods. For a discontinuous discretization, the unknowns are distributed on the cells. Their number depends on the shape and the approximation degree. The number of unknowns for discontinuous discretizations is recalled in Table 3.1. When dealing with continuous discretization, the continuity constraint leads to a smaller number of unknowns, but with a larger spreading on all the entities of the mesh. The number of unknowns for continuous discretizations is recalled in Table 3.2.

For handling memory, an undirected graph based on the mesh is defined. The vertices of this graph are *all* the entities of the mesh, namely all the parts of the mesh that *may* bear data: points, lines, triangles, quads, tetrahedra, hexaedra, prisms and pyramids. An edge links two vertices of the graph if one of the two entities *may* need the other one for completing its unknown. For example, with a continuous discretization, a line bears only $(k-1)$ unknowns, and the two missing unknowns are located on the points linked with this line. Therefore an edge links the vertex of the line and each of the vertex of its points. The resulting graph includes all the cells, faces, sides and points of the mesh, and the edges of the graph are all the links between each cell and its faces, sides and points. This graph containing all the entities is called the *enriched graph*. A mesh example and its associated *enriched graph* are shown in Figure 3.2.

Once the graph is built, we would like to keep the original informations of the mesh, namely with which type of entity it matches (face, cell, side), and which shape it is (triangle, quadrangle, tetrahedron, etc...). Moreover, additional informations are needed on the vertices of the graph: the approximation degree, additional informations on faces (whether it is boundary, interior or periodic), and the curvature degree of the entity. All these informations are compactly stored in a 32-bit integer called an *entity type*. Each of this information has a reserved number of bits within this integer. The approximation degree is coded on five bits, the face information on two bits, the shape on three bits, and the curvature on four bits. For

	1d	2d	3d
Points	0	0	0
Lines	$k+1$	0	0
Triangles	0	$\frac{(k+1)(k+2)}{2}$	0
Quads	0	$(k+1)^2$	0
Tetrahedron	0	0	$\frac{(k+1)(k+2)(k+3)}{6}$
Hexahedron	0	0	$(k+1)^3$
Prisms	0	0	$\frac{(k+1)^2(k+2)}{2}$
Pyramids	0	0	$\frac{(k+1)(k+2)(2k+3)}{6}$

Table 3.1: Localization and number of unknowns for discontinuous discretizations with approximation degree $k \geqslant 0$.

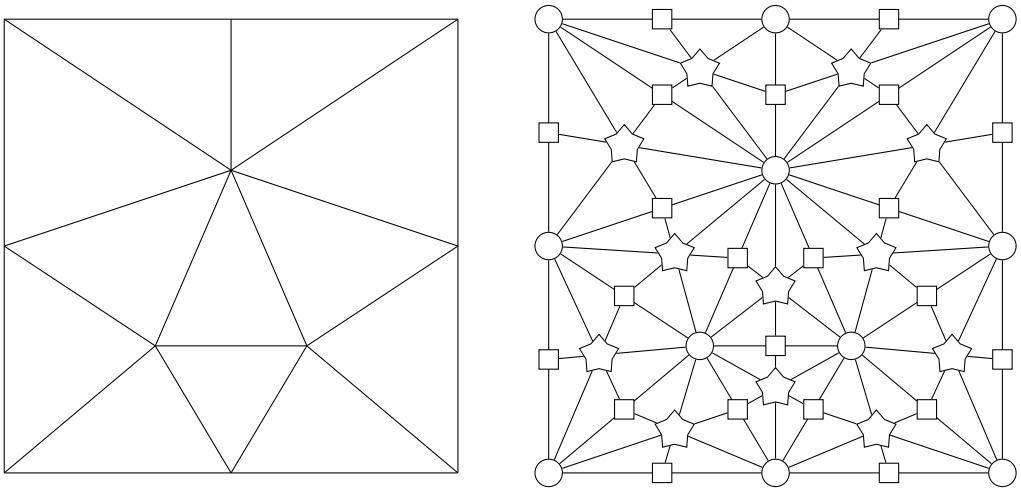


Figure 3.2: The initial mesh (left), and its graph representation (right). In the graph representation, vertices matching with cells are represented with a star shaped, vertices matching with sides are represented with square shape, and vertices matching with points are represented with circles.

Points	1
Lines	$k-1$
Triangles	$\max\left(\frac{(k-1)(k-2)}{2}, 0\right)$
Quads	$\max\left((k-1)^2, 0\right)$
Tetrahedron	$\max\left(\frac{(k-1)(k-2)(k-3)}{6}, 0\right)$
Hexahedron	$\max\left((k-1)^3, 0\right)$
Prisms	$\max\left(\frac{(k-1)^2(k-2)}{2}, 0\right)$
Pyramids	$\max\left(\frac{(k-1)(k-2)(2k-3)}{6}, 0\right)$

Table 3.2: Localization and number of unknowns for continuous discretizations with degree $k \geqslant 1$. On the contrary to the discontinuous case (see Table 3.1), the number of unknowns does not depend on the dimension.

example, the bitwise definition of a tetrahedron with approximation degree 4, of curvature 2 is the following integer

$$\underbrace{00100}_{\text{degree}}\ \underbrace{00}_{\text{face information}}\ \underbrace{110}_{\text{shape}}\ \underbrace{0100}_{\text{curvature}} = 2148.$$

We are now concerned with the memory allocation. For the same mesh as in Figure 3.2, the localization of the degrees of freedom are shown in Figure 3.3 for discontinuous discretization, and on Figure 3.4 for continuous discretization for different approximation degrees. The memory allocation is based on the fact that for a given approximation type and approximation degree, all vertices of the *enriched graph* with the same *entity type* bears the same number of degrees of freedom. Therefore a pointer is allocated for each *entity type* of the mesh. The size of the pointer for the *entity type* e is obtained by multiplying the number of vertices of the *enriched graph* of *entity type* e by the number of degrees of freedom borne by each entity of this type given in Table 3.1 and Table 3.2 depending on whether continuous or discontinuous discretizations are addressed.

The memory allocation for continuous and discontinuous discretizations of degree 4 is shown in Figure 3.5

- In the **continuous case**, each point bears 1 degree of freedom, each side bears 3 degrees of freedom, and each triangle bears 3 degrees of freedom. A pointer is allocated for each of the entity bearing data. The mesh contains 11 points, 22 sides, and 12 cells. Therefore, 3 pointers are allocated: one of size 11 for point data, one of size 66 for side data, and one of size 36 for triangle data.

- In the **discontinuous case**, only the triangles bear data, and each triangle bears 15 degrees of freedom. As the mesh contains 12 cells, 1 pointer of size 180 is allocated for triangle data.

The resulting vector of unknowns is a collection of pointers, which are gathered in a `std::map` indexed by the *entity type*.

Note that this way of managing the data is very generic for finite elements methods. It can be extended easily to other types of high order discretization. For example, we are currently extending it to cubature based continuous finite elements, in which we should only change the values given in Table 3.2. This could also be extended for example to edge based finite elements, and to hybridizable discontinuous Galerkin methods.

From an implementation point of view, the mesh is read in the AeroSol library, and the parallel *enriched graph* is built from the parallel mesh reading. The enriched graph is given to the PaMPA library. Then the PaMPA library is able to redistribute the *enriched graph* and to compute the appropriate overlap. From the graph, the PaMPA library provides iterators for looping on cells, sides, cells of sides,... PaMPA is also able to perform point-to-point communications.

3.2.2 Basic classes and their unit test

All the computations in the AeroSol library are based on the reference element. Suppose that we want to integrate the cell term of the discontinuous Galerkin approximation of a hyperbolic system

$$\partial_t \mathbf{U} + \mathrm{div}_{\mathbf{x}} \mathbf{F}(\mathbf{U}) = 0,$$

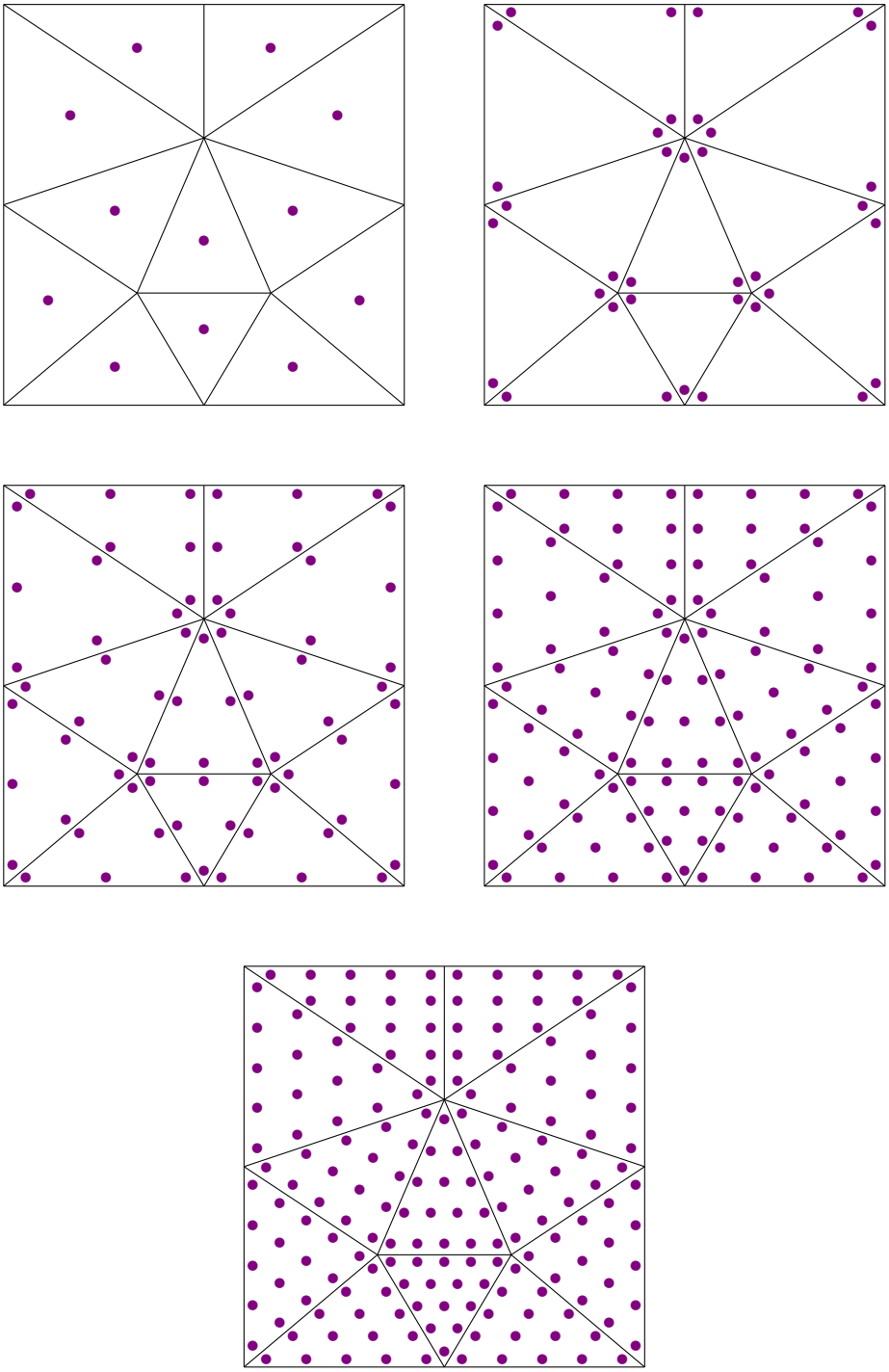


Figure 3.3: Distribution of the unknowns for discontinuous discretization on a triangular mesh with approximation degree 0,1,2,3 and 4.

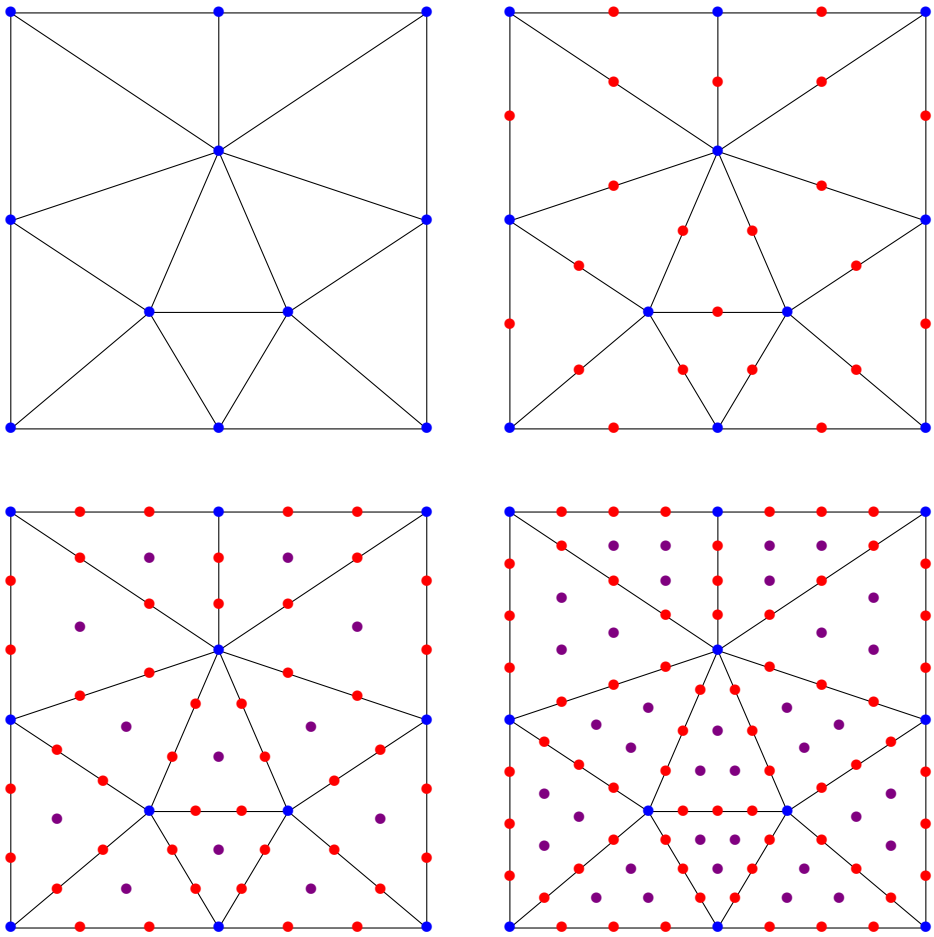


Figure 3.4: Distribution of the unknowns for continuous discretization on a triangular mesh with approximation degree 1,2,3 and 4.

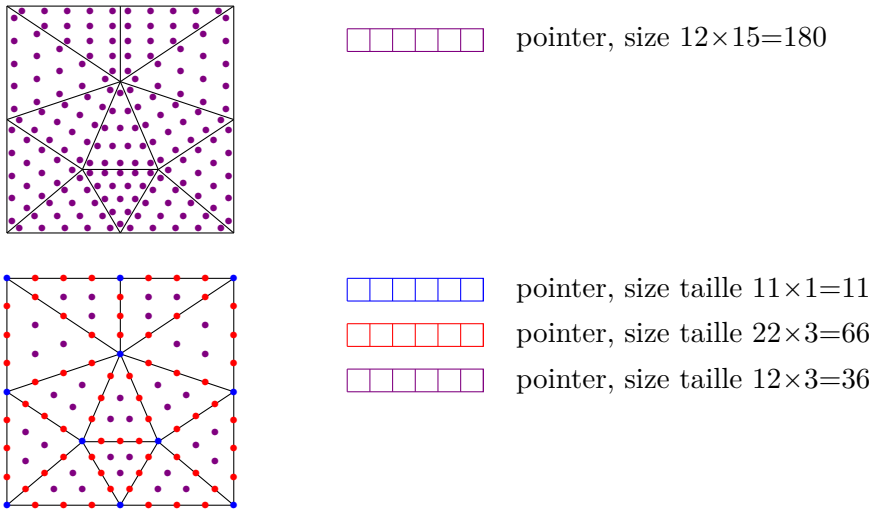


Figure 3.5: Memory allocation for a triangular mesh with discontinuous and continuous discretization with an approximation degree 4. The number of squares does not depict the pointer size.

on the cell K which is

$$\forall k \qquad \int_K \mathbf{F}(\mathbf{U}(\mathbf{x})) \cdot \nabla \varphi_k(\mathbf{x}) \, \mathbf{dx} \tag{3.1}$$

where φ_k are the finite element basis. We denote by T_K the transformation that maps the reference element $\hat{K}$ onto the physical element K, and by $\{\hat{\varphi}_k\}$ the finite element basis on the reference element. Then the finite element basis $\{\varphi_k\}$ on the cell K is defined as

$$\forall k \qquad \varphi_k(\mathbf{x}) = \hat{\varphi}_k \left(T_K^{-1}(\mathbf{x}) \right)$$

We denote by $\hat{\mathbf{x}}$ the variables in the reference element, and we apply the variable change $\mathbf{x} = T_K(\hat{\mathbf{x}})$ in (3.1), which gives

$$\begin{aligned} \int_K \mathbf{F}(\mathbf{U}(\mathbf{x})) \cdot \nabla \varphi_k(\mathbf{x}) \, \mathbf{dx} &= \int_{\hat{K}} \mathbf{F}(\mathbf{U}(T_K(\hat{\mathbf{x}}))) \cdot \nabla \varphi_k \left(T_K \left(\hat{\mathbf{x}} \right) \right) \left| \det DT_K(\hat{\mathbf{x}}) \right| \mathbf{d\hat{x}} \\ &= \int_{\hat{K}} \mathbf{F}(\mathbf{U}(T_K(\hat{\mathbf{x}}))) \cdot DT_K(\hat{\mathbf{x}})^{-1} \nabla \hat{\varphi}_k(\hat{\mathbf{x}}) \left| \det DT_K(\hat{\mathbf{x}}) \right| \mathbf{d\hat{x}}. \end{aligned}$$

Last, this integral is approximated with a quadrature formula of size n_{qf} with points $\hat{\mathbf{x}}_i$ and weights ω_i

$$\int_K \mathbf{F}(\mathbf{U}(\mathbf{x})) \cdot \nabla \varphi_k(\mathbf{x}) \, \mathbf{dx} \approx \sum_{i=1}^{n_{qf}} \omega_i \mathbf{F}(\mathbf{U}(T_K(\hat{\mathbf{x}}_i))) \cdot DT_K(\hat{\mathbf{x}}_i)^{-1} \nabla \hat{\varphi}_k(\hat{\mathbf{x}}_i) \left| \det DT_K(\hat{\mathbf{x}}_i) \right|. \tag{3.2}$$

For computing this last expression, we therefore need

- a finite element basis defined on the reference element $\hat{\varphi}$, and the expression of the gradient of the finite element basis.
- a quadrature formula defined on the reference element $(\omega_i, \hat{\mathbf{x}}_i)$,

- a geometrical transformation between the reference element $\hat{K}$ and the physical element K, including the interpolation operator T_K, invert of its Jacobian and the determinant of the Jacobian,
- the definition of the physical flux $\mathbf{F}$.

In AeroSol, these classes are considered as *basic*, in the sense that unit tests can be developed independently from any other class within the library. For example, each quadrature formula is implemented in one class, and all the quadrature formulae inherits from a virtual class

```
class QuadratureFormula::Interface {

  //Get the degree of the quadrature formula
  virtual unsigned int getDegree() const=0;

  //Return the ith weight
  virtual double omega(const unsigned int& i) const=0;

  //Return the ith point
  virtual Coord3 point(const unsigned int& i) const=0;

};
```

The virtual functions are implemented in the bottom classes. A unit test is then developed for checking that the implemented quadrature formula is of the degree returned by the function `getDegree()`, i.e. that the quadrature formula is exact for all the polynoms of degree lower or equal to `getDegree()`. This unit test also checks that for triangles and tetrahedra, the formula is symmetric with respect to the element.

In the same manner, a unit test is developed for checking that the finite element basis, that inherits from the class

```
class FiniteElement::Interface {

  //Fill the pointer values of the finite element basis evaluated in c
  virtual void fillBasisFunction( const Coord3& c, Real* const values) const =0;

  //Fill the matrix derivatives of the gradient of the finite element basis
  //evaluated in c
  virtual void fillBasisFunctionDerivative( const Coord3& c, Matrix& derivative) const =0;

};
```

is consistent. Unisolvance is checked by checking either that the basis is a Lagrange basis, or by checking that it is an orthogonal basis for spectral basis. The function returning the gradient of the basis functions, `fillBasisFunctionDerivative` is validated by checking that the numerical derivative of `fillBasisFunction` matches with the result of `fillBasisFunctionDerivative` on a significant number of different points of the reference element.

The validation of the classes computing the geometry are based on similar ideas as for the finite element classes: the consistency between the function defining the interpolation operator T_K and DT_K, DT_K^{-1} and $\det DT_K$ are checked by computing numerical derivatives of T_K.

Last, The validation of the model class is led with the same ideas. For example, for a hyperbolic model, functions are defined for computing the d components of the flux, the

normal flux, the Jacobian of the flux, the Jacobian in the normal direction, and the eigenvalues of the Jacobian in the normal direction. Then consistency between the flux and the Jacobian is checked by numerical derivative, and the consistency between the normal Jacobian and the eigenvalues is checked by resorting to Lapack for computing a numerical approximation of eigenvalues.

3.2.3 Unit tests for data management

The aim of the AeroSol library is to deal with high order continuous and discontinuous finite element discretization. This requires to manage correctly data, including in a parallel environment.

For continuous finite elements, the challenge consists in gathering data that may be spread on different pointers associated to its points, edges, faces. The data management test in this case consists in performing a loop on the cells. On each cell, the degrees of freedom are filled with the Lagrange points of the cell, and scattered in the different pointers associated with points, edges, faces. These pointers may be visited several times, because data of points, faces and edges are shared between several cells. In a second loop, Lagrange points of the cell are recomputed, and checked to match with the data of the cells. This test can also checks the parallel memory management, by performing a synchronization of data after the first cell loop.

For discontinuous approximations, the challenge is to perform the matching between the right and left values of the faces of the mesh. For this, the data on the cells are initialized by using a continuous polynomial function which degree is lower than the approximation degree, therefore ensuring the continuity of the projection of the polynomial function. Then, a loop on the faces is performed. Data of neighbouring cells are gathered, and interpolated on the common face. The test consists then in checking that data on the left and right cells are matching on the common face. This test can also be performed in parallel by performing a synchronization of data after the initialization.

For continuous and discontinuous discretizations, another unit test consists in checking that the interpolation or finite elements projection of any polynomial function of degree lower or equal to the approximation degree is exact.

3.2.4 Loop nesting and choice of abstraction

AeroSol is developed in C++. This language offers two ways of doing abstraction

- **Polymorphism.** Polymorphism consists in defining a hierarchy of classes inheriting from a virtual `Interface` class. The code is written using a pointer towards this `Interface`. The pointer is then allocated at runtime. This implementation may allow a fast development of a library, but the polymorphism is known to be costly at runtime because of the overheads induced by the calls to the virtual table.

- **Template.** When using templates, the different implementations are explicitly compiled. The compilation time maybe large, but a faster execution is expected.

The tradeoff between these two types of abstractions is not always easy to do; our choice was based on the nesting of the loops within a typical time loop in which (3.1) is computed at each time step, for which the algorithm is written in algorithm 1.

```
Initialize U to U_0;
for iTime =0 to NTime do
    for iCell =0 to NCell do
        Gather data of cell iCell;
        Set cell residual to 0;
        for iqp =0 to NQuadraturePoints do
            Compute the finite elements basis at the quadrature point iqp ;
            Compute U at the quadrature point;
            Compute F(U) at the quadrature point;
            Compute the geometrical transformations DT_K^{-1} and det DT_K;
            Add the increment at the quadrature point iqp to the residual ;
        end
        Scatter data of the residual of cell iCell;
    end
    Add the residual to U;
end
```

Algorithm 1: Time loop for a spatial numerical scheme including the integral (3.1)

Polymorphism induces a small time overhead each time a pointer towards an abstract class is used. This type of abstraction is therefore very costly when used in the most nested loops. In algorithm 1, it is clear that polymorphism should not be used for finite elements basis, quadrature formula, and geometrical transformations. For low order, the quadrature points loop maybe quickly executed, so that we choose to use polymorphism only at the level of the time loop, and not below.

Last, we clarify how the three nested loops are implemented in AeroSol: the top one (the time loop) is implemented with polymorphism. All the time iteration classes inherits from an abstract class `TimeIterator::Interface`, and is allocated at runtime. Time schemes implemented are explicit Runge-Kutta, implicit BDF schemes, stationary linear solvers, and stationary nonlinear solvers based on Newton method. Time iterator classes contain a pointer towards an abstract class `SpatialScheme::Interface`, in which the spatial discretization (residual and Jacobian) are computed. The memory management, and so the link with the PaMPA library is done at this level. The cell and face loops are also performed at this level, therefore matching with the second nested loop of algorithm 1. The third nested loop (the loop on the quadrature points) is performed in a different type of class, called `Integrator`, which we detail in the next section. The collaboration scheme between these classes and the dependencies is summarized in Figure 3.6.

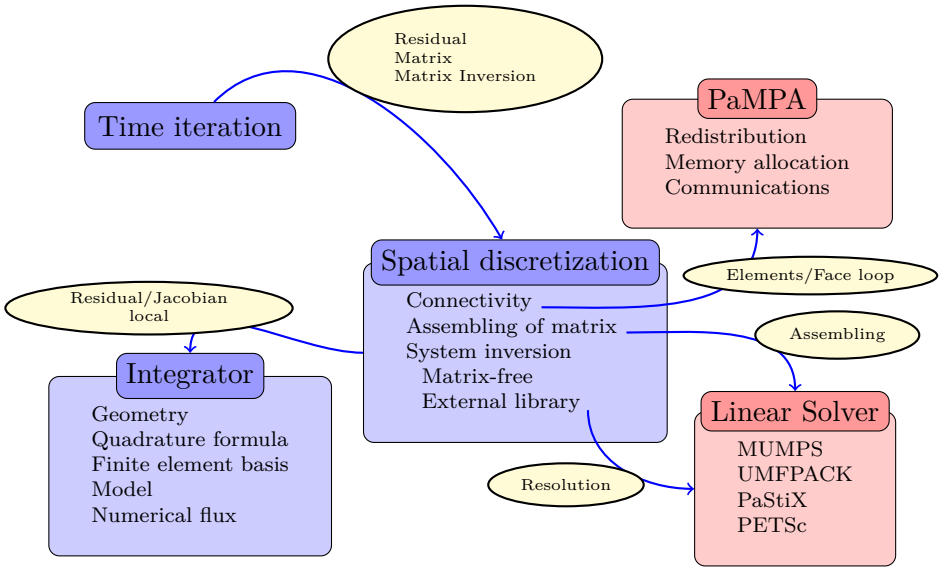


Figure 3.6: Main classes of the libary AeroSol and their collaboration with the PaMPA library and with external solver libraries.

3.2.5 Integrator classes

The most general model that is supported in AeroSol reads

$$\partial_t \mathbf{U} + \mathrm{div}_{\mathbf{x}} \mathbf{F}(\mathbf{U}) + \sum_{i=1}^{d} B_i(\mathbf{U}) \partial_{\mathbf{x}_i} \mathbf{U} = \mathrm{div}_{\mathbf{x}} \left(A(\mathbf{U}) \nabla \mathbf{U} \right) + S(\mathbf{U}).$$

The discontinuous Galerkin discretization with the BR2 stabilization of the parabolic terms reads

$$\begin{aligned}
&\sum_{K\in\mathcal{T}_h} \int_K \varphi \partial_t \mathbf{U} - \sum_{K\in\mathcal{T}_h} \int_K \left(\mathbf{F}(\mathbf{U}) \nabla\varphi - \sum_{i=1}^{d} \varphi B_i(\mathbf{U}) \partial_{\mathbf{x}_i} \mathbf{U} \right) + \sum_{S\in\mathcal{S}_i} \int_S \tilde{\mathbf{F}}(\mathbf{U},\mathbf{n}) [\![\varphi]\!] \\
&\quad + \sum_{S\in\mathcal{S}_b} \int_S \tilde{\mathbf{F}}^b(\mathbf{U},\mathbf{n}) [\![\varphi]\!] = - \sum_{K\in\mathcal{T}_h} \int_K A(\mathbf{U}) \nabla \mathbf{U}) \nabla\varphi + \sum_{S\in\mathcal{S}_i} \int_S \{\!\{ A^T(\mathbf{U}) \nabla\varphi \}\!\} [\![\mathbf{U}]\!] \\
&\quad + \sum_{S\in\mathcal{S}_i} \int_S \{\!\{ A(\mathbf{U}) \nabla \mathbf{U} \}\!\} [\![\varphi]\!] + \sum_{S\in\mathcal{S}_i} \int_S \{\!\{ A(\mathbf{U}) r_h([\![\mathbf{U}]\!]) \}\!\} [\![\varphi]\!] \\
&\quad + \sum_{S\in\mathcal{S}_b} \int_S \{\!\{ A^T(\mathbf{U}) \nabla\varphi \}\!\} [\![\mathbf{U}]\!]^b + \sum_{S\in\mathcal{S}_b} \int_S \{\!\{ A(\mathbf{U}) \nabla \mathbf{U}^b \}\!\} [\![\varphi]\!] \\
&\quad + \sum_{S\in\mathcal{S}_b} \int_S \{\!\{ A(\mathbf{U}) r_h([\![\mathbf{U}]\!]^b) \}\!\} + \sum_{K\in\mathcal{T}_h} \int_K \varphi S(\mathbf{U}) [\![\varphi]\!]
\end{aligned} \tag{3.3}$$

In the AeroSol library, each of the integral of the above numerical scheme has a dedicated integrator. These integrators may depend on the model (all but the mass one) through $\mathbf{F}$, A or S, and some may depend on the numerical flux $\tilde{\mathbf{F}}$. As said before, the abstraction at this level is obtained by template. Each needed integrator is initialized on the matching *entity type* (see subsection 3.2.1). During initialization, memory is allocated for storing finite

element basis, quadrature formula, and geometrical functions. Considering (3.2), we see that the finite element basis *functions* are not needed, as the functions or their gradients are always evaluated on the quadrature points of the reference element, no matter the geometry of the physical element is. Therefore the finite element basis informations are stored as a pointer of size $n_{qf} \times n_{dof} \times (dim + 1)$. Still considering (3.2), the geometrical functions are needed because they differ from one physical element to another one, even if they are of the same *entity type* (i.e. are the image of the same reference element, with same approximation degree and curvature degree). We could have resorted to template abstraction, however if we consider the number of possible shapes, 8, and the number of possible curvature degrees, which we fixed to 4, this makes 32 possibilities and would have considerably slowed down the compilation. Instead, we used another feature of C++, the *function pointers*: this avoids both template and polymorphism, but lose the class features (e.g. private members of a class) of C++, which fortunately were useless for geometrical functions. All the needed data on one entity type (quadrature formula, finite element basis evaluated on the reference quadrature points, and function pointers for geometrical functions) are stored in a `std::map` indexed by the *entity type*.

3.2.6 Optimization

In the implementation presented in the previous section, the geometrical data involved in (3.2) ($\det T_k(\mathbf{x}_i)$, $DT_k(\mathbf{x}_i)$ and its invert) are computed on each cell, and at each time step. However, if the mesh does not move, all these data are constant and can be computed once for all. Still taking the example of (3.2), the expression to be evaluated is

$$\sum_{iq=1}^{n_{qf}} \omega_i \mathbf{F}(\mathbf{U}(T_K(\hat{\mathbf{x}}_i))) \cdot DT_K(\hat{\mathbf{x}}_i)^{-1} \nabla \hat{\varphi}_k(\hat{\mathbf{x}}_i) \left|\det DT_K(\hat{\mathbf{x}}_i)\right|.$$

This evaluation can be led into three steps

1. Interpolation of $\mathbf{U}$ on the quadrature points. This is the result of the following matrix-matrix product
$$\begin{pmatrix} \varphi_1(\hat{\mathbf{x}}_1) & \dots & \varphi_{n_{dof}}(\hat{\mathbf{x}}_1) \\ \vdots & & \vdots \\ \varphi_1(\hat{\mathbf{x}}_{n_{qf}}) & \dots & \varphi_{n_{dof}}(\hat{\mathbf{x}}_{n_{qf}}) \end{pmatrix} \times \mathbf{U},$$
provided $\mathbf{U}$ is ordered such as having n_{var} columns and n_{dof} lines.

2. A nonlinear step, in which $\mathbf{F}(\mathbf{U})$ is computed on all the quadrature points.

3. Projection of $\mathbf{F}(\mathbf{U})$ on the degrees of freedom. If $\mathbf{F}(\mathbf{U})$ is ordered such that having n_{var} columns and $n_{qf} \times dim$ rows, then this step can also be computed as a matrix-matrix product
$$P \times [\mathbf{F}(\mathbf{U})],$$
with
$$P_{i_{dof}, i_{qf} \times dim + i_{dim}} = \omega_{i_{qf}} \left[DT_K(\hat{\mathbf{x}}_{i_{qf}})^{-1} \nabla \hat{\varphi}_{i_{dof}}(\hat{\mathbf{x}}_{i_{qf}})\right]_{i_{dim}} \left|\det DT_K(\hat{\mathbf{x}}_{i_{qf}})\right|,$$
and where $[\mathbf{F}(\mathbf{U})]$ was ordered such that its index of row is $i_{qf} \times dim + i_{dim}$ and its index of column is i_{var}.

From an implementation point of view, the first and third step may be implemented by resorting to BLAS library. In the second step, the evaluation of the flux shall be called n_{qf} times, which can be accelerated using vectorization.

These three steps can actually be encountered in all the integrals of (3.3) with

- A linear step in which $\mathbf{U}$, $\nabla\mathbf{U}$ or $r_h\left(\llbracket \mathbf{U} \rrbracket\right)$ is computed at the quadrature points of the cell or the face.
- A nonlinear step, in which the physical hyperbolic flux, the numerical flux or the diffusive flux is computed at each quadrature point on what was computed in the first step.
- A linear step in which the flux on the quadrature points is projected on the degrees of freedom.

An implementation can therefore be proposed in which all the geometrical data are precomputed. Note that this implementation is even more attractive when considering discontinuous Galerkin methods because the unknowns are already stored contiguously by cell, so that the first and third step can be computed by using pointer references, thus avoiding memory copies.

3.3 Some results

3.3.1 Validation results

3.3.1.1 A scalar result

The aim of this test is to quickly test the convergence order on the following one dimensional scalar advection-diffusion problem

$$\partial_t u + \beta \partial_x u = \mu \partial_{xx} u,$$

on $[0,1]$ with $\beta = 3.2$, $\mu = 241$. Dirichlet boundary conditions are imposed on the left and on the right with $u(0) = u_0 = 2$ and $u(1) = u_1 = 1$. If $\beta \neq 0$, the analytical solution is

$$\frac{(u_1 - u_0)\exp(\beta x/\mu) + u_0 \exp(\beta/\mu) - u_1}{\exp(\beta/\mu) - 1}.$$

Convergence results are shown in Figure 3.7, and prove that for an approximation degree p, the convergence order is $p+1$.

3.3.1.2 Laplace equation on a ring

This case aims at assessing the usage of curved meshes with the Laplace equation. The Laplace equation is solved on a ring delimited by the circles of radius R_{int} and R_{ext}, with the following Dirichlet boundary conditions imposed: u_{ext} on the circle $r = R_{ext}$ and u_{int} on the circle $r = R_{int}$. The analytical solution in polar coordinates is

$$\frac{(u_{ext} - u_{int})\log(r) + u_{int}\log(R_{ext}) - u_{ext}\log(R_{int})}{\log\left(\dfrac{R_{ext}}{R_{int}}\right)}.$$

Convergence curves for approximation degree 0, 1 and 2 in Figure 3.8. The convergence order obtained is respectively 1, 2 and 2 on a Q_1 mesh, and 1, 2 and 3 on a Q_2 mesh. This enlightens the necessity to have a mesh curved with degree p for an approximation degree p.

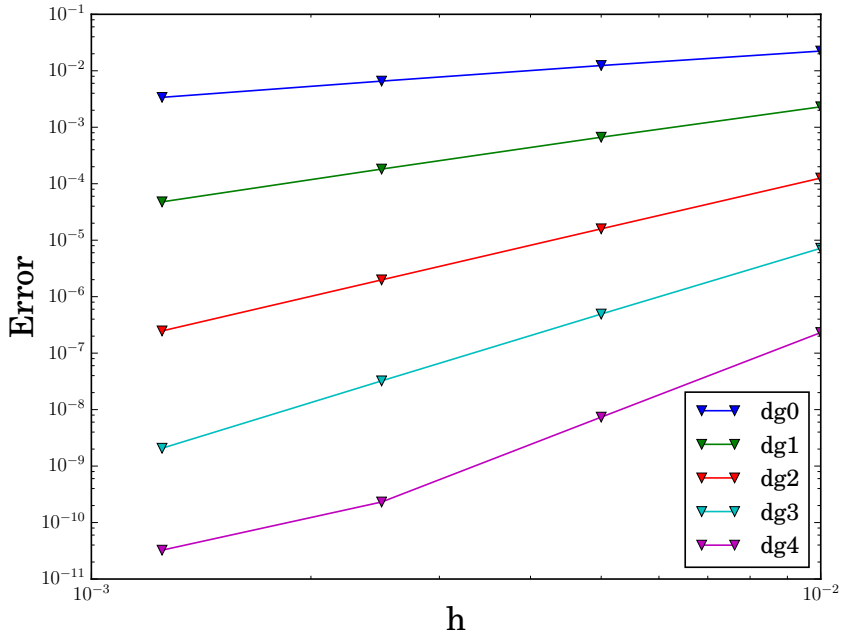


Figure 3.7: Convergence order for the one dimensional scalar advection-diffusion test case.

3.3.1.3 Poiseuille flow

In order to assess the high order for the stationary compressible Navier-Stokes system, a compressible Poiseuille-Haagen flow is tested. The domain is $[0,1]^2$. Periodic boundary conditions are imposed in the x direction, and isothermal wall boundary conditions are imposed at $y = 0$ and $y = 1$, where the imposed temperature is uniform, equal to T_w. The aim is to have a parabolic profile for the velocity

$$\mathbf{u}(x,y) = 4u_c y(1-y)\mathbf{e}_x$$

For achieving this, the following force is imposed

$$\mathbf{F}_v = 8\mu u_c \mathbf{e}_x$$

as a right hand side of the momentum equation. The work of the volumic force is imposed as the right hand side of the energy equation

$$\mathbf{F}_v \cdot \mathbf{u} = 32u_c^2 \mu y(1-y).$$

The stationarity for the energy equation leads to a second order equation for the temperature, which can be integrated. The flow is characterized by its Prandtl number Pr and its Mach number Ma, and the temperature is then

$$T(x,y) = T_w \left(1 + \frac{16Pr\,Ma^2\,(\gamma-1)}{3}\, y(1-y)\left(y^2 - y + \frac{1}{2}\right)\right)$$

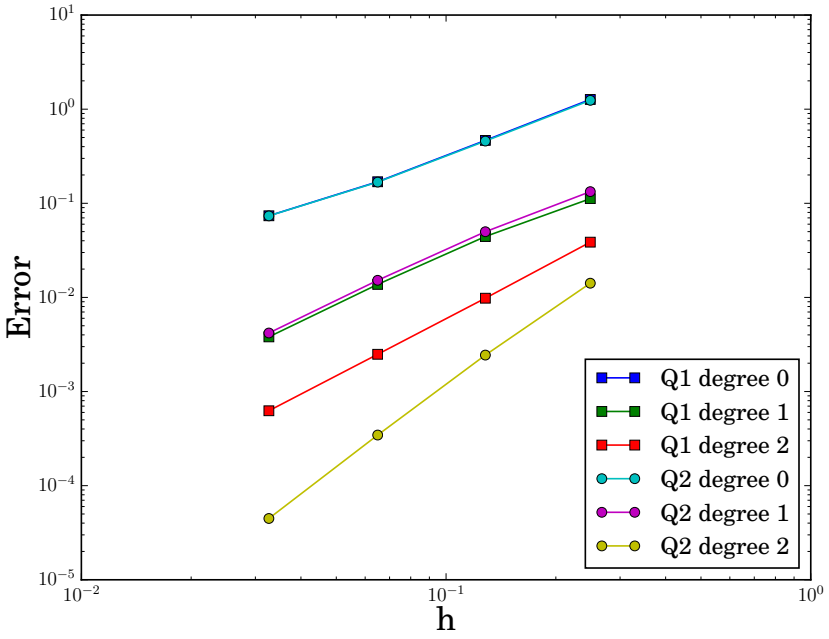


Figure 3.8: Convergence curves for the solution of the Laplace equation on a ring. The test is launched with approximation degrees 0, 1 and 2, on Q_1 (straight) and Q_2 (curved) meshes. For approximation degree 2, the optimal order of convergence is obtained only on Q_2 meshes.

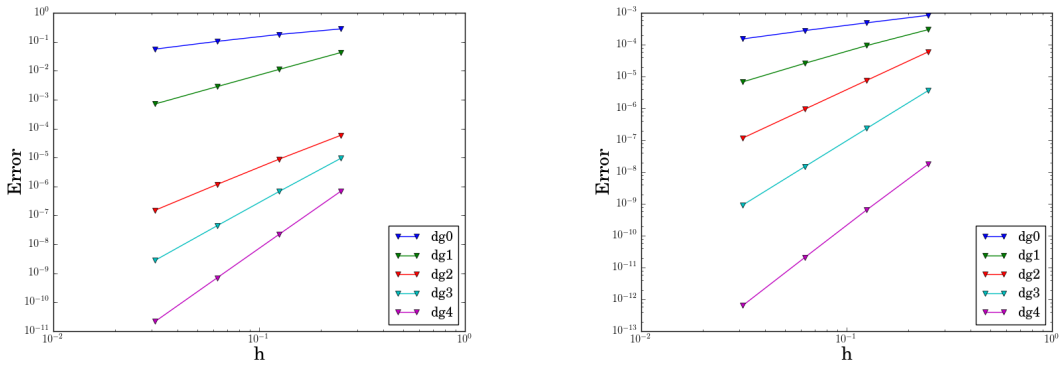


Figure 3.9: Convergence curve for the Poiseuille flow in a channel, on the velocity (left) and temperature (right).

If we denote by ρ_{ref} a reference density, then the pressure is uniform, equal to $(\gamma-1)c_v\rho_{ref}T_w$, and the density is equal to $\rho_{ref}T_w/T$. The test is launched on a set of meshes of size 4×4, 8×8, 16×16 and 32×32, with approximation degree of 0,1,2,3 and 4. Convergence curves on the velocity and temperature are shown in Figure 3.9. The optimal order is obtained for all the approximation degrees.

3.3.1.4 Taylor-Green vortex

This test case is classical for the assessment of turbulence decay in a tri-periodic box. Here, we consider the version of [20], with uniform density.

The geometry is the $[-\pi L_0, \pi L_0]^3$ box, where L_0 is a given length scale, with periodic boundary conditions in all the directions. The initial flow is characterized by its Reynolds, Mach and Prandtl number. Moreover, three adimensioning parameters are chosen, a density ρ_0, a velocity V_0 and a temperature T_0. In the computations, the following data are used: $V_0 = 1/\pi$, $L_0 = 1/\pi$, $\rho_0 = 1$ and $T_0 = 1$. The viscosity is obtained from the Reynolds number

$$\mu = \frac{\rho_0 V_0 L_0}{Re},$$

the reference pressure P_0 is obtained from the Mach number

$$P_0 = \frac{\rho_0 V_0^2}{\gamma Ma^2},$$

the heat capacity is obtained by the Mach number and the reference temperature

$$c_v = \frac{V_0^2}{Ma^2\gamma(\gamma-1)T_0},$$

and the thermal diffusion coefficient is defined from the Prandtl number

$$\kappa = \frac{\mu\gamma c_v}{Pr}.$$

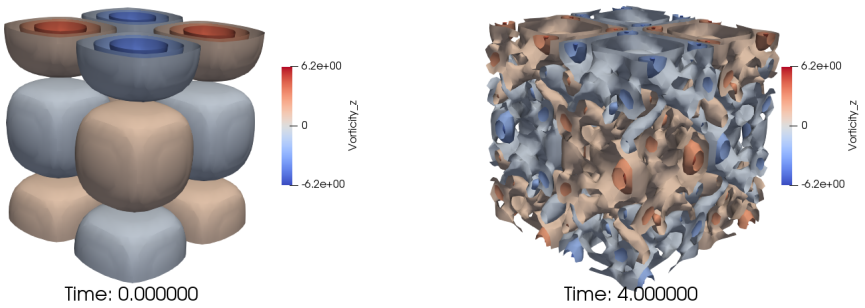


Figure 3.10: Volume rendering of the z–component of the vorticity at time $t = 0$ (left) and $t = 4$ (right) for the Taylor-Green vortex at Reynolds $Re = 500$.

The initial density is uniform, equal to ρ_0, the initial velocity is

$$\mathbf{u}(x,y,z) = \begin{cases} V_0 \sin\left(\dfrac{x}{L_0}\right) \cos\left(\dfrac{y}{L_0}\right) \cos\left(\dfrac{z}{L_0}\right) \\ -V_0 \cos\left(\dfrac{x}{L_0}\right) \sin\left(\dfrac{y}{L_0}\right) \cos\left(\dfrac{z}{L_0}\right) \\ 0 \end{cases},$$

and the initial pressure is

$$P = P_0 + \frac{\rho_0 V_0^2}{16}\left(\cos\left(\frac{2x}{L_0}\right) + \cos\left(\frac{2y}{L_0}\right)\right)\left(\cos\left(\frac{2z}{L_0}\right) + 2\right).$$

The test is launched with $Re = 500$, $Pr = 0.71$ and $Ma = 0.1$. The volume rendering of the isovalues of the z–component of the vorticity are shown in Figure 3.10 at time $t = 0$ and $t = 4$. They show that large structures are transmitted to small scales and last, dissipated. The time evolution of the volumic average of the kinetic energy and enstrophy are shown in Figure 3.11, and compared with the reference solution of [20]. The DG^2 solution is not able to render the peak of enstrophy, whereas the DG^3 solution matches well with the reference solution. These results are in agreement with the discontinuous Galerkin results of [20].

3.3.2 Performances

We are now interested in the performances of the library. We compare the performances with and without storing the geometry, then do some scalability tests.

3.3.2.1 Optimized and non-optimized comparisons

In subsection 3.2.6, two algorithmic versions were proposed for the discontinuous Galerkin method: one where the geometrical data are recomputed, and the other one in which interpolation and projection operators are stored. In this section, the two versions are compared in dimension 2 and 3, and with different approximation degree.

In two dimensions, the test case is a two dimensional version of the Taylor-Green vortex, with periodic boundary conditions in all the directions, and with a 20×20 quadrangular mesh. In three dimensions, the test case is the Taylor-Green vortex with a $3 \times 3 \times 3$ mesh.

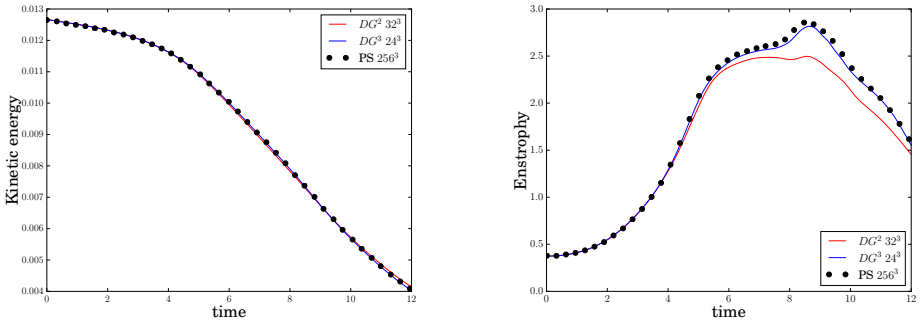


Figure 3.11: Results obtained with the Taylor-Green Vortex at Reynolds 500. Time evolution of the volumic average kinetic energy (left) and time evolution of the volumic average enstrophy (right). Results obtained with a DG^3 approximation on a 24^3 mesh and with a DG^2 approximation on a 32^3 mesh are compared with a reference solution obtained with a pseudo-spectral code (PS) obtained in [20].

For the two dimensional and three dimensional test case, and for a given approximation degree, the optimized and non optimized versions of the code are used with the same number of iterations. The code is instrumented so as to compute the time spent in the residual computation. For each dimension and approximation degree, the test is launched 100 times and the residual computation time is averaged. The ratio between the optimized version and the non optimized version is shown in Figure 3.12. The version with geometry storing is between 4 and 25 times faster than the version without storing the geometry. Note however that the version with geometry storing implies more data movement, and a thorough study including a cache memory usage should be done for obtaining a fair comparison.

3.3.2.2 Parallel performances

In this section, some performance results obtained in parallel are reported

The first test is a scalability test done within the CEMRACS project Colargol [6], which aimed at comparing performances of AeroSol with the discontinuous Galerkin code developed at Onera. The test is a two dimensional vortex advection, but performed with a three dimensional mesh (one layer of hexaedra). This is a weak scalability test, in which each MPI process has a piece of 10×10 cells. Results are shown in Figure 3.13. A good scalability is observed in general (over 60%). For lower orders, the scalability is weaker, showing that when the number of process increases, the computational intensity is not high enough for overlapping the overhead of the communications (induced by the latency of the network, for example). The scalability is nearly perfect at order 5 (DG^4).

The second test is a load balancing test done for one computation of the Impact-ae project (see next section for details about this project). The geometry is typical of what can be challenging in aerodynamics: the computation includes boundary layers and channel-like geometries which can be meshed with hexaedra, but also some parts that can hardly be meshed with hexaedra. In this case, tetrahedral meshes can be used, and pyramids and prisms are also needed as transitional elements between hexaedra and tetrahedra. This induces some strong disparities in the computational cost which is hard to estimate: for example, hexaedra have much more degrees of freedom, but the number of quadrature points grows faster for

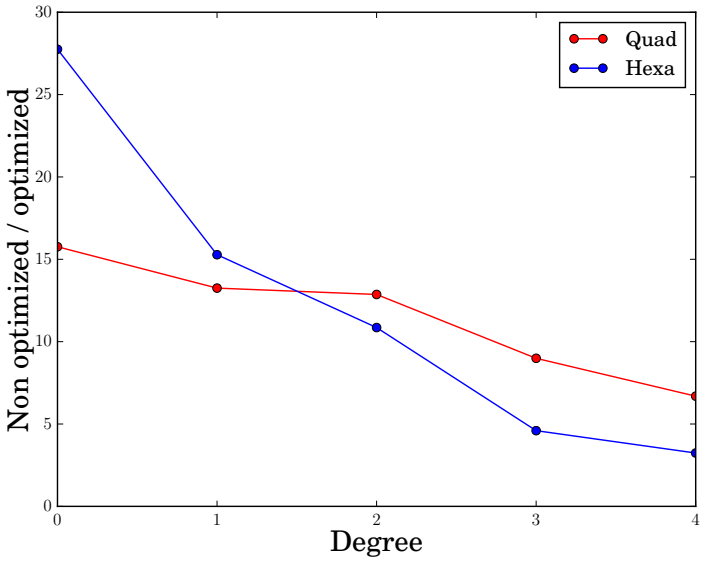


Figure 3.12: Ratio of the time spent in the residual computation between the non optimized and optimized version, on a quadrangular mesh in 2d and on an hexaedral mesh in 3d.

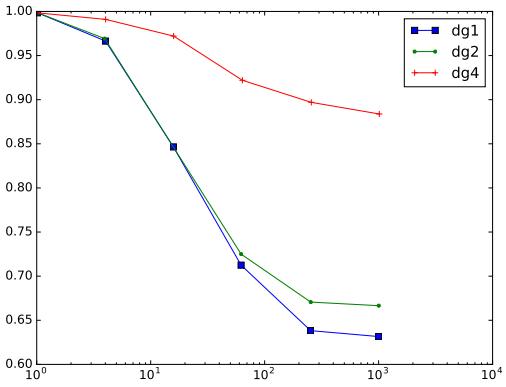


Figure 3.13: Results for the weak scalability test. Each MPI process owns 10×10 cells, and the test is performed with 1 to 1024 processors. Note that the Y axis begins at 60% efficiency.

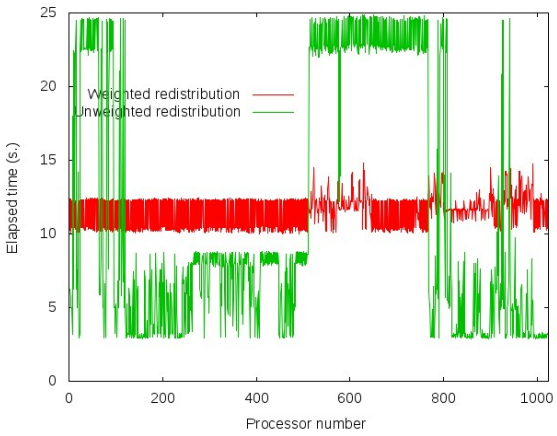


Figure 3.14: Computational cost without (green) and with (red) load balancing on 1024 process. Without load balancing, the computational cost of each process is between 4 and 25, whereas with load balancing, the computational cost is between 10 and 12.

tetrahedra. As a consequence, if the cells are distributed regardless their shape, a MPI process which includes a lot of hexaedra will have a higher workload than a process which has a lot of tetrahedra. This heterogeneity problem is addressed in two steps. First, a computation is done, in which the work distribution is made uniformly. The time for the cell loop t is registered. Then a linear model is guessed on the computational cost:

$$t = N_{Tet}t_{Tet} + N_{Hex}t_{Hex} + N_{Pyr}t_{Pyr} + N_{Pri}t_{Pri},$$

where N_{Tet}, N_{Hex}, N_{Pyr} and N_{Pri} is the local number of tetrahedra, hexaedra, pyramids, and prisms, and $t_\star$ is the computational time on one shape of type $\star$. A linear regression is done, which allows to compute the $t_\star$. The following computing time for each shape were found

t_{Tet}	t_{Pyr}	t_{Pri}	t_{Hex}
3.08	8.87	9.99	29.31

These times were then used as weight on the cells during a second mesh partitioning step. In Figure 3.14, the computational cost by MPI processor is shown, with, and without the weighted mesh partitioning. The weighted partitioning shows a much better load balance than the one without weight.

3.4 Results within the IMPACT-AE project

3.4.1 Presentation of the project

The library was used within the IMPACT-AE project[1], a European project I was managing for Pau University, which was funded at the 2010 call of the 7th framework program. The name of the project is an acronym meaning Intelligent design Methodologies for low PollutAnt CombusTors for Aero-Engines.

[1] https://cordis.europa.eu/project/id/265586

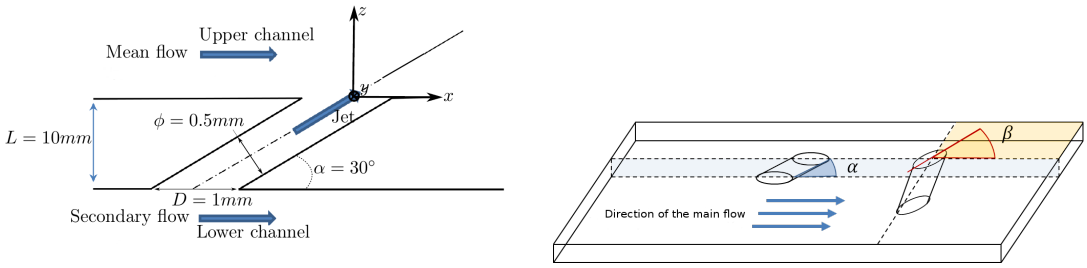


Figure 3.15: Left: jet in cross flow configuration without skidding. A cylindrical hole links the lower and upper channel. Each of the channel includes a flow (mean and secondary flow) aligned with the hole direction. Right: the most right hole has a skidding: the mean and secondary flows are still aligned, but the hole direction is no more aligned with these flows with an angle β.

Within this project, Pau University was in charge of generating a DNS database of a jet in cross flow, a flow configuration typical of effusion cooling of combustion chamber walls. For ensuring the least deterioration of the walls of a combustion chamber, their temperature should be kept under a given threshold. Isolation of the walls from the heat of the combustion is achieved by effusion cooling. Effusion cooling consists in piercing small holes in combustion chamber walls; the pressure difference between inside and outside the combustion chamber induces a suction inside the combustion chamber, leading to a jet of cool air isolating the wall. Flaring and pattern of the holes have a major role in the efficiency of effusion cooling. The configuration retained within the IMPACT-AE project was a single jet in cross flow with a moderate Reynolds number; the first geometry was with a hole aligned with the mean flow, and the second with a 90° skidding between the hole direction and the mean flow. These configurations are illustrated in Figure 3.15. Two flow conditions were considered: the first with the same temperature in the upper and lower channel, and the second one more representative of the effusion cooling, with a temperature of $300K$ in the lower channel and a temperature of $1200K$ in the upper channel.

3.4.2 Inlet boundary conditions

In the flow we want to compute, the upper and lower channel are turbulent. Reaching a fully developed turbulent flow may require a large upstream flow, which would strongly increase the computational cost. For avoiding this, one can resort to a precursor computation [94] or a recycling strategy [88]. Still, a precursor computation is costly from a computational and storage point of view, and recycling strategies are better adapted to simpler geometries.

We finally choose the Synthetic Eddy Method (SEM) of [108], an extension of the method proposed by [72]. This method consists in injecting shape function that are close to Navier-Stokes solutions, and which statistics match with the statistics that should be imposed. Details on the generation of a turbulent signal were the aim of one deliverable of the project [114], and can be found in the PhD thesis of Simon Delmas [34, Chapter 4.2.2].

We focus now on how the imposed Reynolds stress to be imposed as inlet boundary condition was determined. The inlet Reynolds stress is generated as follows. First, the axial velocity and the R_{xx} component of the Reynolds stress are measured by using the LDV

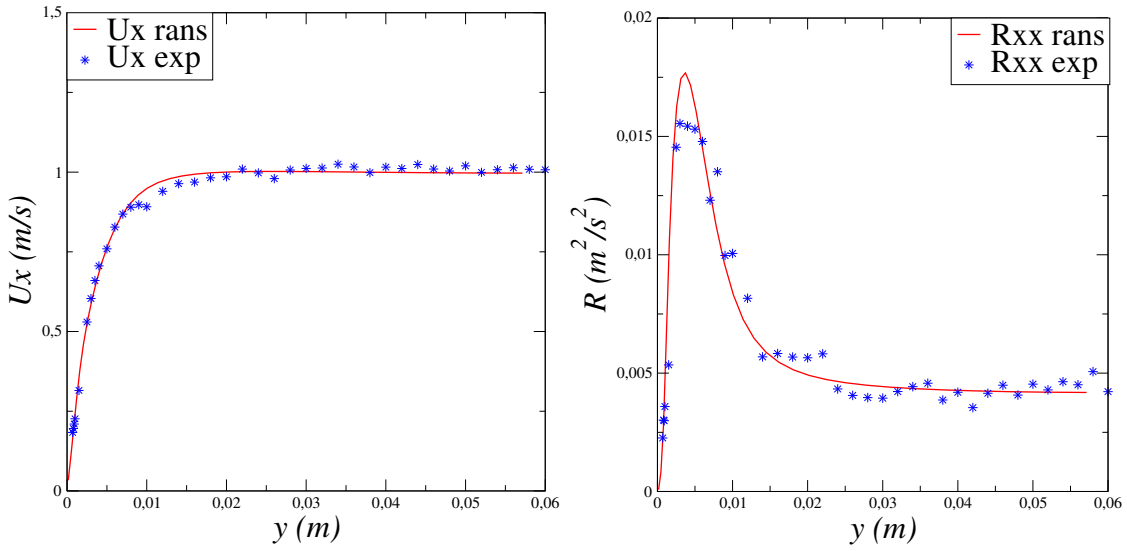


Figure 3.16: Comparison of the axial velocity and the first component of the Reynolds number in the channel obtained with the SATURN code and the LDV measurements done with the MAVERIC experimental bench.

method on the MAVERIC experimental bench on several distances from the wall. Then a RANS computation is led with the SATURN code. The computational parameters (e.g. the turbulent kinetic energy k) are iteratively changed by targeting the axial velocity shape, the turbulent kinetic energy on the axis and the position of the peak of energy with respect to the wall. After convergence of this iterative procedure, the results shown in Figure 3.16 are obtained. The other components of the Reynolds stress are then given by the SATURN computations. With this method, a complete inlet Reynolds stress was obtained, which is such that the axial velocity and the R_{xx} component of the Reynolds stress are matching with the experimental data.

3.4.3 Computational strategy

The mesh is chosen so as to ensure that the typical mesh size is around $5y^+$ (lower than in the turbulent channel computations [77, 26]). This results in a mesh with around 30000 cells. With a DG^2 approximation, this gives 800 000 approximation points, and so 4 millions unknowns. For the computation of the different cases, a grant of 8 millions of computing hours was obtained on the BlueGene /Q cluster TURING. Due to the particular structure of this machine (slow cores with low consumption, and topology of the nodes), the computation had to be performed on at least 1024 cores, which was the upper limit in our case, for having enough cells per process for ensuring overlap of communication by computation. Then the computation is led with the following steps

1. **Mesh partitioning.** This stage consists in reading the mesh and partitioning it for balancing the load between the different cores. This step is costly in memory, so that it must be done on 2048 cores (but still with 1024 MPI process) for increasing the available memory on each core. An example of mesh partitioned is shown in Figure 3.17.

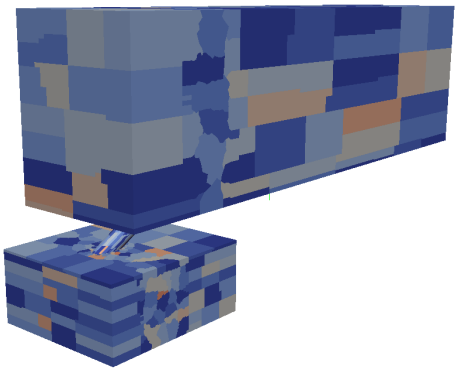


Figure 3.17: Partitioning of the mesh used for the case without giration on 1024 nodes.

2. **Initialization of the computation.** The domain is initialized as follows: in the whole domain, the density is set to the reference density. In the upper channel, a velocity uniform in x and y, and following the measured profile in z is set. The same kind of initialization is done in the bottom channel. The velocity is set to 0 in the hole. The pressure is set uniform, equal to the reference pressure in each channel. It follows a linear law in the hole.

3. **One convective time with a DG^1 discretization.** A first convective time t_c is made, for eliminating the transitory phenomena induced by initial conditions. It is made with a DG^1 discretization, for dissipating quickly the spurious phenomena, and also for lowering the computational cost of this step. A jet is observed at the end of this computation, but as the numerical method is very dissipative, the jet follows a trajectory close of the wall, which does not match with experiments, as can be observed in Figure 3.18 (left).

4. **One high order convective time step.** Once the transitory effects have been evacuated, the simulation is restarted from the solution obtained in the previous step, and a simulation is led with a third order scheme. As the numerical scheme is less dissipative, the jet is expanding in the whole upper channel, see Figure 3.18 (right).

5. **One convective time with the SEM.** The aim of this step is to leave the turbulence develop in both of the channels.

6. **Statistics collection.** In the whole domain, probes are positioned in order to register all the instantaneous variables. As for the LDV measures, a spectrum and the probability density function can be computed. The positions of the probes are detailed in the Figure 3.22.

The numerical strategy explained was used for the isothermal configuration. For the non-isothermal configuration, the solution is restarted after step 3. Then the top channel is progressively heated up for reaching $1200K$, see Figure 3.19 (left). This step is performed with a DG^1 numerical scheme. Then the steps 4, 5 and 6 are done. One high order result after heating is shown in Figure 3.19 (right).

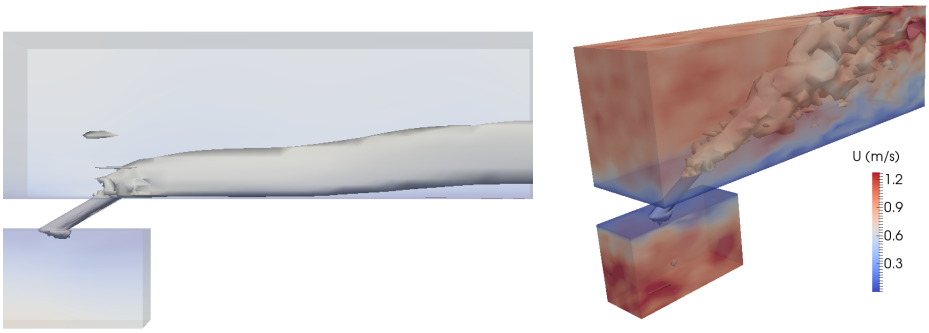


Figure 3.18: Results obtained with the jet in cross flow configuration. **Left**: isovelocities obtained for DG^1 computations. **Right**: result obtained with the DG^2 numerical scheme.

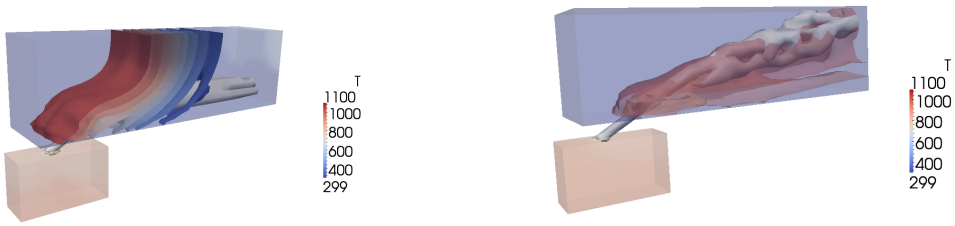


Figure 3.19: Jet in cross flow with a hot upper channel. **Left**: heating of the upper channel with a DG^1 scheme. **Right**: flow obtained after switching to high order (DG^2).

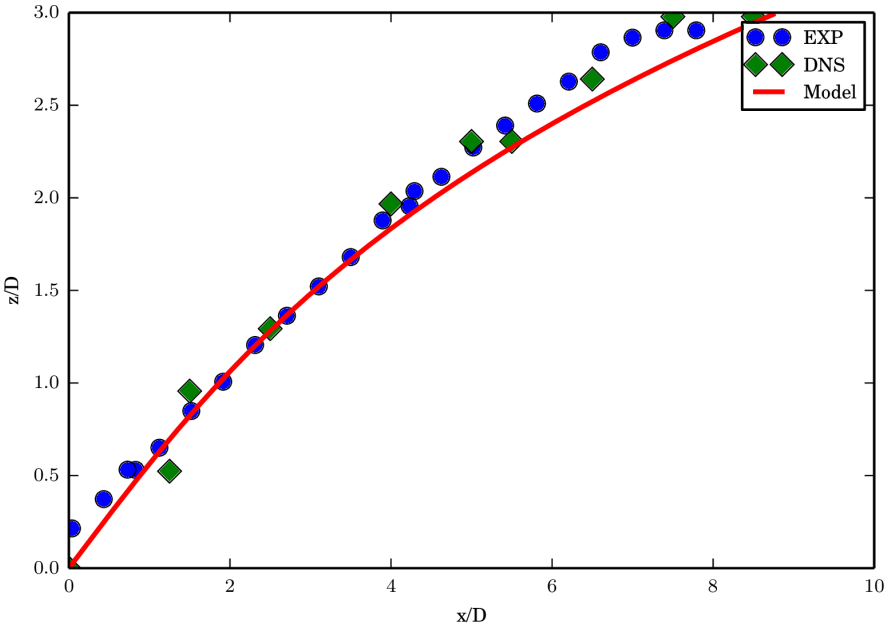


Figure 3.20: Jet trajectory in the case without skidding. The obtained position of the jet is compared with experimental results and with the model of [90].

3.4.4 Results

We give here just some results, all the other results can be found in [115, 116]. In Figure 3.20, the trajectory of the isothermal jet without skidding is compared with experimental results and with a model developed in [90]. A good matching with the model and with the experimental result is observed. In Figure 3.22, the position of the probes in the case without jet skidding is shown, and the high order statistics are shown for enlightning that the flow is statistically converged. In Figure 3.21, a visualization of a non isothermal jet in cross flow with skidding is shown.

3.5 Preliminary results on tasked based algorithms with a runtime scheduler

As explained in Figure 3.1, the aim of the AeroSol library is to address heterogeneous architectures (NUMA effects, host/accelerators configurations) with the runtime scheduler StarPU. For the moment, the library is parallelized with distributed memory with MPI. The library PaMPA is in charge of the memory distribution, overlaps computing, and point-to-point communications. Using a runtime scheduler may be very intrusive in a code. Moreover addressing parallelism (even with distributed memory or with coarse grain OpenMP) requires to partition the mesh and to manage overlaps. This may be time consuming as far as the development cycles are concerned, even more when addressing unstructured hybrid meshes. This is why we decided to begin to test the runtime StarPU on a structured mock-up code

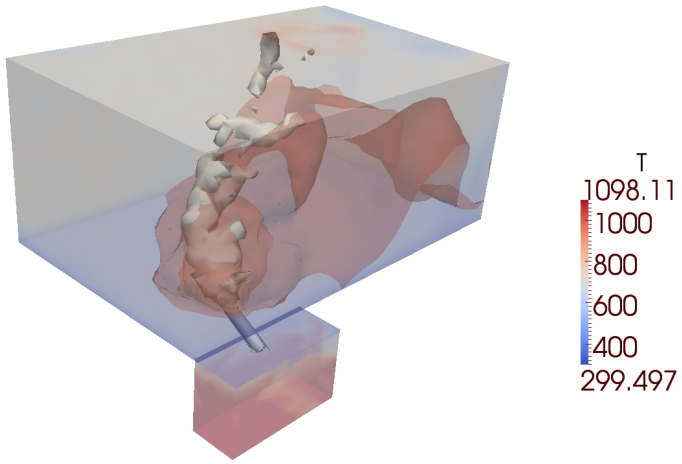


Figure 3.21: Non isothermal jet in cross flow with a jet skidding. The jet is identified as an isovelocity in grey, surrounded with a red isotemperature at $1100K$.

called HODINS. This work was done in collaboration with the authors of [44].

3.5.1 Task based programming

As far as computational methods based on meshes are concerned, an usual way to consider parallelisation consists in working on a partitioned mesh. The computation of the residual on the interior of each part of the mesh can then be performed in parallel. The remaining part of the computation consists in communicating or copying the data on borders of the parts, and computing the residual there. The execution order can be optimized for ensuring that with a sufficient computational intensity, the communications or copies can be overlapped by computation. Also, the code defines explicitly in which order the different parts of the code are executed (this is called "static scheduling"), and the parts of the mesh are mapped on a given core of a node.

However imbalances may be observed at execution. This may come from the computation itself, which may be heterogeneous (see Figure 3.14 of subsubsection 3.3.2.2), but also on the mapping: for example, a MPI communication is faster between two cores of the same socket than between two computational nodes if these nodes are far in the topology of the cluster network. In the same manner, a computation is usually faster on an accelerator, but the memory transfer overhead between the master node and its accelerator should also be taken into account. All of this can be optimized if the memory movements pattern of the code is known, and the exact characteristics of the machine are known. However, this problem becomes more and more complex because of the higher complexity of the computational nodes (NUMA effects, accelerators). Moreover, this optimization should be performed on each specific architecture. Last, oversubscription or undersubscription of computational cores may occur for example in postprocessing or I/O phases.

An alternative of this static scheduling is given by task based programming: instead of mapping work explicitly on computational cores and explicitly give the execution order and mapping, the program is expressed in term of *tasks*. For keeping a consistent execution, and

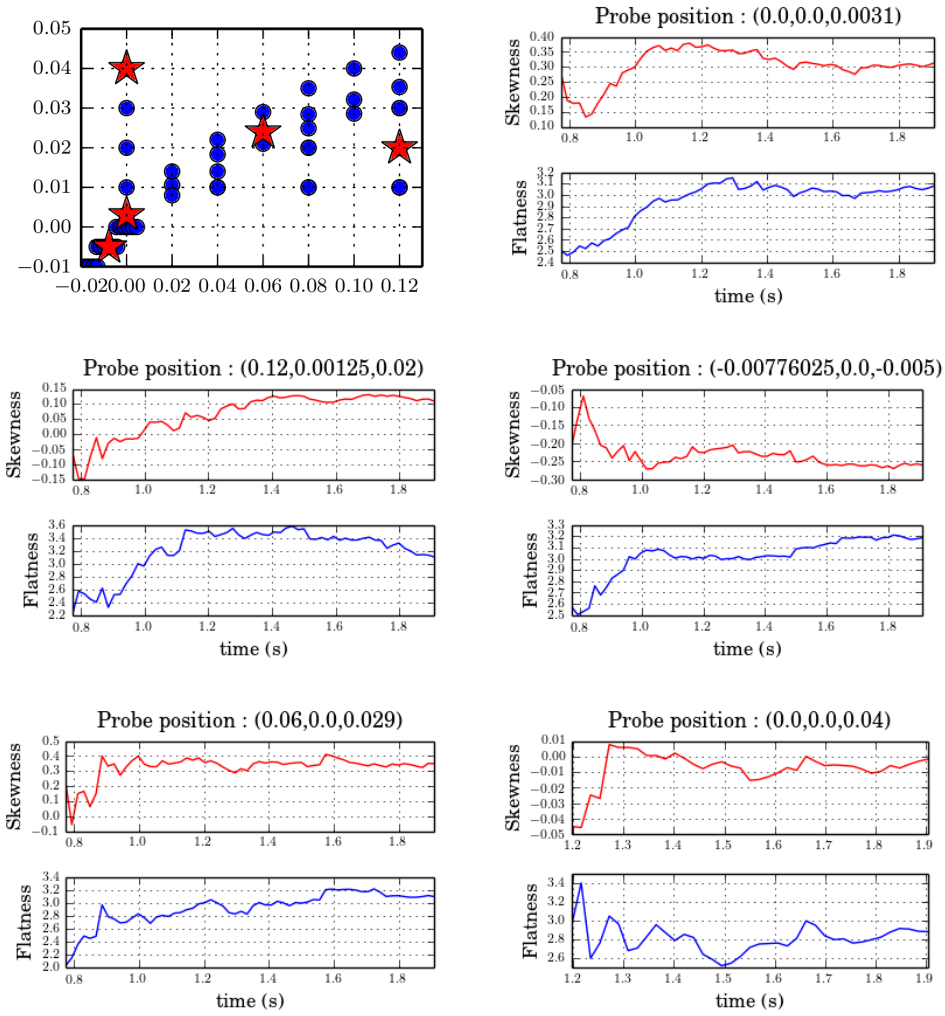


Figure 3.22: **Top Left:** position of the probes in the computational domain, depicted with blue bullets and red stars. **Other figures:** The red stars depict the probes which statistics are shown in the remaining part of this figure. On these pictures, the skewness and flatness of the time distribution of the velocity are shown. These statistics are nearly flat at time 2, which prove that the flow is statistically converged.

avoiding concurrent memory access, dependencies on the tasks must be expressed. Tasks and tasks dependencies give a *direct acyclic graph* that is given to a *scheduler*. The scheduler is then in charge of mapping the work on the available resources. Depending on the schedulers, and on the a priori knowledge of the programmer, the scheduling may be optimized. Resorting to a scheduler automatically reduces the oversubscribing or undersubscribing of resources. Over the last ten years, task programming has gained a strong popularity in the high performance community: it is for example the basic concept of Intel TBB, and is also a more and more important concept in OpenMP.

3.5.2 The runtime scheduler StarPU

The runtime StarPU has been developed for about ten years in the Runtime and then Storm Inria teams. In StarPU, a *task* is defined from a *codelet* which includes computing *kernels*. These *codelets* are applied to a set of *data handles*. More specifically:

- **data handles** are the memory managers.
- **kernels** are the functions that will be executed on a dedicated architecture. This is where the classical code, with loops, conditions, flux computation are written.
- **codelets** are a set *kernels* which may be applied to a set of data handles with explicit Read, Write or Read and Write properties. Several kernels may be defined: for example if the codelet is to be executed on both a CPU and a GPU, then the codelet should at least include one *kernel* written in C, and another written in CUDA or OpenCL.

Data dependencies can be given explicitly. It can also be given implicitly by defining Read Only, Write Only or Read-and-Write properties on data handles. From this, StarPU is able to build the graph of tasks, and to schedule it on available computational resources. Also, StarPU is able to execute some work on accelerators provided the appropriate kernel was defined. Data transfers between a node and its accelerator are done by StarPU.

3.5.3 Implementation

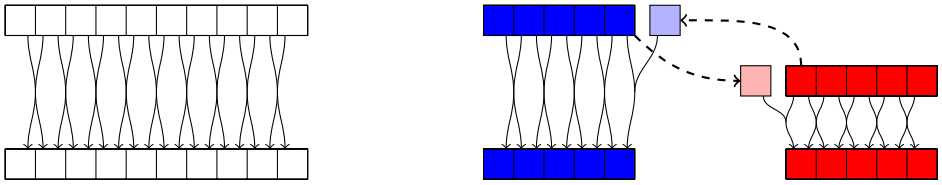
In this section, we give an overview on what are the codelets, tasks and pointers defined for a finite volume code. A classical way of parallelizing the computation of the residual of a one dimensional finite volume scheme is shown in Figure 3.23.

For expressing the algorithm with StarPU, the mesh is partitioned into N parts. We define the **data handles** as

- u_i $(1 \leqslant i \leqslant N)$, the data of the vector u on the part i,
- f_i $(1 \leqslant i \leqslant N)$, the data of the residual vector f on the part i,
- $u_i^{Ovp,j}$, the data needed for computing the residual at the border of the part i which comes from the data of the jth part.

Then the following **codelets** are defined

- **CopyOverlaps.** This codelet takes a local u_j as a read only data handle and copies the needed values in a write only $u_\star^{Ovp,j}$,



Sequential algorithm

Partitioned algorithm with overlaps

Figure 3.23: Sequential and partitioned algorithm with overlaps on a one dimensional 10 cells mesh. The curved solid arrows represent a flux computation, and dashed lines represent a copy. Partitions are represented with colors (blue for one partition and red for the other one). Additional overlaps buffers are represented with light colors.

- **InternalResidual.** This codelet takes a local u_i as a read only data handle and computes the internal residual in f_i,
- **BorderResidual.** This codelet takes a local u_i as a read only data handle, and all the $u_i^{Ovp,\star}$ and computes the border residual in the write only f_i,
- **update.** This codelet takes a local u_i as a read and write data handle and computes the new u_i by using the read only f_i.

From the above codelet and data handles definitions, the tasks for computing the residual on a one dimensional mesh with 10 cells is shown in Figure 3.24. Beyond the fact that the blue and red InternalResidual and BorderResidual can be executed in parallel, we can easily deduce the following (even if this is automatically taken in charge by StarPU):

- copyOverlaps and InternalResidual can be executed in parallel, because they do not write into the same data handles.
- InternalResidual and BorderResidual cannot be executed in parallel because they write in the same data handles.

Note that the choice of data handles is of paramount importance for ensuring a good potential of parallel execution. If we suppose that in Figure 3.23, u_i and $u_i^{Ovp,j}$ are a single data handle, this jeopardizes the parallel execution of copyOverlaps and InternalResidual, because u_i would then be R for InternalResidual, and W for copyOverlaps. Once implemented, the execution graph output by StarPU for a finite volume code with a mesh partitioned into two pieces is depicted in Figure 3.25.

Last, we remark that using a runtime scheduler is not free. Before launching a task, some data management and task scheduling is performed, leading to a time overhead each time a task is launched. Scalability can therefore be jeopardized by this overhead if the launching task time overhead is not enough smaller than the average duration of a task, see Figure 3.26. From this figure, we can state for example that for ensuring good scalability on a 2 dodeca-core Haswell Intel Xeon E5-2680, tasks should be larger than $64\mu s$ if only 12 cores are used, and larger than $256\mu s$ if the full node is used. It is therefore necessary to find a balance between having a lot of tasks, which gives more flexibility to the scheduler for parallel executions, and not having too small tasks, for ensuring scalability.

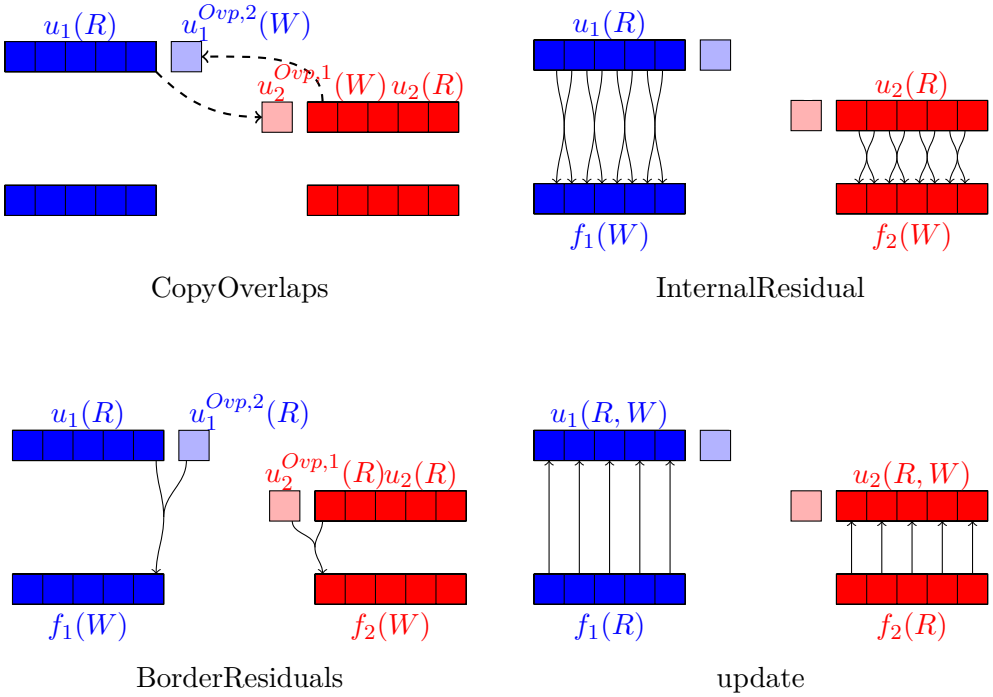


Figure 3.24: Tasks during the computation of one time step of the finite volume method with a one dimensional mesh with 10 cells and partitioned into two parts. Each picture illustrates a codelet applied to two sets of data handles, and so two tasks. Data handles with only R are Read-Only, with W are Write-Only, and with R and W are Read-and-Write.

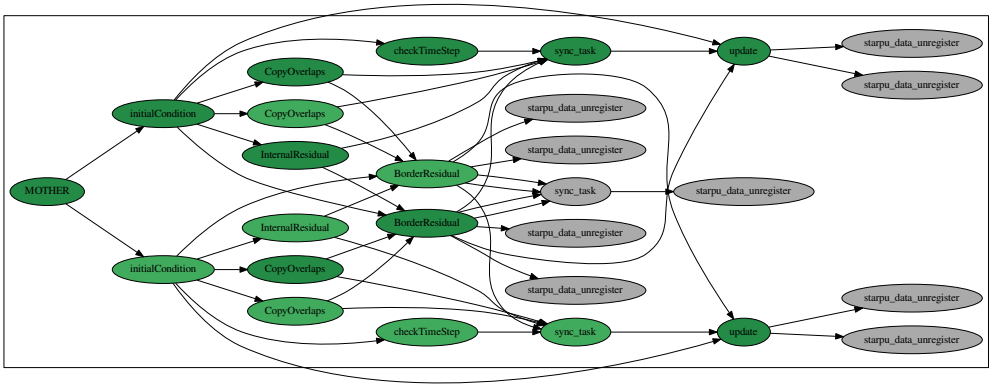


Figure 3.25: Task diagram ouput for one time step of a finite volume scheme with a mesh partitioned into two pieces.

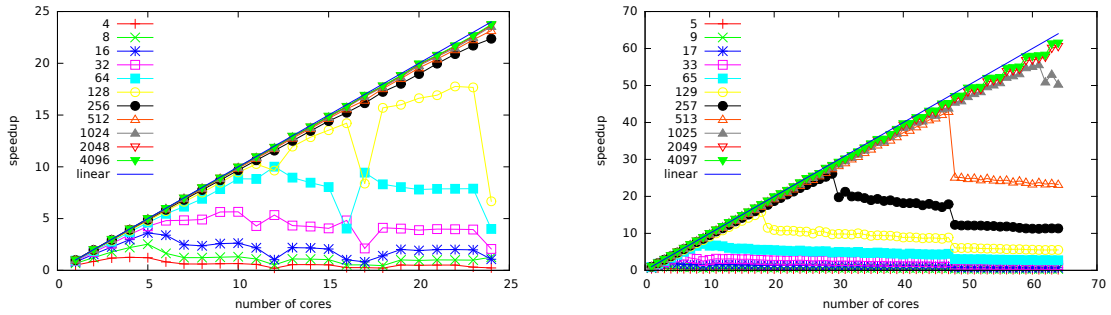


Figure 3.26: Task size overhead with the EAGER scheduler: scalability results obtained with duration of tasks varying between 4 and 4096μs on two different types of nodes. **Left**: 2 dodeca-core Haswell Intel Xeon E5-2680. **Right**: Xeon Phi KNL.

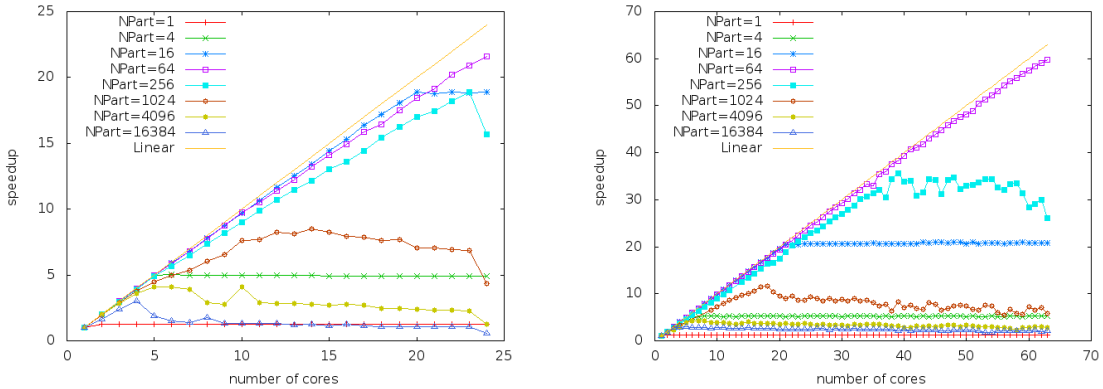


Figure 3.27: Strong scaling with the EAGER scheduler: scalability results obtained on two different architectures with a number of subdomain NPart varying from 4 to 128^2.

3.5.4 Results

We put here only two results, the others (including automatic offloading to accelerators) can be found in [44].

The first result we show is a strong scalability result obtained with a two dimensional test case. We consider a mesh of 1024×1024 cells, which is divided into $1, 2^2, \ldots 128^2$ parts. In Figure 3.27, results obtained on a 2 dodeca-core Haswell Intel Xeon E5-2680 and a Xeon Phi KNL are shown. The previous remark on task size overhead is very clear here, because an optimal sclability is obtained on the Haswell node with 64 parts: with 16 parts, there is too few parallelism, and with 256, the tasks are too short compared with the task size overhead. A result we did not expect but that can be considered as a strong benefit induced by task based programming is the ability to scale on 5 cores with a mesh partitioned into 4 parts on the Haswell node, and to scale on 20 cores on the KNL node with a mesh partitioned into 16 parts. This kind of surprising result could of course also be obtained with classical programming, but we consider the code development time much cheaper with task based programming.

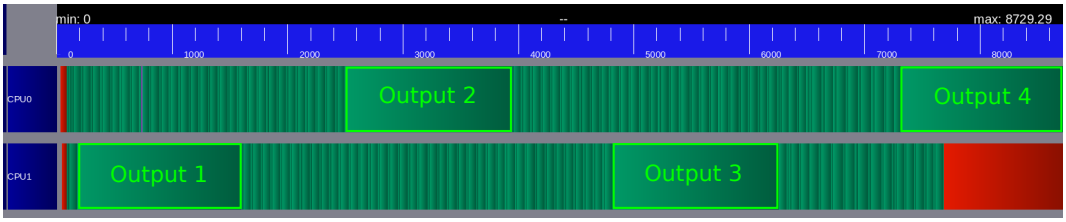


Figure 3.28: Gantt chart for 60 time iterations of the finite volume scheme on two cores with an output every 20 iterations.

The second result we show is an example of implementation of a two phase I/O algorithm: in order to avoid interaction between too many cores and the filesystem, which may burden the whole computation, it may be faster to reduce data on a smaller set of cores which are then in charge of performing I/O. Task based implementation on a single node with only one core in charge of output is detailed in [44], and an example of Gantt-chart obtained on two cores is shown in Figure 3.28. At the beginning of the computation, the two cores are involved in the computation. The Gantt-chart shows that when an output is done, a core is allocated to it. The computation keeps on on a single core until the two cores are available again.

3.6 Conclusion and prospects concerning the AeroSol library

In this chapter, the AeroSol library was presented. Its design and performances were exposed, and it was proven to be able to address challenging flows at large scale such as the ones that were computed within the Impact-AE project. Some preliminary results obtained with the runtime scheduler StarPU on a mock-up code were also exposed.

The generic features of AeroSol are:

- **It is high order**: It can be theoretically any order of accuracy, but the finite element basis, and quadrature formula are implemented for having up to a fifth order of accuracy.
- **It can work with hybrid and curvilinear meshes**: AeroSol can deal with up to fifth order conformal meshes composed of lines, triangles, quadrangles, tetrahedra, hexaedra, prism, and pyramids.
- **It can deal with continuous and discontinuous discretization**.

Strong emphasis have been put on:

Its development environment. For allowing a good collaborative work and a functional library, modern collaborative tools are used for developing our software. This includes the active use of a repository, the use of CMake for the compilation, the constant development of unitary and functional tests for all the parts of the library (using CTest), and the use of the continuous integration tool Jenkins for testing the different configurations of AeroSol and its dependencies.

Its genericity. A lot of classes are common to all the discretization types, for example classes concerning I/O, finite element functions, quadrature, geometry, time integration,

linear solver, models and data management with PAMPA. Adding simple features (e.g. models, numerical flux, finite element basis or quadrature formula) can be done easily by writing the class, testing it with the dedicated unitary test, and declaring its use in a single method of a single class of the code. The library is able to deal with generic nonlinear advection-diffusion-reaction problems, with possibly non conservative hyperbolic flux. The implemented models currently include Euler and Navier-Stokes systems, diphasic Euler system, linear Wave system, barotropic Euler system, Burgers equation, and linear advection-diffusion equation.

Its efficiency. Modularity is achieved by means of template abstraction for keeping good performances. Dedicated efficient implementation, based on the data locality of the discontinuous Galerkin method have been developed. As far as parallelism is concerned, point-to-point communications are used, external parallel linear solvers are used, and the HDF5 library allows for parallel I/O. The library was tested and ported on several clusters, and a good scalability was obtained above 1 000 cores.

The main challenges we want to address in the next years are:

Computing efficiently at large scale on heterogeneous architectures. If some of the current supercomputer technologies can be addressed with MPI, the trend of the HPC machines is to include accelerators; for example, two of the three main supercomputer in France include accelerators: the Irene cluster at CEA includes KNL, whereas one partition of the the Jean Zay cluster at IDRIS includes NVIDIA TESLA. Even if no accelerator is considered, the memory hierarchy on the computing nodes is more and more complex, making the optimal static scheduling a nearly intractable problem. In this case, the MPI technology should be supplemented with another one (MPI+X), but there is no agreement on who should be X: it may be OPENMP, OPENACC or any other runtime scheduler. Due to different boundary conditions, different cell shapes used in the mesh, the computations we are doing are essentially heterogeneous. We believe that runtime schedulers are an attractive solution for addressing these heterogeneous computations on heterogeneous machines [8, 27, 14]. The encouraging results we obtained in shared memory with finite volume schemes are currently being extended to high order discontinuous Galerkin methods by Sangeeth Simon, who began his postdoc on the 1st of July in our team. This work is also done in collaboration with Jonathan Jung and Matthieu Haefele.

Mitigating I/O cost at large scale. Beyond computing, what we expect from a CFD code is to produce usable (and useful) data: statistics if turbulent computations are performed, lift and drag coefficients, etc... For low scale computations, it is possible to store the full solution in time and to postpone the postprocessing, but the huge size of the full solution on large scale computations jeopardizes this strategy. Moreover, the ratio between computational capabilities and I/O bandwidth has been constantly increasing over the last years. This naturally leads to consider *in situ* or *in transit* postprocessing for minimizing I/O size. Several libraries have been developed over the last years: PDWFS[2], PDI[3], including its plugin for FLOWVR[4]. For the moment, we are

[2] https://github.com/cea-hpc/pdwfs
[3] https://pdi.julien-bigot.fr/master/index.html
[4] http://flowvr.sourceforge.net/

testing these tools, for choosing the best match with our problems, by also considering intrusiveness in our library. This work is done in collaboration with Matthieu Haefele.

Implementing scalable and efficient implicit methods at large scale. If the parallelization algorithm for explicit methods is easy, addressing implicit methods is much more complicated. As the compressible Navier-Stokes system is fully nonlinear, integrating this system implicitly requires to linearize it (with the Newton method for example), and then to solve a serie of linear systems. For the moment, the strategy in the AeroSol library consists in assembling the Jacobian, and solving the linear systems with an external library. Beyond the computational cost of the system resolution, the memory footprint of the matrix for high order methods becomes very large when dealing with three dimensional computations. A large memory bandwidth is also required when assembling the matrix. This naturally leads to considering Jacobian free methods. This kind of method was successfully used with Krylov methods[80]. However, Krylov methods are known for converging slowly if the system is ill–conditioned, which is the case for example when low Mach number flows are considered. For accelerating the convergence, preconditioning can be used. For this, multigrid preconditioning has the following benefits: if geometrically aggregated multigrid method is used, the method can still be Jacobian free, and moreover, multigrid preconditioning theoretically scales linearly with respect to the number of unknowns. Aggregation efficiency, based on the damping of the Fourier modes, was studied in [143] for discontinuous Galerkin methods. This kind of study could be extended to nonlinear system, in order to build an efficient nonlinear solver that would be both scalable and Jacobian free. This solver would then benefit from the optimization of the residual computation proposed in the beginning of this list of challenges. This work was already addressed during the Marsu project, by implementing a method based on [11], but has been stucked since the mesh aggregation feature disappeared in the last versions of PaMPA.

Disseminate the library towards coastal engineering community. For four years, the Uhaina project has been developed in order to built an operational open source code gathering the last development in term of modeling and high order numerical approximation of Green-Naghdi system [46]. The Uhaina code can be seen as a plugin of AeroSol: some models and numerical methods are specifically developed for this application, but the code relies on AeroSol for parallel memory management, high order aspects, and I/O.

Extend the library to other high order numerical methods. The AeroSol library is able to manage both continuous and discontinuous finite elements methods. The memory is managed in such a flexible way that it can be easily extended to other high order numerical methods. For example, cubature-based continuous finite element methods are currently being implemented, and we will also consider using hybridizable discontinuous Galerkin methods for solving the elliptic step of the Green-Naghdi system within the Uhaina project.

Chapter 4

Work done during my leave at NIA

In the beginning of 2017, I was invited by Alireza Mazaheri (NASA Langley) for working with him on numerical problems related with reentry problems in a broad sense. Reentry problem happens either on military applications, or in the return to earth phase of a manned flight. During this phase in which the object is basically falling at high velocity, the spacecraft is inside a flow with a large Mach number. A detached shock is in front of the object, and the object is submitted to a large heat flux. The following problems were addressed: for reaching a high accuracy in the fluxes, a first order formulation of the Navier-Stokes system was searched, in which the fluxes are considered as additional unknowns. This work is detailed in section 4.1, and submitted in [118]. Another aspect of flows in reentry problems was addressed: the stabilization of high order methods with strong shocks. This was addressed by considering compact WENO limiters, which are also able to ensure the positivity of the density and pressure. This is detailed in section 4.2, and was published in [93]. In this chapter, each section has its own conclusion, the two topics tackled being strongly independent.

4.1 First order hyperbolic formulation for Navier-Stokes equations

4.1.1 Introduction

This topic was driven by previous work of Alireza [91], in which a numerical scheme for a scalar advection-diffusion problem was derived by using a first order hyperbolic formulation of the equation. The basic ideas can be understood in the following one dimensional scalar advection–diffusion equation

$$\partial_t u + c\partial_x u = \nu \partial_{xx} u \tag{4.1}$$

with $\nu \in \mathbb{R}_+^\star$ and $c \in \mathbb{R}$. In [18, 101], it was remarked that the regular stationary solutions of (4.1) are formally equal to the regular stationary solutions of the following system

$$\partial_\tau \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} c & -\nu \\ -\dfrac{1}{T_u} & 0 \end{pmatrix} \partial_x \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ -\dfrac{v}{T_u} \end{pmatrix} \tag{4.2}$$

where T_u is a relaxation parameter. It is straightforward to check that the first-order part of (4.2) is a hyperbolic system.

Considering the approximation of (4.2) instead of (4.1) may appear as attractive for the following reasons

- **Ensuring consistency between diffusion and advection.** Often, discretizations of (4.2) are performed by considering on one hand the advection discretization and on the other hand, the diffusion approximation. This may lead to a nonuniform accuracy of approximations, see for example [104], in which a nonuniform accuracy with fluctuating/Galerkin scheme is shown. Using (4.2) may allow to have a unified view of the discretization of advection and diffusion terms, see for example [1] for an application to diffusive equations of residual distribution schemes originally designed for advection.

- **Get an accurate representation of gradients.** In some applications the interesting predictive quantities are based on gradients (for example, the heat flux for the reentry problem). An accurate discretization for (4.1) is not necessarily accurate for the gradients, whereas a scheme based on (4.2), where the gradients are part of the unknowns, should be accurate also for gradients, see e.g., [92] for a discontinuous Galerkin approximation that is also accurate for gradients.

- **Relax the stiffness of the system.** By tuning the relaxation parameter T_u of (4.2), it is possible to get a well-conditioned system [101], where the conditioning is $O(1/h)$ instead of $O(1/h^2)$, therefore leading to a $O(1/h)$ instead of $O(1/h^2)$ number of time step for reaching convergence with classical iterative approach (Jacobi, Gauss-Seidel) for solving the linear system.

- **Derive boundary conditions.** In [43], a first-order formulation of the isothermal Navier-Stokes equations was used for deriving boundary conditions.

However, extending the formulation (4.2) for a scalar equation to a nonlinear system is not straightforward. For example, if the Navier-Stokes system is considered, the aim is to substitute the terms of the diffusive tensor by additional unknowns. But even if any choice of combination of the tensor variables can lead to a system that includes only first order derivatives, it is not sure that the system is hyperbolic. We indeed proved in [118] that some of the so-called hyperbolic formulations of the Navier-Stokes system [103, 102] are *not* hyperbolic.

In the meanwhile, Ilya Peshkov and collaborators had successfully proposed some hyperbolic formulations for Navier-Stokes systems [119, 41, 40]. This work is rather physically inspired; for example, the relaxation times that naturally appear in the first order formulation (as in (4.2))are calibrated for ensuring that the time dependent equations are still consistent with a fluid behavior. Our objective is slightly different: we just aim at getting a hyperbolic formulation whose stationary solutions are consistent with the stationary solutions of the stationary system. Also, in the spirit of the work of Nishikawa [101, 103, 102], we aim at using the relaxation times for relaxing the numerical stiffness of the system, and this is why we will look for formulations including several relaxation times in order to maximize the number of degrees of freedom which may be used for relaxing the stiffness of the system. Last, we aim at having a minimal number of additional variables and equations.

4.1.2 The scalar case

This case is presented just in order to understand the main issues and the strategy we will use. We are interested in the following nonlinear advection-diffusion equation

$$\partial_t u + \mathrm{div}_{\mathbf{x}} \mathbf{f}(u) = \mathrm{div}_{\mathbf{x}}(A(u)\nabla u) \tag{4.3}$$

where $u \in \mathbb{R} \mapsto \mathbf{f}(u) \in \mathbb{R}^d$ and $u \in \mathbb{R} \mapsto A(u) \in M_{d,d}(\mathbb{R})$ are regular, and $A(u)$ is a symmetric nonnegative matrix for all u. Then the regular stationary solutions of (4.3) are also stationary solutions of the following system

$$\begin{cases} \partial_\tau u + \mathrm{div}_{\mathbf{x}} \mathbf{f}(u) - \mathrm{div}_{\mathbf{x}} \mathbf{v} = 0 \\ \partial_\tau \mathbf{v} - \dfrac{1}{T_u} A(u)\nabla u = -\dfrac{\mathbf{v}}{T_u} \end{cases} \tag{4.4}$$

We aim at proving that (4.4) is hyperbolic. We can then gradually deal with the following cases

A is strictly nonnegative. In this case, the matrix

$$\begin{pmatrix} \dfrac{1}{T_u} & 0 \\ 0 & A^{-1}(u) \end{pmatrix}$$

is a symmetrizer of (4.4), so that (4.4) is hyperbolic. However, this does not hold if A is only nonnegative.

A is nonnegative. In this case, only a part of the matrix A must be chosen as additional variables. Existence of a matrix P, independent of u can be proven, such that

$$A(u) = P^T \begin{pmatrix} 0_{d-r,d-r} & 0_{d-r,r} \\ 0_{r,d-r} & \bar{A}(u) \end{pmatrix} P$$

where $\bar{A}(u)$ is the $r \times r$ symmetric, strictly nonnegative matrix of the endomorphism of $A(u)$ induced on the image of $A(u)$. We denote by $\bar{P}$ the r last rows of P. Then the following system

$$\begin{cases} \partial_\tau u + \mathrm{div}_{\mathbf{x}} \mathbf{f}(u) - \mathrm{div}_{\mathbf{x}}(\bar{P}^T \mathbf{v}) = 0 \\ \partial_\tau \mathbf{v} - \dfrac{1}{T_u} \bar{A}(u)\bar{P}\nabla u = -\dfrac{\mathbf{v}}{T_u} \end{cases}$$

can be proven to be symmetrizable.

Multiple relaxation times. A formulation involving several relaxation times can be obtained if we additionally suppose that the image of $A(u)$ can be decomposed into k spaces V_i, stable by $A(u)$, which do not depend on u

$$\mathrm{Im}(A(u)) = \bigoplus_{j=1}^{k} V_j.$$

Then, an orthonormal matrix P exists such that

$$A = P^T \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & \bar{A}_1(u) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \bar{A}_k(u) \end{pmatrix} P \tag{4.5}$$

where all the $\bar{A}_i$ are symmetric. Denoting by $\bar{P}_i$ the rows of P which indices are matching with the indices of the block $\bar{A}_i$ in (4.5). Then the following system

$$\begin{cases} \partial_\tau u + \mathrm{div}_{\mathbf{x}} \mathbf{f}(u) - \displaystyle\sum_{i=1}^{k} \mathrm{div}_{\mathbf{x}} \left(\bar{P}_i^T \mathbf{v}_i \right) = 0 \\ \partial_\tau \mathbf{v}_i - \dfrac{1}{T_i} \bar{A}_i(u) \bar{P}_i \nabla u = -\dfrac{\mathbf{v}_i}{T_i} \qquad \forall i = 1 \dots k \end{cases}$$

can be proven to be symmetrizable.

4.1.3 System case

We are now interested in the following system

$$\partial_t \mathbf{U} + \mathrm{div}_{\mathbf{x}} \mathbf{F}(\mathbf{U}) = \mathrm{div}_{\mathbf{x}} \left(\mathbf{A}(\mathbf{U}) \nabla \mathbf{U} \right) \tag{4.6}$$

We aim at finding a hyperbolic formulation, and for proving hyperbolicity, we would like to prove symmetrizability. In the system case, the diffusive tensor is not symmetric in general: for example, the diffusive tensor of the Navier-Stokes system expressed in temperature and velocity is not symmetric. We also must be careful with the advection part, which is an additional difficulty with respect to the scalar case.

Relying on results of [136, 137], and supposing that the system is equipped with an entropy, the following set of variables was chosen

- for the non dissipative equations (e.g. mass equation for the Navier-Stokes system), keep the conservative variable,
- replace all the other variables by entropic variables.

This change of variables allows to make appear several symmetric matrices in the system: in front of the time derivative, the matrix of the variable change is symmetric; the matrices that appear in the advection part are also symmetric according to [136, 137]. Moreover, in the applications we are addressing, the diffusive tensor is also symmetric. The general case is detailed in [118], and basically follows the steps of the scalar case: full rank diffusive tensor, degenerate diffusive tensor, and multiple relaxation times. We give here only the formulations found in two important cases

The isothermal Navier-Stokes system .

$$\begin{cases} \partial_t \rho + \mathrm{div}_{\mathbf{x}}(\rho \mathbf{u}) = 0 \\ \partial_t (\rho \mathbf{u}) + \mathrm{div}_{\mathbf{x}} \left(\rho \mathbf{u} \otimes \mathbf{u} + P \mathrm{I}_d \right) = \mathrm{div}_{\mathbf{x}} \boldsymbol{\tau} \end{cases} \tag{4.7}$$

where ρ is the density, $\mathbf{u}$ is the velocity. The pressure P is a function of the density, such that $P'(\rho) > 0$, and the diffusive tensor is

$$\boldsymbol{\tau}_{i,j} = \mu \left(\partial_{\mathbf{x}_j} \mathbf{u}_i + \partial_{\mathbf{x}_i} \mathbf{u}_j \right) + \lambda \delta_{i,j} \, \mathrm{div}_{\mathbf{x}} \mathbf{u}$$

where μ is a function of ρ, and $\lambda = -2\mu/3$.

Then we denote by $\mathbf{v}_i$, $1 \leqslant i \leqslant d-1$ an orthonormal basis of the kernel of the d dimensional matrix

$$\begin{pmatrix} 1 & \dots & 1 \\ \vdots & & \vdots \\ 1 & \dots & 1 \end{pmatrix}$$

and by $\hat{P}$ the (orthonormal) matrix

$$\begin{pmatrix} & \mathbf{v}_1^T & \\ & \vdots & \\ & \mathbf{v}_{d-1}^T & \\ \frac{1}{\sqrt{d}} & \cdots & \frac{1}{\sqrt{d}} \end{pmatrix}.$$

the following first-order formulation of the isothermal Navier-Stokes equations is hyperbolic:

$$\begin{cases} \partial_\tau \rho + \mathrm{div}_{\mathbf{x}}(\rho \mathbf{u}) = 0 \\ \partial_\tau (\rho \mathbf{u}) + \mathrm{div}_{\mathbf{x}}(\rho \mathbf{u} \otimes \mathbf{u} + P \mathrm{I}_d) = \nabla \left(\hat{P}^T \hat{\boldsymbol{\tau}} \right) + \mathrm{div}_{\mathbf{x}} \check{\boldsymbol{\tau}} \\ \partial_\tau \check{\boldsymbol{\tau}}_{i,j} - \dfrac{\mu}{T_{2\mu}} \left(\partial_{\mathbf{x}_i} \mathbf{u}_j + \partial_{\mathbf{x}_j} \mathbf{u}_i \right) = -\dfrac{\check{\boldsymbol{\tau}}_{i,j}}{T_{2\mu}} \qquad \forall 1 \leqslant i < j \leqslant d \\ \partial_\tau \hat{\boldsymbol{\tau}}_i - \dfrac{1}{T_{2\mu}} \mathbf{v}_i \cdot \begin{pmatrix} \partial_{\mathbf{x}_1} \mathbf{u}_1 \\ \vdots \\ \partial_{\mathbf{x}_d} \mathbf{u}_d \end{pmatrix} = -\dfrac{\hat{\boldsymbol{\tau}}_i}{T_{2\mu}} \qquad \forall 1 \leqslant i \leqslant d-1 \\ \partial_\tau \hat{\boldsymbol{\tau}}_d - \dfrac{2\mu + d\lambda}{T_{2\mu+d\lambda}\sqrt{d}} \left(\displaystyle\sum_{i=1}^{d} \partial_{\mathbf{x}_i} \mathbf{u}_i \right) = -\dfrac{\hat{\boldsymbol{\tau}}_d}{T_{2\mu+d\lambda}} \end{cases}$$

with $\check{\boldsymbol{\tau}}_{ii} = 0$ and $\check{\boldsymbol{\tau}}_{i,j} = \check{\boldsymbol{\tau}}_{j,i}$. Note that the proposed formulation includes two relaxation times, $T_{2\mu}$ and $T_{2\mu+d\lambda}$.

The full Navier-Stokes system. Based on the same method, the following first order formulation of the Navier-Stokes, including one relaxation time T can be proven to be hyperbolic

$$\begin{cases} \partial_\tau \rho + \mathrm{div}_{\mathbf{x}}(\rho \mathbf{u}) = 0 \\ \partial_\tau (\rho \mathbf{u}) + \mathrm{div}_{\mathbf{x}}(\rho \mathbf{u} \otimes \mathbf{u} + P \mathrm{I}_d) = \mathrm{div}_{\mathbf{x}} \boldsymbol{\tau} \\ \partial_\tau (\rho E) + \mathrm{div}_{\mathbf{x}}((\rho E + P)\mathbf{u}) = \mathrm{div}_{\mathbf{x}} \boldsymbol{q} \\ \partial_\tau \boldsymbol{\tau}_{i,j} - \dfrac{1}{T} \left(\mu \left(\partial_{\mathbf{x}_i} \mathbf{u}_j + \partial_{\mathbf{x}_j} \mathbf{u}_i \right) + \delta_{i,j} \lambda \left(\displaystyle\sum_k \partial_{\mathbf{x}_k} \mathbf{u}_k \right) \right) = -\dfrac{\boldsymbol{\tau}_{i,j}}{T} \quad \forall 1 \leqslant i \leqslant j \leqslant d \\ \partial_\tau \boldsymbol{q}_i - \dfrac{1}{T} \left(\displaystyle\sum_{k,l} \mu \left(\partial_{\mathbf{x}_k} \mathbf{u}_l + \partial_{\mathbf{x}_l} \mathbf{u}_k \right) \mathbf{u}_i + \lambda \mathbf{u}_i \left(\sum_k \partial_{\mathbf{x}_k} \mathbf{u}_k \right) + \kappa \partial_{\mathbf{x}_i} \theta \right) = -\dfrac{\boldsymbol{q}_i}{T} \quad \forall 1 \leqslant i \leqslant d \end{cases}$$

Note that in this formulation, the auxiliary variables coming from the energy equation are the components of the full energy diffusion (including the work of the stress strain tensor), and not the Fourier terms.

4.1.4 Numerical results

Based on the different first order formulation, a discontinuous Galerkin discretization was proposed. One of the difficulties is that in general, the hyperbolic system is not conservative. The formulation we used was derived from [125], and stabilized with Lax-Friedrich type diffusion on sides.

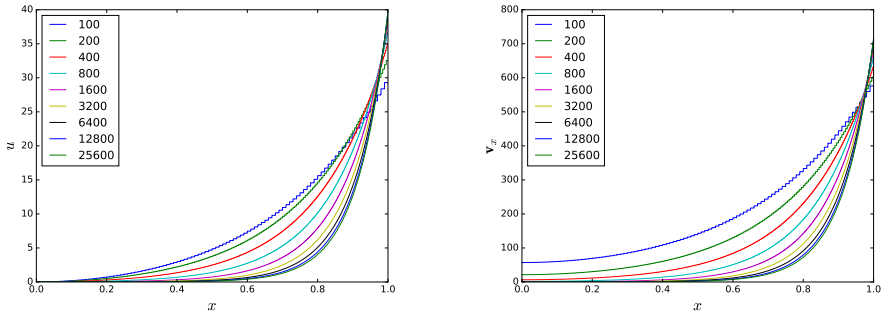


Figure 4.1: Solution u (left) and the diffusive flux $\mathbf{v}_x$ (right) obtained with the one dimensional test case with the first-order scheme with 100 to 25600 cells.

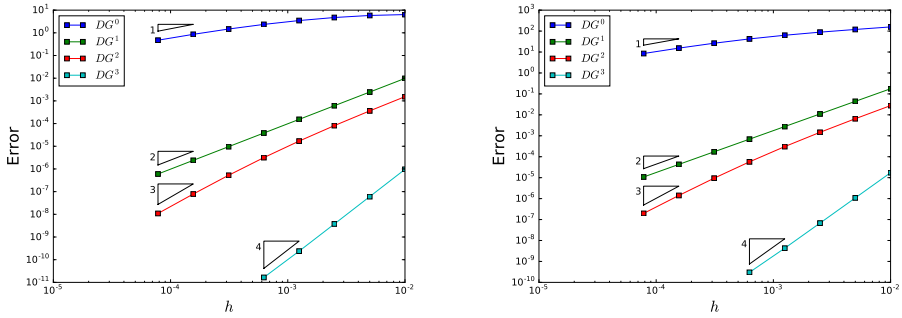


Figure 4.2: Error obtained for u (left) and for $\mathbf{v}_x$ (right) in log-log coordinates for DG^p with $p = 0, 1, 2, 3$.

4.1.4.1 Scalar case

We are considering a one dimensional linear problem on $[0; 1]$. Dirichlet boundary conditions are enforced on both sides of the domain: $u(x = 0) = u_0$ and $u(x = 1) = u_1$. The analytical solution is

$$\begin{cases} u(x) = \dfrac{(u_1 - u_0)\exp\left(\dfrac{cx}{\mu}\right) + u_0 \exp\left(\dfrac{c}{\mu}\right) - u_1}{\exp\left(\dfrac{c}{\mu}\right) - 1} \\ \mathbf{v}_x = \mu \dfrac{(u_1 - u_0)\dfrac{c}{\mu}\exp\left(\dfrac{cx}{\mu}\right)}{\exp\left(\dfrac{c}{\mu}\right) - 1} \end{cases}.$$

In the applications, c is fixed to 18, μ to 1.5, $u_0 = 0$ and $u_1 = 40$. The relaxation time is fixed to 1. In Figure 4.1, the numerical solution obtained with one-dimensional meshes with 100 to 25600 cells with the first-order scheme are shown, showing convergence towards the analytical solution. Then, in Figure 4.2, the order obtained on the two variables are shown. For each variable and each degree, the order obtained is the optimal one (i.e. $p + 1$).

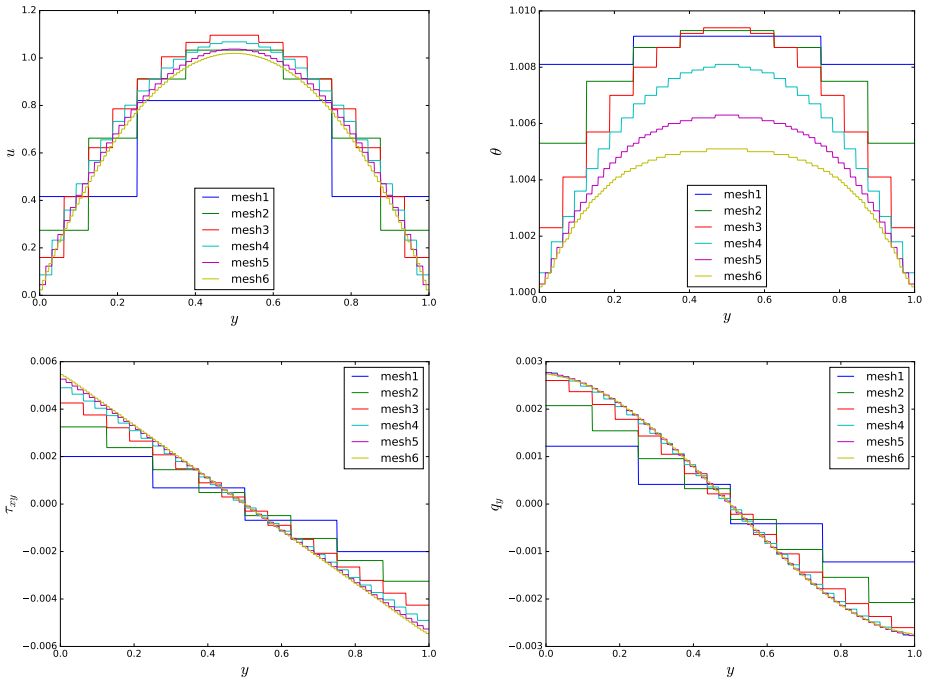


Figure 4.3: Results obtained for the horizontal velocity (top left), the temperature (top right), $\boldsymbol{\tau}_{xy}$ and $\boldsymbol{q}_y$ for the Poiseuille flow, with the first-order scheme, with meshes composed of 4^2, 8^2 $\dots 128^2$ cells.

4.1.4.2 Navier-Stokes system

We test the hyperbolic formulation with a discontinuous Galerkin formulation. The test case is the Poiseuille flow, which was already described in subsubsection 3.3.1.3 (Note that in subsubsection 3.3.1.3, T was used for the temperature, whereas we are using θ here). Additionally, the exact solution on the auxiliary variables is

$$\begin{aligned}
\boldsymbol{\tau}_{xx} &= \boldsymbol{\tau}_{yy} = \boldsymbol{q}_x = 0 \\
\boldsymbol{\tau}_{xy} &= 4\sqrt{2}\mu u_c(1-2y) \\
\boldsymbol{q}_y &= 16\mu u_c^2 y(1-y)(1-2y) + \frac{\kappa 16\theta_w Pr M^2(\gamma-1)}{3}(1-2y)\left(2y^2-2y+\frac{1}{2}\right)
\end{aligned}$$

In Figure 4.3, the numerical solution obtained with the first order scheme for the different meshes is shown, in which convergence towards the exact solution is observed. Then, in Figure 4.4, the order obtained on the velocity, temperature, and also $\boldsymbol{\tau}_{xy}$ and $\boldsymbol{q}_y$ are shown. The observed order matches with the optimal order, namely $p+1$ for a DG^p method.

4.1.5 Conclusion

In this section, a framework for deriving first-order hyperbolic formulation of nonlinear advection-diffusion dissipative systems was derived. This allowed to propose hyperbolic first order formulation for the isothermal and non isothermal Navier-Stokes system. Then a discontinuous Galerkin discretization for the hyperbolic formulation was proposed. Numerical results proved that this formulation is accurate for discretizing nonlinear dissipative systems,

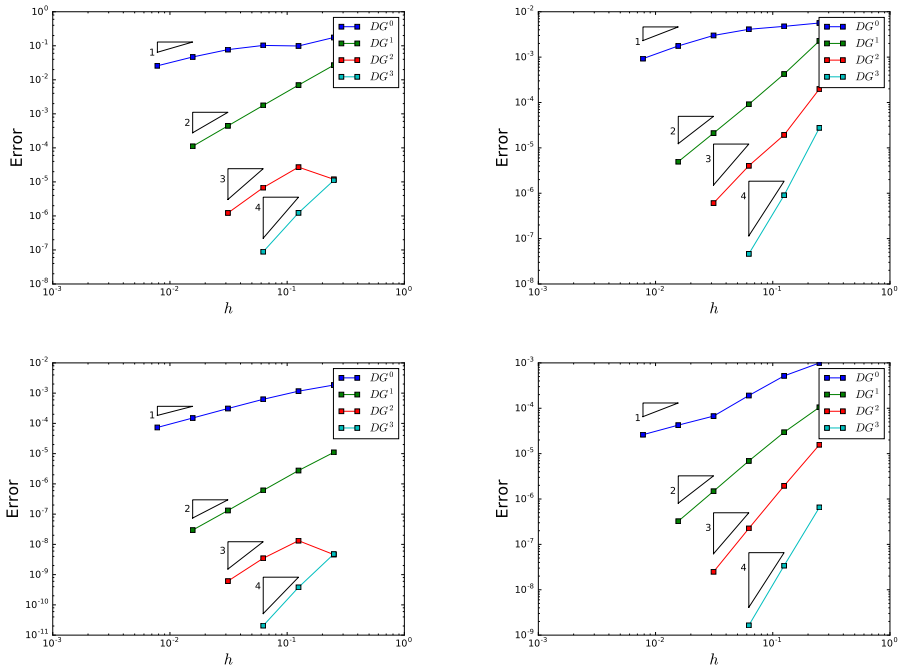


Figure 4.4: Order obtained on the velocity (top left), temperature (top right), on $\boldsymbol{\tau}_{xy}$ (bottom left) and on $\boldsymbol{q}_y$ (bottom right).

by ensuring a good representation of the gradients. The following directions will be explored in the coming years

Study the dependency of the numerical scheme obtained on the relaxation time(s). The relaxation time(s) are free parameter that can be adjusted, depending on what is needed; a first study could be the dependency of the accuracy of the numerical scheme with respect to the relaxation time. As announced in the introduction, the relaxation time may be optimized for relaxing the stiffness of the system. Another objective could be to study the error obtained with respect to the Reynolds number, for ensuring a uniform convergence behavior with respect to the Reynolds number. Note that this whole point should be first studied in the scalar case, possibly with an anisotropy in the diffusion.

Reduce the number of unknowns. If the previous point is convincing, it does not seem rational to consider industrial applications with such an increase in the number of variables and unknowns. Following [92], it should be possible to eliminate the auxiliary variables for obtaining a rational number of unknowns. This would then be interesting to compare the schemes with the classical theory of discretization of elliptic systems with the discontinuous Galerkin method [7].

Test the formulation for the implementation of boundary conditions. As described in the introduction, such a formulation can be useful for the implementation of boundary conditions, and was used in the context of isothermal Navier-Stokes system [43]. We would like to explore the extension to the full Navier-Stokes system.

4.2 Compact and bounded WENO limiters for discontinuous Galerkin methods for the Euler system

4.2.1 Introduction

In this section, we address the limiting of polynomials approximations with compact WENO methods developed in [93]. The advantage of WENO type limiters is that they can be applied to arbitrary high order methods, without affecting the expected order of accuracy of the underlying DG scheme if applied to smooth regions. Available WENO schemes are either non-compact [73, 69, 120, 138, 122, 121, 153, 150], which is undesirable for discontinuous Galerkin methods, or have difficulties in reducing the residuals [73, 69, 150]. The proposed WENO-FV and WENO-FD schemes of Zhu and Shu [154, 155] address the convergence difficulties that are observed in the earlier WENO procedures [85, 151].

Here, two classes of compact WENO polynomial limiters are presented for DG methods on simplex elements that may be considered extensions of the WENO-FV [154] and WENO-FD [152] schemes. Although the original compact WENO limiters for finite volume and finite difference schemes rely on an extended stencil for polynomial reconstruction, the main idea behind the compact WENO for DG schemes is to construct a limited polynomial directly from the underlying DG scheme. The compact WENO-DG limiters presented here, therefore, require a stencil only as large as $n_{faces}+1$ number of elements, where n_{faces} corresponds to the number of faces of the elements in which the limiter is being applied. In the first proposed approach, the compact stencil is considered, and a series of linear polynomials is constructed by solving a $n_{dof}\times n_{dof}$ linear system, where n_{dof} denotes the number of degrees of freedom for linear approximation. In the second approach, a linear polynomial is constructed by solving a series of constrained least-squares (LSQ) minimization problems within the compact stencil. In both approaches, a WENO polynomial limiter is obtained with a convex combination of the original polynomial and the constructed linear polynomials that are added using nonlinear weights. The numerical fluxes are evaluated with the Local Lax-Friedrichs flux, and a compact positivity preserving limiter [149], already detailed in subsubsection 1.4.3.2, is also applied to the solutions to ensure pressure and density remain bounded and physical at all time.

4.2.2 Limitation proposed

On each cell, the solution is approximated by a polynomial. The WENO limitation consists in building other polynomial approximations on the cell $\tilde{p}^{(i)}$ with $i>0$, and in building a new polynomial representation $\tilde{p}^{(0)}$ based on the averaging between the original approximation and the new polynomial representations:

$$\tilde{p}^{(0)} = \gamma_0 \left(\frac{1}{\gamma_0} p^{(0)} - \sum_i \frac{\gamma_i}{\gamma_0} \tilde{p}^{(i)} \right) + \sum_i \gamma_i \tilde{p}^{(i)}, \tag{4.8}$$

which holds for arbitrary positive coefficients γ_j satisfying $\sum_j \gamma_j = 1$, where, γ_j, are the linear weights defined as

$$\gamma_j = 0.001\,(j \neq 0), \qquad \gamma_0 = 1 - \sum_{j\neq 0} \gamma_j,$$

and the $\tilde{p}^{(i)}$ are the to-be-reconstructed polynomials. The linear weights are then replaced with the nonlinear weights, ω_i, and the following reconstructed candidate polynomial is sought for the target element

$$p^{(0)new} = \omega_0 \left(\frac{1}{\gamma_0} p^{(0)} - \sum_{j>0} \frac{\gamma_j}{\gamma_0} \tilde{p}^{(j)} \right) + \sum_{j>0} \omega_j \, \tilde{p}^{(j)}, \tag{4.9}$$

where the nonlinear weights are computed by adopting the high-order WENO-FD [154] and WENO-FV [155] techniques of Zhu and Shu for the present WENO-DG schemes. These techniques are based on the WENO-Z strategy given in [13, 17, 36]. The nonlinear weights are then defined as

$$\omega_i = \frac{\overline{\omega}_i}{\sum_j \overline{\omega}_j}, \quad \overline{\omega}_j = \gamma_j \left(1 + \frac{\tau}{(\epsilon + \beta_j)^2} \right), \quad \tau = \left(\frac{\sum_i |\beta_0 - \beta_i|}{n} \right)^2, \tag{4.10}$$

where n is the total number of reconstructed polynomials used in constructing the WENO polynomial $p^{(0)new}$ as given in Eq. (4.9) (i.e., maximum integer value of i) plus the polynomial of the target element K_0. Here $\epsilon = 10^{-12}$ is a small number to avoid zero denominator, and β is the classical smoothness indicator defined as (see e.g., Jiang and Shu [73] and Kolb [81])

$$\beta_i = \sum_{|l|=1}^{k} |K_0|^{(|l|-1)} \int_{K_0} \left(\frac{\partial^{|l|}}{\partial x^{l_1} \, \partial y^{l_2}} \, \tilde{p}^{(i)} \right)^2, \quad l = (l_1, l_2), \tag{4.11}$$

where k denotes the polynomial order, and $|K_0|$ is the volume of the target element.

The above procedure is complete with reconstruction of the polynomials $\tilde{p}^{(i)}$. In [93], two $\tilde{p}^{(i)}$ reconstruction procedures are proposed. In the two procedures, the polynomials $\tilde{p}^{(i)}$ are linear, and one of the challenge is to ensure conservation in the formula (4.9), which is ensured provided

$$\forall j > 0 \qquad \frac{1}{|K_0|} \int_{K_0} \tilde{p}^{(j)} = \overline{u}^{(0)}.$$

In the following, we denote by K_0 the target cell, and by K_j, $j \geqslant 1$ the direct neighboring cells. The two following procedures are proposed

- The first procedure is adapted to simplices; the number of polynomials is equal to the number of neighbors. For each neighbor j, a first order polynomial $\tilde{p}^{(j)}$ is computed such that

  $$\forall k \neq j \qquad \frac{1}{|K_k|} \int_{K_k} \tilde{p}^{(j)} = \overline{u}^{(j)}. \tag{4.12}$$

  Such a polynomial exists because we are dealing with a simplex, which, in dimension d, has $d+1$ degrees of freedom for linear polynomials, and (4.12) is imposing $d+1$ constraints. The total number of polynomials reconstructed is equal to $d+1$.

- The second procedure is suitable on any type of mesh. In this case, a single polynomial is rebuilt, such that

  $$arg \min_{\tilde{u}} \sum_{l \in S} \left(\frac{1}{|K_l|} \int_{K_l} \tilde{p}^{lsq}(\tilde{u}, x) \, d\mathbf{x} - \overline{u}^{(l)} \right)^2, \tag{4.13}$$

where S is the set of the neighbors of the target cell. The minimization is made under the constraint that the mean of the constructed polynomial remain the same as the mean of the original polynomial

$$\frac{1}{|K_0|}\int_{K_0} \tilde{p}^{lsq}\, d\mathbf{x} = \overline{u}^{(0)}. \tag{4.14}$$

The two procedures have been presented for the scalar case. When a system is considered, the procedure is applied characteristic wise in all the directions defined by the faces of the target element K_0. The final limited polynomial is obtained as the arithmetic average of the limited polynomial in all the directions.

4.2.3 Numerical results

In practical computations, classical trouble cell indicators are not employed. The proposed WENO is applied to all the computational elements, which is beneficial in ensuring that the proposed WENO does not affect the desired order of accuracy of the DG scheme even if it is applied to smooth region.

4.2.3.1 Order verification

Consider the two-dimensional Euler equations

$$\mathbf{w}_t + \nabla \cdot \mathbf{f}(\mathbf{w}) = \mathbf{S}, \quad \mathbf{w}(\mathbf{x}, t = 0) = \mathbf{w}_0(\mathbf{x}), \tag{4.15}$$

with the vector of conservative variables $\mathbf{w}$ and the source $\mathbf{S}$,

$$\mathbf{S} = \begin{pmatrix} 0.4\cos(x+y) \\ 0.6\cos(x+y) \\ 0.6\cos(x+y) \\ 1.8\cos(x+y) \end{pmatrix}, \tag{4.16}$$

in domain $(x, y) \in [0, 2\pi]$. This system has the following exact steady state solution

$$\rho = 1 + 0.2\sin(x+y), \quad u = 1, \quad v = 1, \quad p = 1 + 0.2\sin(x+y), \tag{4.17}$$

which is imposed on the domain boundaries. A series of randomly generated irregular triangular grids is considered and steady state solutions of the DG (Pk), $k = 1, 2, 3, 4$, scheme with and without the presented WENO and positivity preserving limiters are obtained. The predicted high-order DG (Pk) solutions are compared with the exact solution on the given grids by computing the L^2 error. The resulting L^2 errors are shown in Figure 4.5. The L^2 errors for both WENO limiters are identical to the decimal point shown and therefore only one set of values are given. The WENO and positivity preserving limiters are both applied to all the elements within the domain. The $(k+1)$th order of accuracy for both DG and the WENO-DG schemes are verified, confirming that the proposed WENO maintains the expected order of accuracy of the underlying DG (Pk) scheme. It is also interesting to note that the presented error magnitudes are order of magnitude smaller than the corresponding results reported with the third-, fourth-, and fifth-order WENO-FV schemes in [152].

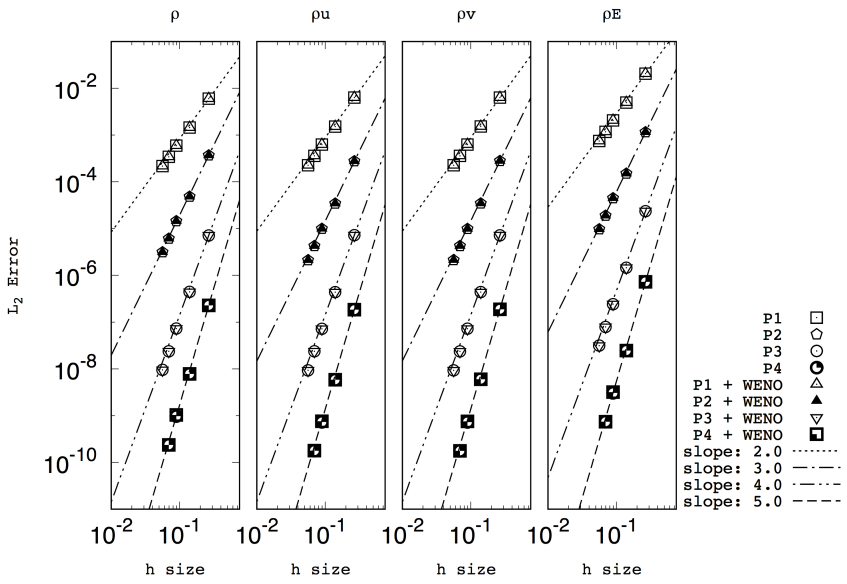


Figure 4.5: Verification – Order of accuracy plots for the conserved density, x and y momentums, and energy for DG and the proposed WENO-DG on irregular triangular elements.

4.2.3.2 The Shu-Osher test

The one-dimensional idealization of the shock-turbulence interaction suggested by Shu-Osher [139] is considered in a two-dimensional framework. The goal of this test is to assess the proposed limiters in capturing the shock wave and its interactions with the unsteady low frequency density fluctuations and its waves propagation downstream of the shock. The computational domain $(x, y) \in [-5.0, -0.1] \times [5.0, 0.1]$ is discretized with triangular elements and a characteristic mesh size of $h/40$. The domain is initialized as

$$(\rho, u, v, p)|_{t=0} = \begin{cases} (3.857143, 2.629369, 0., 10.33333), & x < 4.0, \\ (1. + 0.2\sin(5x), 0., 0., 1.), & x \geqslant 4.0. \end{cases} \tag{4.18}$$

This corresponds to a Mach 3 shock ($\gamma = 1.4$) interacting with the sine waves density field. The results at $t = 1.8$ are presented in Figure 4.6, Figure 4.7 and Figure 4.8. The fine structured of the shock-density wave interactions are clearly captured by the proposed limiter.

4.2.3.3 Double Mach reflection

The double Mach reflection problem is originally proposed in [146] as a benchmark test for Euler solvers. The problem consists of a Mach 10 shock front that meets a 30° inclined ramp. As it is a customary, this problem is solved in a computational domain $(x, y) \in [0, 4] \times [0, 1]$ with a coordinate system that is aligned with the ramp. The domain is discretized with irregular triangular elements and a mesh size of $h = 1/200$.

The density contours for a truncated domain $(x, y) = [0, 3.2] \times [0, 1]$ (the computational domain extends to $x = 4$) with 41 equally spaced isolines from 1.5 to 22.5 at $t = 0.2$ are shown in Figure 4.9 for the second-, third-, fourth-, and fifth-order WENO-DG. The closeup views

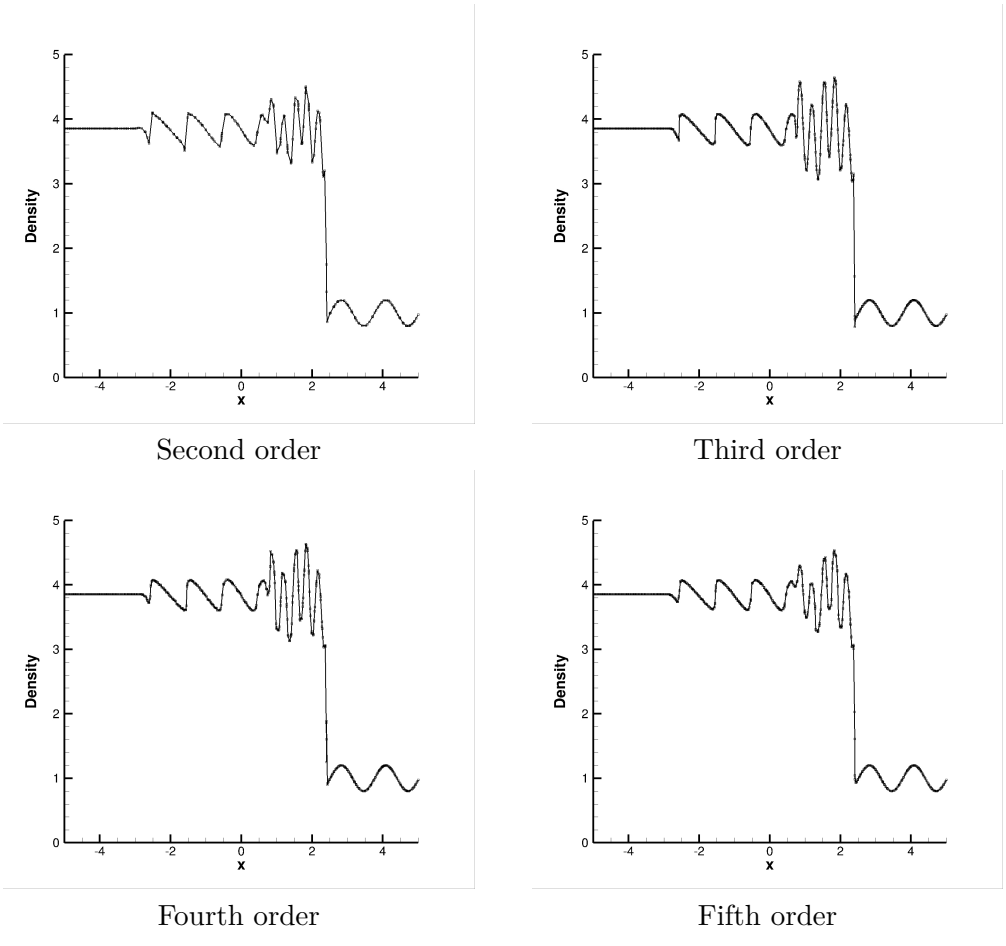
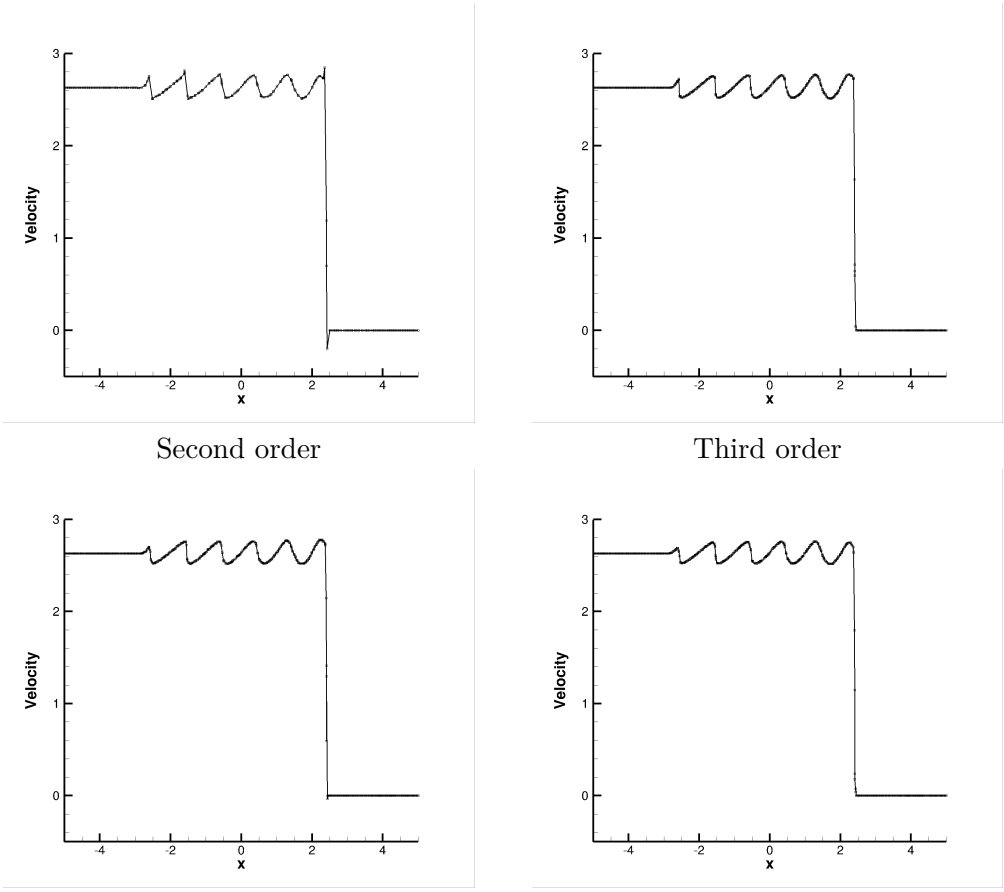


Figure 4.6: Predicted density profile for the Shu-Osher problem with $h/40$ irregular triangular elements at $t = 1.8$.



Second order | Third order

Fourth order | Fifth order

Figure 4.7: Predicted velocity profile for the Shu-Osher problem with $h/40$ irregular triangular elements at $t = 1.8$.

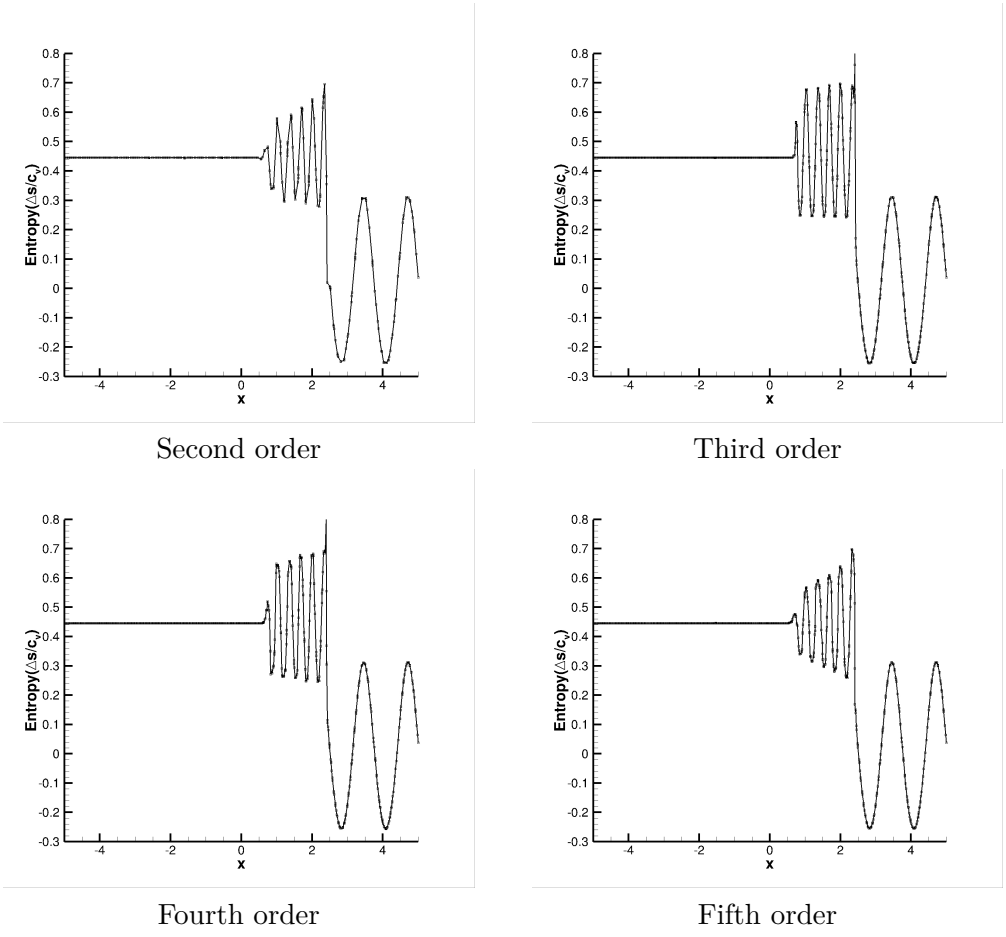


Figure 4.8: Predicted entropy profile, $ln(p/\rho^{\gamma})$, for the Shu-Osher problem with $h/40$ irregular triangular elements at $t = 1.8$.

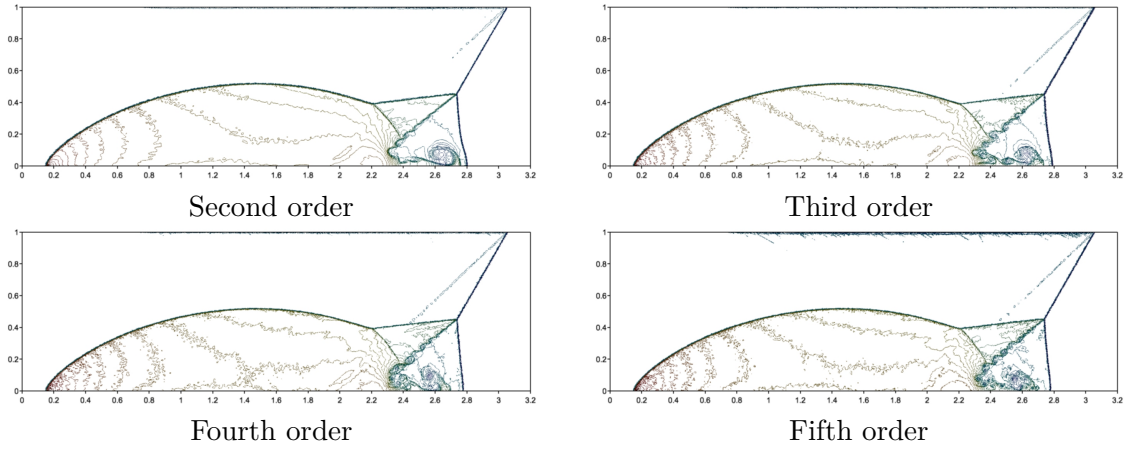


Figure 4.9: High-order DG with proposed WENO and positivity preserving with irregular triangular elements ($h = 1/200$) in $(x, y) \in [0, 4] \times [0, 1]$. Shown are 41 equidistance density contour lines at $t = 0.2$ from 1.5 to 22.5.

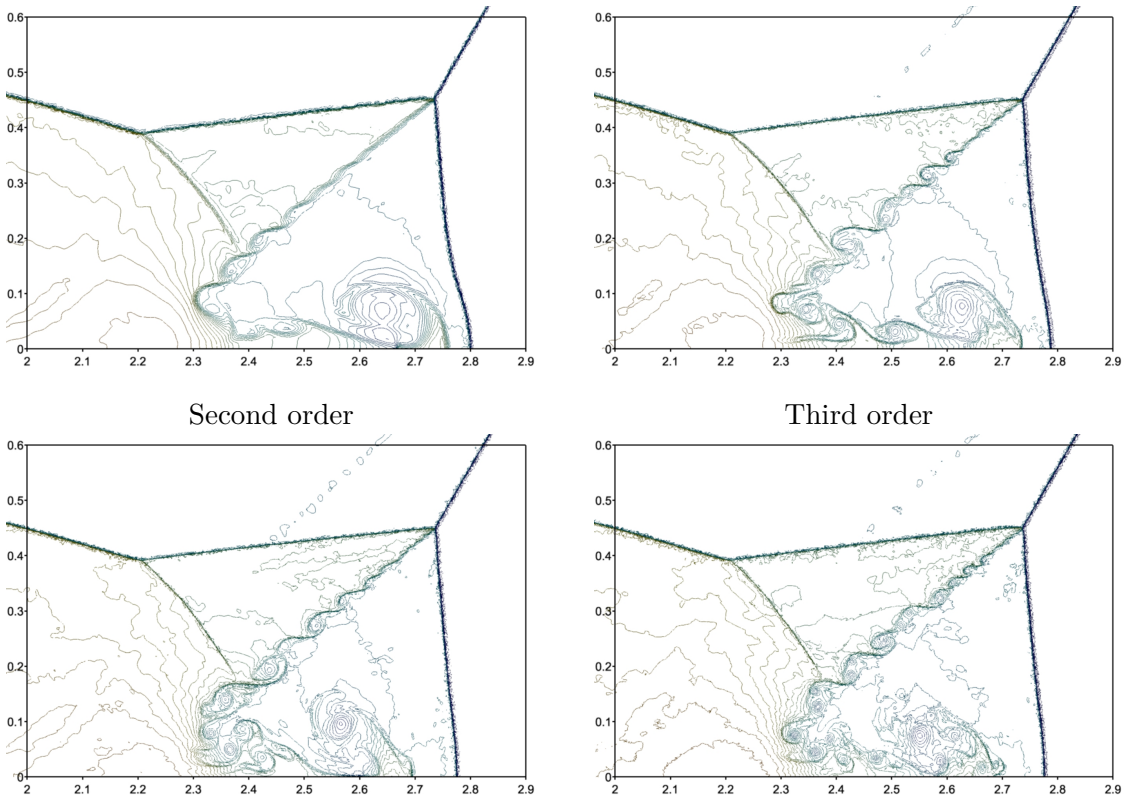


Figure 4.10: High-order DG with proposed WENO and positivity preserving with irregular triangular elements ($h = 1/200$) in $(x, y) \in [0, 4] \times [0, 1]$. Shown are zoom-in views around the Mach stems at $t = 0.2$ with 41 equidistance density contour lines from 1.5 to 22.5.

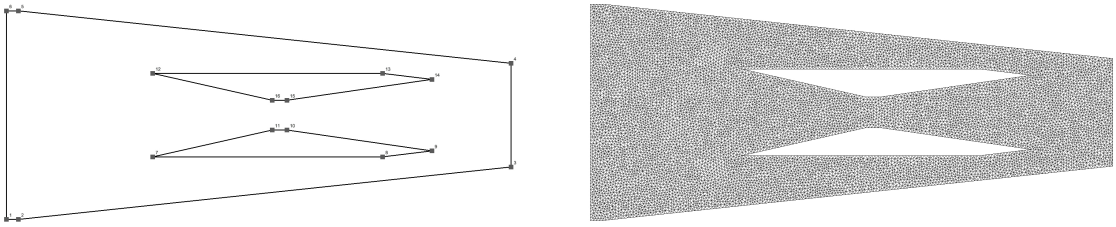
of the triple Mach points region, $(x, y) \in [2, 2.9] \times [0, 0.6]$, are shown in Figure 4.10. Clearly, the complexity of the triple Mach points is captured by the proposed limiters on the irregular triangular mesh. The quality of the solution improves with increasing the polynomial order on the same identical irregular triangular mesh. The resolution of the curled flow structures along the primary slip line, which is caused by its interactions with the secondary reflected shock emanating from the secondary triple point, is often used for judging the quality of the numerical scheme. The presented results provide a remarkable curled flow structures compared to the solutions reported by Hu and Shu [69], Zhu et al. [156], and Dumbser et al [38]. Further improvement in the WENO could reduce the noise in the predicted contour lines. Grid adaptation could also enhance the results further.

4.2.3.4 Scramjet

Consider a two-strut scramjet (see Figure 4.11 and Table 4.1) with a Mach 3 inflow imposed on the left surface boundary. The corresponding density contours in the range of 1.5 and 8.0 are shown in Figure 4.12. These results are in agreement with the previously published results [82, 42, 5] on a similar configuration.

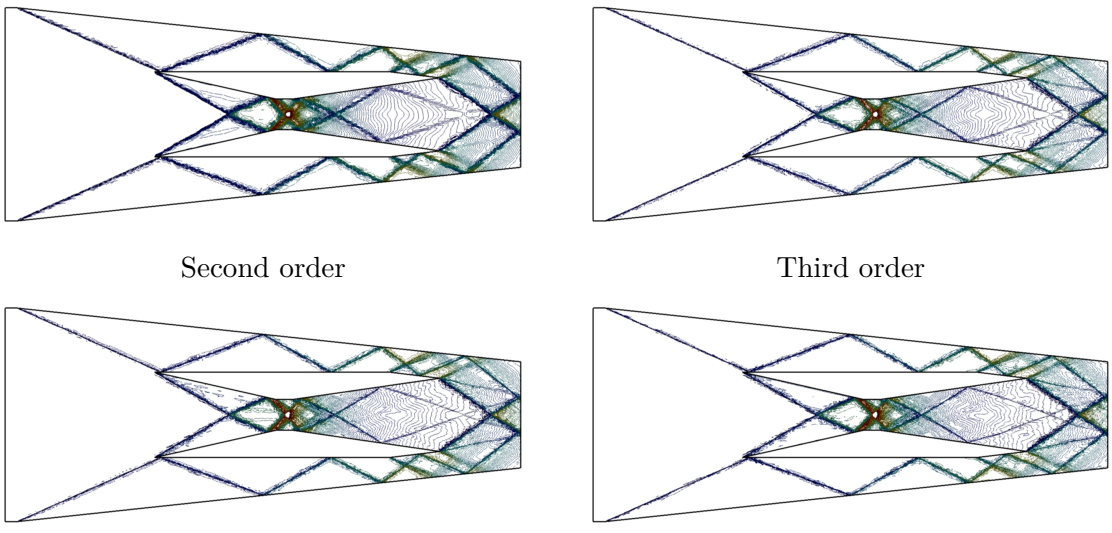
Table 4.1: The coordinates of the lower half portion of the two-strut scramjet geometry shown in Figure 4.11.

Points	1	2	3	7	8	9	10	11
(x, y)	(0,-3.5)	(0.4,-3.5)	(16.9,-1.74)	(4.9,-1.4)	(12.6,-1.4)	(14.25,-1.2)	(9.4,-0.5)	(8.9,-0.5)



Geometry (see Table 4.1) Mesh

Figure 4.11: The geometry of the two-strut scramjet test case with a sample irregular grid with 10,000 vertices. Coordinates of the lower half portion of the geometry is given in Table 4.1.



Second order Third order

Fourth order Fifth order

Figure 4.12: Hundred equally spaced density contours in the range of 1.5 and 8.0 using the second presented WENO approach for the Mach 3 two-strut scramjet test case with 10,000 unadapted irregular triangular elements.

4.2.4 Conclusion

The compact WENO limiter combined with the positivity preserving limiter allowed to get results on challenging tests until fifth order of accuracy. The limiter is easy to implement and does not destroy the order of accuracy in the smooth zones. Also, its compactness is well suited with parallel computing and with the original compact stencil of the discontinuous Galerkin method. Last, it can be easily extended to hybrid meshes and to three dimensional meshes, and this is what we plan to do in the future years.

Conclusion

In this document, the results I obtained over the last decade on high order methods for compressible flows have been summarized. They deal with flows in different regimes: low Mach number flows, large Mach number flows with strong shocks, multiphase flows. I have also been a major contributor to the AeroSol library, a scalable high order finite element library able to deal with hybrid meshes and with continuous and discontinuous discretizations. Possible research directions for the modeling and numerical approximation of multiphase compressible flows have been exposed in section 1.6, in section 2.6 for low Mach number flows, in section 3.6 for the AeroSol library, in subsection 4.1.5 for the hyperbolic formulation of Navier-Stokes equations, and in subsection 4.2.4 for compact WENO limiters. Beyond these research directions, we mention the following rather transversal research directions for the future

Derive optimal k–exact quadrature formulas adapted to maximum preserving limiters. In chapter 1 and chapter 4, the positivity preserving method of [149] was used on triangles and quads. However, the number of quadrature points for the quadrature formula proposed in [149] increases a lot when the degree increases. Also, the points have been derived on line, quads, and triangles, but not on general three dimensional shapes. Last, the theoretical CFL number depends on a ratio between the weights of the k–exact cell quadrature formula and the weights of the side quadrature formula. We should then derive a quadrature formula that optimizes this CFL number, under the constraint of being k–exact. From a technical point of view, some modern tools have been developed for the derivation of quadrature formulas [145] and could be a starting point for such a study.

Consider low Mach number flows in multiphase flows. chapter 1 deals with multiphase flows, and chapter 2 deals with low Mach number flows. In some usual configurations, for example a multiphase flow involving liquid water and gaseous air, the strong disparity between the sound velocities induces that the Mach number in the water is low. This problem was addressed for example in [84], and it would be interesting to study if the methods for single phase low Mach number flows we develop could be applied to multiphase flows, especially to the numerical method discussed in chapter 1.

Address the discretization of turbulent models. The convergence between modern high order methods and modern methods for turbulence modeling is one of the research direction of the Cagire team. We aim first at discretizing RANS models. I advised

one internship on this topic two years ago, and another one is currently working on the implementation of Spalart-Almaras and $k-\varepsilon$ models. Within the ASTURIES project, second order RANS models will be considered such as EB-RSM models [89]. The high order discretization of these systems raises challenges for the imposition of boundary conditions. This research will be done in collaboration with Rémi Manceau.

Address the challenge of reproducibility in high order methods and possibly large scale computing. The development of high order finite element methods has introduced a strong technological gap with respect to traditional second order finite volumes methods. Indeed, when dealing with high order finite element methods, it is required to define finite element basis, quadrature formulas, and geometrical functions for handling curved meshes, all of this being useless when second order methods are implemented. Moreover, and especially with discontinuous Galerkin methods, a strong variability may be observed in the technical implementation: for example, the basis functions can be defined either on the reference element or on the physical element, and also the basis can be of Lagrange or Modal type among others. This becomes even more complicated when it comes to deal with stabilizations: often, several limiters are piled up for ensuring positivity of physical variables, inequality of entropy, and stability for the shocks. Often, these limiters depend on suitable parameters that may change from one test case to another. Last, some of these limiters are based on the basis of the reference element, while others use a polynomial approximation in the physical space.

Another problem is raised by the large scale computations: it is well known that classical operations such as multiplication and addition that are associative and commutative are actually no more when floating point arithmetic is considered. As a consequence, the same series of operations executed on two different threads or two different MPI process *but in a different order* may not give the same result. This behavior is error prone, and often happens only with optimized version at large scale, which makes it nearly impossible to track.

This strong complexity may jeopardize the maintainability of the code, but also question the significance of published results if the exact implementation and execution conditions cannot be reproduced. This is why a lot of unit tests have been developed in AEROSOL. Also, the dependencies are under a tight control thanks to the usage of the package manager GUIX. In the future, we would like to go further, not only by extending the number of operational test cases, but also by publishing systematically for each numerical result the git hash of the code, the full parameter file used, the mesh used, and the tools used for postprocessing. Reproducibility is one of the cornerstone of science, and we have a room for improvement on this concern.

Bibliography

[1] R. Abgrall, D. De Santis, and M. Ricchiuto. High-order preserving residual distribution schemes for advection-diffusion scalar problems on arbitrary grids. *SIAM Journal on Scientific Computing*, 36(3):A955–A983, 2014.

[2] R. Abgrall and V. Perrier. Asymptotic expansion of a multiscale numerical scheme for compressible multiphase flow. *Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal*, 5(1):84–115, 2006.

[3] R. Abgrall and R. Saurel. Discrete equations for physical and numerical compressible multiphase mixtures. *Journal of Computational Physics*, 186(2):361–396, 2003.

[4] R. J. Adler and J. E. Taylor. *Random fields and geometry*. Springer Science & Business Media, 2009.

[5] F. Alauzet and P. Frey. Estimateur d'erreur géométrique et métriques anisotropes pour l'adaptation de maillage: Partie II: exemples d'applications. Technical report, INRIA, 2003.

[6] D. Amenga-Mbengoue, D. Genet, C. Lachat, E. Martin, M. Mogé, V. Perrier, F. Renac, M. Ricchiuto, and F. Rue. Comparison of high order algorithms in Aerosol and Aghora for compressible flows. *ESAIM: Proceedings*, 43:1–16, Dec. 2013.

[7] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39(5):1749–1779, 2002.

[8] J. Badwaik, M. Boileau, D. Coulette, E. Franck, P. Helluy, C. Klingenberg, L. Mendoza, and H. Oberlin. Task-based parallelization of an implicit kinetic scheme. *ESAIM: Proceedings and Surveys*, 63:60–77, 2018.

[9] M. Baer and J. Nunziato. A two-phase mixture theory for the deflagration-to-detonation transition (DDT) in reactive granular materials. *International Journal of Multiphase Flow*, 12(6):861–889, 1986.

[10] G. Bal, J. Garnier, S. Motsch, and V. Perrier. Random integrals and correctors in homogenization. *Asymptotic Analysis*, 59(1-2):1–26, 2008.

[11] F. Bassi, L. Botti, A. Colombo, D. A. Di Pietro, and P. Tesini. On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations. *Journal of Computational Physics*, 231(1):45–65, 2012.

[12] F. Bassi, C. De Bartolo, R. Hartmann, and A. Nigro. A discontinuous Galerkin method for inviscid low Mach number flows. *Journal of Computational Physics*, 228(11):3996–4011, 2009.

[13] R. Borges, M. Carmona, B. Costa, and W. S. Don. An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. *Journal of Computational Physics*, 227(6):3191–3211, 2008.

[14] B. Bramas, P. Helluy, L. Mendoza, and B. Weber. Optimization of a discontinuous Galerkin solver with OpenCL and StarPU. *International Journal on Finite Volumes*, 15(1):1–19, 2020.

[15] D. Bresch and M. Hillairet. A compressible multifluid system with new physical relaxation terms. *Annales Scientifiques de l'École Normale Supérieure*, 52(2):255–295, 2019.

[16] P. Bruel, S. Delmas, J. Jung, and V. Perrier. A low Mach correction able to deal with low Mach acoustics. *Journal of Computational Physics*, 378:723–759, Feb. 2019.

[17] M. Castro, B. Costa, and W. S. Don. High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 230(5):1766–1792, 2011.

[18] C. Cattaneo. A form of heat conduction equation which eliminates the paradox of instantaneous propagation. *Compte Rendus*, 247(4):431–433, 1958.

[19] C. Chalons, M. Girardin, and S. Kokh. An all-regime Lagrange-Projection like scheme for the gas dynamics equations on unstructured meshes. *Communications in Computational Physics*, 20(1):188–233, 2016.

[20] J.-B. Chapelier, M. De La Llave Plata, F. Renac, and E. Lamballais. Evaluation of a high-order discontinuous Galerkin method for the DNS of turbulent flows. *Computers & Fluids*, 95:210–226, 2014.

[21] A. Chinnayya, E. Daniel, and R. Saurel. Modelling detonation waves in heterogeneous energetic materials. *Journal of Computational Physics*, 196(2):490–538, 2004.

[22] A. Chinnayya, E. Daniel, and R. Saurel. Modelling detonation waves in heterogeneous energetic materials. *Journal of Computational Physics*, 196:490–538, 2004.

[23] S. Clain, S. Diot, and R. Loubère. A high-order finite volume method for systems of conservation laws—multi-dimensional optimal order detection (MOOD). *Journal of Computational Physics*, 230(10):4028–4050, 2011.

[24] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework. *Mathematics of Computation*, 52(186):411–435, 1989.

[25] B. Cockburn and C.-W. Shu. Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16(3):173–261, 2001.

[26] G. N. Coleman, J. Kim, and R. Moser. A numerical study of turbulent supersonic isothermal-wall channel flow. *Journal of Fluid Mechanics*, 305:159–184, 1995.

[27] D. Coulette, E. Franck, P. Helluy, M. Mehrenberger, and L. Navoret. High-order implicit palindromic discontinuous Galerkin method for kinetic-relaxation approximation. *Computers & Fluids*, 190:485–502, 2019.

[28] G. Dal Maso, P. G. Lefloch, and F. Murat. Definition and weak stability of nonconservative products. *Journal de Mathématiques Pures et Appliquées. Neuvième Série*, 74(6):483–548, 1995.

[29] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, and M. Wesenberg. Hyperbolic divergence cleaning for the MHD equations. *Journal of Computational Physics*, 175(2):645–673, 2002.

[30] P. Degond and M. Tang. All speed scheme for the low Mach number limit of the isentropic Euler equations. *Communications in Computational Physics*, 10(1):1–31, 2011.

[31] S. Dellacherie. Analysis of Godunov type schemes applied to the compressible Euler system at low Mach number. *Journal of Computational Physics*, 4(229):978–1016, 2010.

[32] S. Dellacherie, J. Jung, P. Omnes, and P.-A. Raviart. Construction of modified Godunov type schemes accurate at any Mach number for the compressible Euler system. *Mathematical Models and Methods in Applied Sciences*, 26(13):2525–2615, Nov. 2016.

[33] S. Dellacherie, P. Omnes, and F. Rieper. The influence of cell geometry on the Godunov scheme applied to the linear wave equation. *Journal of Computational Physics*, 229(14):5315–5338, 2010.

[34] S. Delmas. *Simulation numérique directe d'un jet en écoulement transverse à bas nombre de Mach en vue de l'amélioration du refroidissement par effusion des chambres de combustion aéronautiques*. PhD thesis, Pau, 2015.

[35] G. Dimarco, R. Loubère, and M.-H. Vignal. Study of a new asymptotic preserving scheme for the Euler system in the low Mach number limit. *SIAM Journal on Scientific Computing*, 39(5):A2099–A2128, 2017.

[36] W.-S. Don and R. Borges. Accuracy of the weighted essentially non-oscillatory conservative finite difference schemes. *Journal of Computational Physics*, 250:347–372, 2013.

[37] D. A. Drew and S. L. Passman. *Theory of multicomponent fluids*, volume 135 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1999.

[38] M. Dumbser, W. Boscheri, M. Semplice, and G. Russo. Central weighted ENO schemes for hyperbolic conservation laws on fixed and moving unstructured meshes. *SIAM Journal on Scientific Computing*, 39(6):A2564–A2591, 2017.

[39] M. Dumbser, A. Hidalgo, M. Castro, C. Parés, and E. F. Toro. FORCE schemes on unstructured meshes II: Non-conservative hyperbolic systems. *Computer Methods in Applied Mechanics and Engineering*, 199(9-12):625–647, 2010.

[40] M. Dumbser, I. Peshkov, E. Romenski, and O. Zanotti. High order ADER schemes for a unified first order hyperbolic formulation of continuum mechanics: Viscous heat-conducting fluids and elastic solids. *Journal of Computational Physics*, 314:824–862, 2016.

[41] M. Dumbser, I. Peshkov, E. Romenski, and O. Zanotti. High order ADER schemes for a unified first order hyperbolic formulation of Newtonian continuum mechanics coupled with electro-dynamics. *Journal of Computational Physics*, 348:298–342, 2017.

[42] A. Eberle, M. Schmatz, and N. Bissinger. Generalized fluxvectors for hypersonic shock-capturing. In *28th Aerospace Sciences Meeting*, page 390, 1990.

[43] S. Eriksson and J. Nordström. Exact non-reflecting boundary conditions revisited: well-posedness and stability. *Foundations of Computational Mathematics*, pages 1–30, 2016.

[44] M. Essadki, J. Jung, A. Larat, M. Pelletier, and V. Perrier. A task-driven implementation of a simple numerical solver for hyperbolic conservation laws. *ESAIM: Proceedings and Surveys*, 63:228–247, Jan. 2017.

[45] R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, et al. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152(2):457–492, 1999.

[46] A. G. Filippini, S. De Brye, V. Perrier, F. Marche, M. Ricchiuto, D. Lannes, and P. Bonneton. UHAINA : A parallel high performance unstructured near-shore wave model. In D. Levacher, M. Sanchez, X. Bertin, and I. Brenon, editors, *Journées Nationales Génie Côtier - Génie Civil*, volume 15 of *La Rochelle*, pages 47–56, La Rochelle, France, May 2018. Editions Paralia.

[47] J. L. Florenciano Merino. *Étude de la réponse d'un écoulement avec transfert pariétal de masse à un forçage acoustique : application au refroidissement des chambres de combustion aéronautiques*. PhD thesis, École doctorale Sciences Exactes Université de Pau et des Pays de l'Adour, 2013.

[48] J.-P. Fouque, J. Garnier, G. Papanicolaou, and K. Solna. *Wave propagation and time reversal in randomly layered media*, volume 56. Springer Science & Business Media, 2007.

[49] R. O. Fox, F. Laurent, and M. Massot. Numerical simulation of spray coalescence in an Eulerian framework: direct quadrature method of moments and multi-fluid method. *Journal of Computational Physics*, 227(6):3058–3088, 2008.

[50] E. Franquet and V. Perrier. Runge-Kutta discontinuous Galerkin method for interface flows with a maximum preserving limiter. *Computers & Fluids*, 65:2–7, Mar. 2012.

[51] E. Franquet and V. Perrier. Runge-Kutta discontinuous Galerkin method for the approximation of Baer and Nunziato type multiphase models. *Journal of Computational Physics*, 231(11):4096–4141, Feb. 2012.

[52] E. Franquet and V. Perrier. Runge-Kutta discontinuous Galerkin method for reactive multiphase flows. *Computers & Fluids*, 83:157–163, Aug. 2013.

[53] D. Furfaro and R. Saurel. A simple HLLC-type Riemann solver for compressible non-equilibrium two-phase flows. *Computers & Fluids*, 111:159–178, 2015.

[54] T. Gallouët, P. Helluy, J.-M. Hérard, and J. Nussbaum. Hyperbolic relaxation models for granular flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(2):371–400, 2010.

[55] D. Genet. *Design of generic modular solutions for PDE solvers for modern architectures.* Theses, Université de Bordeaux, Dec. 2014.

[56] F. Golay and P. Helluy. Numerical schemes for low mach wave breaking. *International Journal of Computational Fluid Dynamics*, 21(2):69–86, 2007.

[57] J.-L. Guermond, B. Popov, and I. Tomas. Invariant domain preserving discretization-independent schemes and convex limiting for hyperbolic systems. *Computer Methods in Applied Mechanics and Engineering*, 347:143–175, 2019.

[58] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational physics*, 152(2):423–456, 1999.

[59] H. Guillard. On the behavior of upwind schemes in the low Mach number limit. IV: P0 approximation on triangular and tetrahedral cells. *Computers & Fluids*, 38(10):1969–1972, 2009.

[60] H. Guillard and B. Nkonga. On the behaviour of upwind schemes in the low Mach number limit: A review. *Handbook of Numerical Analysis*, 18:203–231, 2017.

[61] H. Guillard and C. Viozat. On the behaviour of upwind schemes in the low Mach number limit. *Computers & fluids*, 28(1):63–86, 1999.

[62] J.-F. Haas and B. Sturtevant. Interactions of weak shock waves with cylindrical and spherical gas inhomogeneities. *Journal of Fluid Mechanics*, 181:4l–76, 1987.

[63] E. Han, M. Hantke, and S. Müller. Modeling of multi-component flows with phase transition and application to collapsing bubbles. Technical report, Institut für Geometrie une Praktische Mathematik, Preprint No. 409, 2014.

[64] E. Han, M. Hantke, and S. Müller. Efficient and robust relaxation procedures for multi-component mixtures including phase transition. *Journal of Computational Physics*, 338:217–239, 2017.

[65] P. Helluy and J. Jonathan. Opencl simulations of two-fluid compressible flows with a random choice method. *International Journal of Finite Volumes (http://www.latp.univ-mrs.fr/IJFV/)*, 10, 2013.

[66] P. Helluy and J. Jung. Interpolated pressure laws in two-fluid simulations and hyperbolicity. In *Finite Volumes for Complex Applications VII-Methods and Theoretical Aspects*, pages 37–53. Springer, Cham, 2014.

[67] J.-M. Hérard. A three-phase flow model. *Mathematical and Computer Modelling*, 45:732–755, 2007.

[68] C. W. Hirt, A. A. Amsden, and J. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational physics*, 14(3):227–253, 1974.

[69] C. Hu and C.-W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150(1):97–127, 1999.

[70] D. Iampietro, F. Daude, P. Galon, and J.-M. Hérard. A Mach-sensitive implicit-explicit scheme adapted to compressible multi-scale flows. *Journal of Computational and Applied Mathematics*, 340:122–150, Oct. 2018.

[71] D. Iampietro, F. Daude, P. Galon, and J.-M. Hérard. A Mach-sensitive splitting approach for Euler-like systems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 2018.

[72] N. Jarrin. *Synthetic inflow boundary conditions for the numerical simulation of turbulence*. PhD thesis, University of Manchester, 2008.

[73] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational physics*, 126(1):202–228, 1996.

[74] J. Jung and V. Perrier. Behaviour of the discontinuous Galerkin method for compressible flows at low Mach number on triangles and tetrahedrons. *In preparation*, 2020.

[75] J. Jung and V. Perrier. Long time behaviour of linear hyperbolic systems and application to approximation of stationary low Mach number flows. *Submitted*, 2020.

[76] A. Kapila, R. Menikoff, J. Bdzil, S. Son, and D. S. Stewart. Two-phase modeling of deflagration-to-detonation transition in granular materials: Reduced equations. *Physics of fluids*, 13(10):3002–3024, 2001.

[77] J. Kim, P. Moin, and R. Moser. Turbulence statistics in fully developed channel flow at low Reynolds number. *Journal of Fluid Mechanics*, 177:133–166, 1987.

[78] S. Klainerman and A. Majda. Singular limits of quasilinear hyperbolic systems with large parameters and the incompressible limit of compressible fluids. *Communications on pure and applied Mathematics*, 34(4):481–524, 1981.

[79] T. Kloczko, C. Corre, and A. Beccantini. Low-cost implicit schemes for all-speed flows on unstructured meshes. *International Journal for Numerical Methods in Fluids*, 58(5):493–526, 2008.

[80] D. A. Knoll and D. E. Keyes. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.

[81] O. Kolb. On the full and global accuracy of a compact third order WENO scheme. *SIAM Journal on Numerical Analysis*, 52(5):2335–2355, 2014.

[82] A. Kumar. Numerical analysis of the scramjet inlet flow field using two-dimensional Navier-Stokes equations. In *19th Aerospace Sciences Meeting*, page 185, 1981.

[83] O. Le Métayer, J. Massoni, and R. Saurel. Modelling evaporation fronts with reactive Riemann solvers. *Journal of Computational Physics*, 205(2):567–610, 2005.

[84] S. LeMartelot, B. Nkonga, and R. Saurel. Liquid and liquid–gas flows at all speeds. *Journal of Computational Physics*, 255:53–82, 2013.

[85] D. Levy, G. Puppo, and G. Russo. Central WENO schemes for hyperbolic systems of conservation laws. *ESAIM: Mathematical Modelling and Numerical Analysis*, 33(3):547–571, 1999.

[86] X.-S. Li and C.-W. Gu. An all-speed Roe-type scheme and its asymptotic analysis of low Mach number behaviour. *Journal of Computational Physics*, 227(10):5144–5159, 2008.

[87] X.-S. Li, C.-W. Gu, and J.-Z. Xu. Development of Roe-type scheme for all-speed flows based on preconditioning method. *Computers & Fluids*, 38(4):810–817, 2009.

[88] T. S. Lund, X. Wu, and K. D. Squires. On the Generation of Turbulent Inflow Conditions for Boundary Layer Simulations. *Journal of Computational Physics*, 140:233–258, 1998.

[89] R. Manceau. Recent progress in the development of the Elliptic Blending Reynolds-stress model. *International Journal of Heat and Fluid Flow*, 51:195–220, 2015.

[90] R. J. Margason. The path of a jet directed at large angles to a subsonic free stream. Technical Report NASA-TN-D-4919, NASA Langley Research Center, 1968.

[91] A. Mazaheri and H. Nishikawa. Very efficient high-order hyperbolic schemes for time-dependent advection–diffusion problems: Third-, fourth-, and sixth-order. *Computers & Fluids*, 102:131–147, 2014.

[92] A. Mazaheri and H. Nishikawa. Efficient high-order discontinuous Galerkin schemes with first-order hyperbolic advection–diffusion system approach. *Journal of Computational Physics*, 321:729–754, 2016.

[93] A. Mazaheri, C.-W. Shu, and V. Perrier. Bounded and compact weighted essentially nonoscillatory limiters for discontinuous Galerkin schemes: Triangular elements. *Journal of Computational Physics*, 395:461–488, Oct. 2019.

[94] F. Montomoli and S. Eastwood. Implementation of synthetic turbulence inlet for turbomachinery LES. *Computers & Fluids*, 46(1):369–374, 2011.

[95] B. Müller. Low Mach number asymptotics of the Navier-Stokes equations and numerical implications. In *VKI Lecture Series on Computational fluid dynamics*, pages 1–52, 1999.

[96] S. Müller, M. Hantke, and P. Richter. Closure conditions for non-equilibrium multi-component models. *Continuum Mechanics and Thermodynamics*, 28:1157–1189, 2016.

[97] C.-D. Munz, S. Roller, R. Klein, and K. J. Geratz. The extension of incompressible flow solvers to the weakly compressible regime. *Computers & Fluids*, 32(2):173–196, 2003.

[98] A. Murrone and H. Guillard. A five equation reduced model for compressible two phase flow problems. *Journal of Computational Physics*, 202(2):664–698, 2005.

[99] A. Nigro, C. De Bartolo, R. Hartmann, and F. Bassi. Discontinuous Galerkin solution of preconditioned Euler equations for very low Mach number flows. *International Journal for Numerical Methods in Fluids*, 63(4):449–467, 2010.

[100] A. Nigro, S. Renda, C. De Bartolo, R. Hartmann, and F. Bassi. A high-order accurate discontinuous Galerkin finite element method for laminar low Mach number flows. *Internat. J. Numer. Methods Fluids*, 72(1):43–68, 2013.

[101] H. Nishikawa. A first-order system approach for diffusion equation. I: Second-order residual-distribution schemes. *Journal of Computational Physics*, 227(1):315–352, 2007.

[102] H. Nishikawa. New-generation hyperbolic Navier-Stokes schemes: O (1/h) speed-up and accurate viscous/heat fluxes. In *20th AIAA Computational Fluid Dynamics Conference*, page 3043, 2011.

[103] H. Nishikawa. Alternative formulations for first-, second-, and third-order hyperbolic Navier-Stokes schemes. In *22nd AIAA Computational Fluid Dynamics Conference*, page 2451, 2015.

[104] H. Nishikawa and P. Roe. On high-order fluctuation-splitting schemes for Navier-Stokes equations. In *Computational Fluid Dynamics 2004*, pages 799–804. Springer, 2006.

[105] S. Noelle, G. Bispen, K. R. Arun, M. Lukáčová-Medviďová, and C.-D. Munz. A weakly asymptotic preserving low Mach number scheme for the Euler equations of gas dynamics. *SIAM Journal on Scientific Computing*, 36(6):B989–B1024, 2014.

[106] S. Osher, R. Fedkiw, and K. Piechor. Level set methods and dynamic implicit surfaces. *Appl. Mech. Rev.*, 57(3):B15–B15, 2004.

[107] K. Oßwald, A. Siegmund, P. Birken, V. Hannemann, and A. Meister. L^2Roe: a low dissipation version of Roe's approximate Riemann solver for low Mach numbers. *Internat. J. Numer. Methods Fluids*, 81(2):71–86, 2016.

[108] M. Pamiès, P. É. Weiss, E. Garnier, S. Deck, and P. Sagaut. Generation of synthetic turbulent inflow data for large eddy simulation of spatially evolving wall-bounded flows. *Physics of Fluids*, 21, 2009.

[109] M. Parisot and J.-P. Vila. Centered-potential regularization for the advection upstream splitting method. *SIAM Journal on Numerical Analysis*, 54(5):3083–3104, 2016.

[110] S. Peluchon, G. Gallice, and L. Mieussens. A robust implicit–explicit acoustic-transport splitting scheme for two-phase flows. *Journal of Computational Physics*, 339:328–355, 2017.

[111] V. Perrier. *Numerical simulation of compressible multiphase flows with or without phase transition. Application to laser plasma interaction.* Theses, Université Sciences et Technologies - Bordeaux I, July 2007.

[112] V. Perrier. The Chapman-Jouguet closure for the Riemann Problem with vaporization. *SIAM Journal on Applied Mathematics*, pages 1333–1359, 2008.

[113] V. Perrier. A conservative method for the simulation of the isothermal Euler system with the van-der-Waals equation of state. *Journal of Scientific Computing*, 48(1-3):296–303, 2011.

[114] V. Perrier. Definition of the configurations calculated by DNS available, Feb. 2013. 1er livrable de l'UPPA dans le cadre du programme européen IMPACT-AE.

[115] V. Perrier. Report on DNS of isothermal flows with adiabatic walls, Jan. 2016. Deuxième livrable de l'UPPA dans le cadre du programme européen IMPACT-AE.

[116] V. Perrier. Report on DNS of non-isothermal flows with adiabatic walls, June 2016. Troisième livrable de l'UPPA dans le cadre du programme européen IMPACT-AE.

[117] V. Perrier and E. Gutiérrez. Derivation and closure of Baer and Nunziato type multiphase models by averaging a simple stochastic model. *Submitted*, 2020.

[118] V. Perrier and A. Mazaheri. First order formulation of advection-diffusion dissipative systems: symmetrizability, hyperbolicity and numerical approximation. *Submitted*, 2020.

[119] I. Peshkov and E. Romenski. A hyperbolic model for viscous Newtonian flows. *Continuum Mechanics and Thermodynamics*, 28(1-2):85–104, 2016.

[120] J. Qiu and C.-W. Shu. Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one-dimensional case. *Journal of Computational Physics*, 193(1):115–135, 2004.

[121] J. Qiu and C.-W. Shu. Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method II: Two dimensional case. *Computers & Fluids*, 34(6):642–663, 2005.

[122] J. Qiu and C.-W. Shu. Runge–Kutta discontinuous Galerkin method using WENO limiters. *SIAM Journal on Scientific Computing*, 26(3):907–929, 2005.

[123] J. Quirk and S. Karni. On the dynamics of a shock-bubble interaction. *Journal of Fluid Mechanics*, 318:129–163, 1996.

[124] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive computation and machine learning. MIT Press, 2006.

[125] S. Rhebergen, O. Bokhove, and J. J. W. van der Vegt. Discontinuous Galerkin finite element methods for hyperbolic nonconservative partial differential equations. *Journal of Computational Physics*, 227(3):1887–1922, 2008.

[126] S. O. Rice. Mathematical analysis of random noise. *Bell System Technical Journal*, 24(1):46–156, 1945.

[127] F. Rieper. On the dissipation mechanism of upwind-schemes in the low Mach number regime: A comparison between Roe and HLL. *Journal of Computational Physics*, 229(2):221–232, 2010.

[128] F. Rieper. A low-Mach number fix for Roe's approximate Riemann solver. *Journal of Computational Physics*, 230(13):5263–5287, 2011.

[129] F. Rieper and G. Bader. The influence of cell geometry on the accuracy of upwind schemes in the low Mach number regime. *Journal of Computational Physics*, 228(8):2918–2933, 2009.

[130] R. Saurel and R. Abgrall. A multiphase Godunov method for compressible multifluid and multiphase flows. *Journal of Computational Physics*, 150(2):425–467, 1999.

[131] R. Saurel, S. Gavrilyuk, and F. Renaud. A multiphase model with internal degrees of freedom: application to shock–bubble interaction. *Journal of Fluid Mechanics*, 495:283–321, 2003.

[132] R. Saurel and C. Pantano. Diffuse-interface capturing methods for compressible two-phase flows. *Annual Review of Fluid Mechanics*, 50:105–130, 2018.

[133] R. Saurel, F. Petitpas, and R. A. Berry. Simple and efficient relaxation methods for interfaces separating compressible fluids, cavitating flows and shocks in multiphase mixtures. *Journal of Computational Physics*, 228(5):1678–1712, 2009.

[134] K. Schmidmayer, F. Petitpas, E. Daniel, N. Favrie, and S. Gavrilyuk. A model and numerical method for compressible flows with capillary effects. *Journal of Computational Physics*, 334:468–496, 2017.

[135] S. Schochet. Fast singular limits of hyperbolic PDEs. *Journal of Differential Equations*, 114(2):476–512, 1994.

[136] D. Serre. The structure of dissipative viscous system of conservation laws. *Physica D: Nonlinear Phenomena*, 239(15):1381–1386, 2010.

[137] D. Serre. Multi-dimensional systems of conservation laws: An introductory lecture. In *Hyperbolic Conservation Laws and Related Analysis with Applications*, pages 243–309. Springer, 2014.

[138] J. Shi, C. Hu, and C.-W. Shu. A technique of treating negative weights in WENO schemes. *Journal of Computational physics*, 175(1):108–127, 2002.

[139] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *Journal of Computational Physics*, 83(1):32 – 78, 1989.

[140] S. Subramaniam. Lagrangian–Eulerian methods for multiphase flows. *Progress in Energy and Combustion Science*, 39(2-3):215–245, 2013.

[141] E. Toro. *Riemann solvers and numerical methods for fluid dynamics. A practical introduction.* Springer-Verlag, Berlin, second edition, 1999.

[142] E. Turkel. Preconditioned methods for solving the incompressible and low speed compressible equations. *Journal of Computational Physics*, 72(2):277–298, 1987.

[143] J. J. van der Vegt and S. Rhebergen. hp-multigrid as smoother algorithm for higher order discontinuous Galerkin discretizations of advection dominated flows: Part I. multilevel analysis. *Journal of Computational physics*, 231(22):7537–7563, 2012.

[144] J. VonNeumann and R. D. Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics*, 21(3):232–237, 1950.

[145] F. D. Witherden and P. E. Vincent. On the identification of symmetric quadrature rules for finite element methods. *Computers & Mathematics with Applications*, 69(10):1232–1241, 2015.

[146] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational physics*, 54(1):115–173, 1984.

[147] S. T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, 31(3):335–362, 1979.

[148] X. Zhang and C.-W. Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *Journal of Computational Physics*, 229(9):3091–3120, 2010.

[149] X. Zhang, Y. Xia, and C.-W. Shu. Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes. *Journal of Scientific Computing*, 50(1):29–62, 2012.

[150] Y.-T. Zhang and C.-W. Shu. Third order WENO scheme on three dimensional tetrahedral meshes. *Communications in Computational Physics*, 5(2-4):836–848, 2009.

[151] J. Zhu and J. Qiu. A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws. *Journal of Computational Physics*, 318:110–121, 2016.

[152] J. Zhu and J. Qiu. A new third order finite volume weighted essentially non-oscillatory scheme on tetrahedral meshes. *Journal of Computational Physics*, 349:220–232, 2017.

[153] J. Zhu, J. Qiu, C.-W. Shu, and M. Dumbser. Runge–Kutta discontinuous Galerkin method using WENO limiters II: unstructured meshes. *Journal of Computational Physics*, 227(9):4330–4353, 2008.

[154] J. Zhu and C.-W. Shu. Numerical study on the convergence to steady state solutions of a new class of high order WENO schemes. *Journal of Computational Physics*, 349:80–96, 2017.

[155] J. Zhu and C.-W. Shu. Numerical study on the convergence to steady-state solutions of a new class of finite volume WENO schemes: triangular meshes. *Shock Waves*, 29(1):3–25, 2019.

[156] J. Zhu, X. Zhong, C.-W. Shu, and J. Qiu. Runge-Kutta discontinuous Galerkin method with a simple and compact Hermite WENO limiter on unstructured meshes. *Communications in Computational Physics*, 21(3):623–649, 2017.